A Computational Model of Color Perception and Color Naming

by

Johan Maurice Gisèle Lammens

June 1994

A dissertation submitted to
the Faculty of the Graduate School of
State University of New York at Buffalo
in partial fulfillment of the requirements for
the degree of
Doctor of Philosophy

Voor mijn ouders.

Acknowledgment

This work was supported in part by Equipment Grant No. EDUD-US-932022 from SUN Microsystems Computer Corporation, and in part by NASA under contract NAS 9-19004, which is gratefully acknowledged.

I would like to thank the members of my committee for their guidance, support, and valued comments. My advisor, Dr. Stuart C. Shapiro, has always given me plenty of leeway to explore new concepts (and occasionally make a fool of myself), yet he has provided direction and advice when necessary. His experience, insight, and open-mindedness are admirable. Dr. K. Nicholas Leibovic introduced me to biophysics and the interdisciplinary study of vision, and has served as my personal model of what a true scientist should be. His attention and encouragement have been a great support. Dr. William J. Rapaport has certainly broadened my philosophical and general intellectual horizon, and has always been helpful and supportive. Dr. Deborah K. Walters, finally, has inspired me to search for meaningful computational models with roots in biology and neurophysiology, and has often provided excellent feedback. My outside reader, Dr. Christopher M. Brown, has on many occasions provided illuminating comments and feedback on color-related issues and beyond. I have always found it challenging, stimulating, and *fun* to carry on a conversation with him, either in person or electronically.

I would also like to thank the Computer Science department for continuing to support me financially and otherwise for almost six years, in my various guises as teaching, laboratory, and research assistant. Ellie Benzel, Sally Elder, Gloria Koontz, and all the other secretaries deserve special mention as collective substitute-mothers for especially the foreign graduate students. They are truly the pillars upon which the department is built. My fellow graduate students in the SNePS research group and the department at large have been good colleagues, and in many instances great friends as well. I have greatly enjoyed meeting and working with people from all parts of the world. Many thanks also to Ken Smith, Devon Bowen, Davin Milun, Harry Delano, and the graduate "labbies" for providing technical support, allowing me to bumble in systems management, and being a great gang of people. I will fondly remember my days as a labbie, and I am probably as proud of my super-user status as of my new title. My only regret is that I cannot take my "sudo privs" with me. Harry has not only been the best possible boss but also a valued friend.

Thanks to Joe Piazza also, for helping me out with color printing, and to Niloufer Mackey, for advice on matters of linear algebra, and for being a good friend. One fellow graduate student deserves special mention: Henry Hexmoor, who has been my friend, house mate, and co-author of many a paper. He not only introduced me to some southern cooking, but also to robotics and related issues, and has certainly influenced my work for the better.

Thanks also to my friends Kulbir Arora, Paula "Peej" Freedman, Terry Wilmarth, and the occasional other members of our little Art Movie/dinner/general fun club. It provided the necessary entertainment and relaxation in times of need or despair. Buffalo is fortunate to have the beautiful North Park theater that shows excellent movies, even subtitled foreign ones (not to mention the cappuccino and chocolate chunk cookies)!

Many net.people have been very helpful in the course of my work. I have often been surprised by the kindness and helpfulness of people to whom I was just an unknown graduate student. *Long live the Internet!* Many thanks, therefore, to Richard Aslin, John Bradley, George Broadwell, Rene Collier, Reiner Eschbach, Brian Funt, Larry Hardin, Stevan Harnad, Roland Hausser, Keith Karns, Paul Kay, Judith Klavans, Peter Lennie, Haim Levkowitz, John Limber, Richard Lively, Luisa Maffi, Gerald Maguire, Walter Makous, David Mark, Ethan Montag, Carol Novak, Stephen Palmer, Steven Shafer, Kathy Straub, Kees Teunissen, Tom Wickham-Jones, David Zubin, and to those whose names I may have omitted unintentionally.

Many thanks also to my dear friends in Belgium and Holland and to my parents and family, for their support and encouragement, and for putting up with my long absence.

And, last but certainly not least, my heartfelt thanks to my s.o. Annalisa, without whose continuing love, support, encouragement, and patience I could not have finished the present work. I will forever fondly remember that particular class in Knowledge Representation where we met!

JL

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Research on human categorization[1] in cognitive science, psychology, linguistics, anthropology, and philosophy has shown that in many cases, categories cannot be properly characterized by a set of necessary and sufficient conditions; i.e. category membership is usually not an all-or-nothing phenomenon but rather a matter of degree, and people can judge the "typicality" and/or degree of membership of potential category members in a consistent way (see e.g. [Wittgenstein 1953, Berlin & Kay 1969, Rosch 1978, Lakoff 1987]). This applies not only to so-called *natural categories*, which correspond roughly to nouns in natural languages, but also to many categories used in science, e.g. biology [Lakoff 1987]. Perhaps the only exceptions are mathematical categories which are explicitly defined by necessary and sufficient conditions. However, even the latter might be seen as abstractions of real-world categories that would not necessarily be definable in the same way, or as being constructed on the basis of such abstractions, cf. [Aleksandrov 1956]. Some of the ideas about graded category membership have been formalized as *fuzzy set theory* [Zadeh 1971], which has found applications in various areas of control theory and AI, for instance [Kosko 1992].

The concepts of *embodiment* [Lakoff 1987], *symbol grounding* [Harnad 1990], and *situated*

[1] Mostly, categorization with respect to "natural categories", as in an ontology of what is "out there".

cognition [Suchman 1988] are related to issues in categorization. *Embodiment* is the notion that the shape or extension[2] of categories (and hence the meaning of symbols representing categories, in a referential type semantics) is in part determined by the physiology of the organism doing the categorization.[3] For instance, Berlin and Kay hypothesize that the universality of basic color category foci can be explained in terms of the underlying neuro-physiological mechanism of color perception, which are the same for all people, regardless of language [Berlin & Kay 1969].

Grounding is concerned with how symbols and their meanings are grounded in categorization and perception of the environment the organism operates in. Harnad's fundamental claim is that the symbols of "traditional" symbolic AI systems are only meaningful *to a human observer*, and not to the system itself.[4] The meaning to a human observer arises from the fact that the symbols are *systematically interpretable* and have a meaning assigned to them via an *external* semantic mapping or model. Such a semantic model is an explanatory device, but it does not play any role in the system's internal functioning. In contrast, Harnad claims that symbols representing categories to people (like nouns in natural languages) are meaningful because they are connected to the world in a *causal* and *non-arbitrary* way, via perception.[5] We can say that the human symbols are embodied, while machine symbols are not.[6]

I essentially agree with Harnad's analysis. Since categories are an essential part of natural language semantics, the analysis implies that models of natural language semantics must take physiology and perception, or in general, the nature of the cognitive mechanism underlying categorization, into account. This in turn implies that a traditional model-theoretic approach to natural language semantics is inadequate. Model theory, as used in logic or modern linguistics, starts with the assumption of a (real or possible) domain consisting of

[2] I am using the term *extension* here in the traditional model-theoretic sense of the set of objects in the model's domain that corresponds to a constant symbol. Although part of the argument developed in this dissertation goes against traditional model theory, I will continue to use the term in this way, for convenience.

[3] In particular, the physiology of perception, but perhaps also the physiology of motor control and even the physiology of emotion; cf. [Lakoff 1987].

[4] This is essentially the point that Searle tried to prove in his famous Chinese Room argument [Searle 1980]

[5] This is true for *directly grounded* symbols corresponding to categories of perception only. The hypothesis is that other symbols are *indirectly grounded* by being constructed out of directly grounded ones. Harnad is not very specific on this point.

[6] The intersection of the set of people using the term "symbol grounding" with the set of people using the term "embodiment" seems to be empty, but I will not let that disturb me.

discrete individuals, properties, and relations, corresponding via a *static* semantic mapping with constant and relation symbols in the language of interest. These models have nothing to say about how such a relation might be established or maintained in the first place. In essence this defines the central problem of semantics[7] away. Since I take natural language semantics to be part of the domain of the general study of intelligence, we may contrast the symbol grounding or embodiment view with the "traditional" symbolic AI view that intelligence (or "mental functions") can be studied in the abstract, without reference to the organism displaying it, or the mechanism implementing it (e.g. [Newell 1979]).

Situatedness holds (among other things) that "Communication ...is not a symbolic process that happens to go on in real-world settings, but a real-world activity in which we make use of language to delineate the collective relevance of a shared environment" [Suchman 1988, p. 180]. Again, the environment or the "real world" is seen as the grounding for language, which implicitly requires perception to be taken into account.

One particular area of natural language semantics where embodiment, grounding, and situatedness seem to play an important role is that of color terms. In their anthropological and linguistic work in the late sixties, Berlin and Kay [Berlin & Kay 1969] were looking for semantic universals in the domain of color terms, hoping to refute the Sapir-Whorf hypothesis which claims that there are no semantic universals, and that each language performs the coding of experience into language in a unique and arbitrary manner. The latter is of course in direct opposition to theories of embodied semantics, since these do predict the existence of universals, based on the observation that all people share a common physiology, regardless of the language they speak or the culture they live in. Berlin and Kay found that there are indeed semantic universals in the domain of color, particularly in the extensions of what they called "basic color terms".[8] When they asked native speakers of widely differing languages to identify (1) the *best examples* and (2) the *boundaries* of basic color categories on a chart of color samples, they found that (1) the best examples (foci) of basic color categories are the same within small tolerances for speakers of any language that has (the equivalent of) the basic color term in question, and (2) there is a hierarchy of languages with respect to how many and which basic color terms they possess, such that,

[7]According to Webster's online dictionary, "a branch of semiotics dealing with the relations between signs and what they refer to and including theories of denotation, extension, naming, and truth".

[8]In English, there are eleven basic color terms: *white, black, red, green, yellow, blue, brown, purple, pink, orange, grey.*

roughly speaking, a language that has $i+1$ basic color terms has all the basic color terms of any language with i basic color terms, and any languages with i basic color terms have the same ones (with respect to their extensions). It is of course apparent from these results that (basic) color categories are characterized by graded membership functions, with some colors clearly being non-members, some being prime examples, some being borderline examples, and with other degrees of membership in between.

[Kay & McDaniel 1978] have made the first attempt to explain these results based on neurophysiological findings in color perception, i.e. to specify how basic color categories are embodied, or to specify a theory of symbol grounding in the domain of colors. They used a fuzzy set model in which they interpreted (stylized versions of) neural response functions as characteristic functions of fuzzy sets representing color categories. Their model is interesting, it explains some of Berlin and Kay's data, and it certainly deserves respect for being the first to attempt to explain the connection between natural language semantics and physiology, but it does leave several questions unanswered, as I will discuss in Chapter 4. So far no adequate model of color term semantics has been computationally defined or implemented, to my knowledge.

In this dissertation I attempt to define a computational model of color perception and color naming, i.e. a semantic model of (basic) color terms grounded in color perception, that is partly based on neurophysiological data and that can explain Berlin and Kay's and other relevant linguistic and anthropological data. In particular, the model attempts to explain the graded nature of color categories and the universality of color foci, and it allows an artificial cognitive agent to name color samples, point out examples of named colors in its environment, and select objects from its environment, specified by color name. Such an agent can participate in an experiment like Berlin and Kay's, and its performance will be consistent with human performance. An implementation for the computational model is presented, as are some experimental results.

1.2 Thesis and Scientific Contributions

My central thesis is the following: *To define an adequate model of the semantics of color terms in natural languages, it is necessary to model the physiology of human color perception.*

Corollaries of the central thesis are:

1. Adequate models of natural language understanding require models of perception.

2. Purely symbolic models of natural language understanding, with or without model-theoretic underpinnings, are inadequate.

3. Adequate models of intelligent behavior require models of perception.

4. Purely symbolic models of intelligent behavior, with or without model-theoretic underpinnings, are inadequate.

Since the central thesis in its current form is hard to prove or falsify, my dissertation is actually concerned with an (admittedly weaker) existence proof of the following kind: It is possible to define an adequate model of the semantics of natural language color terms, and an adequate model of color naming and color pointing behavior, by modeling the physiology of human color perception. I consider an adequate model one that enables an autonomous robotic agent to name colors of objects in its field of view, and to point out examples of objects with specified colors in its environment, both in close agreement with human performance on the same tasks.

The scientific contributions of this dissertation are in the following areas:

1. Cognitive Agent Architecture. The work presented here is a case study of embodiment, symbol grounding, and situatedness in a natural language understanding context, and as such can help to clarify the problems involved. I present an analytic and well-defined approach to symbol grounding in a particular domain, which aids in the construction of a situated cognitive agent.

2. Natural Language Understanding. I define a computational model of the semantics of basic color terms that is not only explanatory, but also productive, in that it can be used in a cognitive agent architecture. The semantic model is intrinsic to the agent using it, and it does not suffer from problems related to the under-determining of reference in traditional symbolic models of natural language understanding and cognition. The model allows a computational cognitive agent to use color terms in a natural language, in a way that is consistent with native speakers' use of those terms.

3. Knowledge Representation and Reasoning. Although my research is not what is usually understood by knowledge representation and reasoning (KRR), namely the use of a formal language to represent human knowledge and perform inference on those representations, it is nevertheless relevant in that context. It provides the grounding for a set of terms which may be terms of a KRR system, e.g. (but not limited to) SNePS [Shapiro & Rapaport 1987]. The more analog nature of the model underlying this grounding can provide important semantic constraints on the formal (syntactic) manipulation of the terms.

4. Color models. I define a computational color model that is based on neurophysiological[9] data, and that can explain psychophysical[10] findings in color perception. The contribution lies in the partial bridging of the psychophysical and neurophysiological domains, explaining the former in terms of the latter. In addition, the model may be useful for computer vision work and potentially for computer graphics as well.

[9] Relating to the processes and phenomena of the nervous system.

[10] Relating the magnitude of a psychological response variable such as amount of red or green percept to some physical quality of the stimulus such as spectral composition of a light source, cf. [S. Edelman 1992].

Chapter 2

Problem Definition

This Chapter discusses in some detail the problems involved in defining a computational model of color perception and color naming, and defines terms and concepts involved.

2.1 Color Perception

As John Locke observed, objects do not have colors [Locke 1690]. Color is not a *physical* phenomenon, but a *perceptual* phenomenon that is related in a complex way to the spectral characteristics of electro-magnetic radiation in the visible wavelengths striking the retina [Wyszecki & Stiles 1982, Boynton 1990, Danger 1987, Ronchi 1957]. An illustration of the perceptual nature of colors is the phenomenon of *metamers*, or spectrally different stimuli that are indistinguishable to human observers, and hence will be perceived as the same color. For example, Judd has pointed out that the color of a Bunsen burner flame into which sodium is introduced is very similar to the color of an orange (the fruit) in daylight, yet their spectra are almost perfectly complementary [McIlwain & Dean 1956, p. 24]. In fact there are infinitely many different spectral stimuli that will be perceived as the same. In the discussion that follows I roughly adhere to a division into the three domains of physics, physiology, and psychology that are involved in optical phenomena, following [Ronchi 1957].[1]

[1] I am using this tripartite structure as an organizational tool only; no theoretical claims about the boundaries between these domains are intended.

The physical stimulus involved in color perception is light, or more precisely: electromagnetic radiation in the visible wavelength range, approximately 380–770 nm. I will represent the spectral energy distribution of the stimulus (the radiation striking the retina) as a function of wavelength $E(\lambda)$ after [Boynton 1990].[2] The stimulus energy distribution $E(\lambda)$ of the radiation that is reflected off (or transmitted by) an object is determined by (1) the spectral distribution of the radiation incident on the object (the light source) $S(\lambda)$, and (2) the spectral reflectance characteristics of the object $R(\lambda)$, so we can write

$$E(\lambda) = S(\lambda)R(\lambda) \tag{2.1}$$

in a simplified form.[3] It is easily seen from this equation that changing the light source or changing the reflectance characteristics of the object changes the stimulus $E(\lambda)$. Were we informally to think of *object color* as $R(\lambda)$, then the task of the color vision system is to recover $R(\lambda)$ from $E(\lambda)$ (for some examples of this approach, see [Maloney & Wandell 1986, Maloney 1993]). Or as [Zeki 1993] puts it:

> The brain strives to acquire a knowledge about the permanent, invariant and unchanging properties of objects and surfaces in our visual world. But the acquisition of that knowledge is no easy matter because the visual world is in a continual state of change. Thus, the brain can only acquire knowledge about the invariant properties of objects and surfaces if it is able to discard the continually changing information reaching it from the visual environment. [p. 355]

The human visual system is to some extent able to identify the same "object color" (surface reflectance, under the view just mentioned) under widely varying lighting conditions, a phenomenon known as *color constancy*, but I will not be concerned with that. Color constancy introduces considerable complexity that is not central to the aims of my work. What I am concerned with is the internal structure of the color categories that invariant object colors are perceived as belonging to. That is not to say that I will adhere to the "color science" aperture mode of color perception either; I will take a psycho-physical approach to modeling

[2] λ is typically used to represent wavelength in nanometer (nm). E then represents the energy of the stimulus as a function of wavelength.

[3] This simplification does not take into account additional factors like incident and reflection angles, transmission absorption and refraction, etc.

color perception, while at the same time keeping neurophysiological findings in mind.

The response of the visual system to $E(\lambda)$ brings us into the domain of neurophysiology. The human visual system is sensitive to differences in $E(\lambda)$ due to the presence of three photo-pigments with different spectral sensitivities in three types of photoreceptor cells (called cones) in the retina [Leibovic 1990b]. The cone types are sometimes called *red*, *green*, and *blue*, for the spectral color corresponding approximately to their wavelength of maximum sensitivity, but more appropriate designations are *long*, *medium*, and *short* wavelength sensitive cones, respectively.[4] If we represent the cone action spectra (or linear transforms thereof, see [Horn 1986]) as $\overline{x}(\lambda), \overline{y}(\lambda), \overline{z}(\lambda)$ and integrate them with the spectral energy distribution $E(\lambda)$ of the stimulus [Boynton 1990, Horn 1986], we obtain the so-called CIE[5] tristimulus values X, Y, Z:

$$X = \int E(\lambda)\overline{x}(\lambda)d\lambda \tag{2.2}$$

$$Y = \int E(\lambda)\overline{y}(\lambda)d\lambda \tag{2.3}$$

$$Z = \int E(\lambda)\overline{z}(\lambda)d\lambda \tag{2.4}$$

Any two stimuli that result in identical X, Y, Z values cannot be distinguished from each other; i.e. they are metamers.

So far I have only described the response of the cone type photoreceptors to $E(\lambda)$. A lot more is involved in the physiology of color perception, and below I will refer to subsequent stages of processing as necessary.

From the psychological (or psychophysical) point of view, there is a lot more to be said about color perception as well, see e.g. [Boynton 1990, Boynton 1979]. Some important psychophysical dimensions of color perception are brightness or lightness (how bright a visual stimulus appears to be, viewed in isolation or in context), chromaticity or hue (what enables us to distinguish between equally bright and texturally identical fields — this is the closest to the intuitive concept of color), and saturation (how pure a perceived color is,

[4] *Spectral colors* are the pure monochromatic components that white light can be split into by use of a prism ("rainbow colors"), corresponding physically to pure single-wavelength radiation. One can also think of spectral colors as impulse functions in the frequency domain.

[5] Commission Internationale de l'Eclairage, or International Committee on Lighting, a Paris-based standards organization.

or how unlike grey). Another important concept is chromatic context, or the finding that the appearance of a color depends importantly on its surroundings. Also, the intensity of the light source, and even memory and psychological context can affect color appearance [Boynton 1979]. I will return to some of these issues below.

2.2 Color Naming

Color naming also belongs in the domain of psychology. I define *color naming* as a mapping $\mathcal{N}$ from visual stimuli to pairs of color terms (or symbols or names) and "confidence", "goodness", or "typicality" measures.

More precisely, since I am not interested in spatial characteristics of visual stimuli, I will represent stimuli as spectral distributions $E(\lambda)$ associated with single points in the visual field only. The domain of $\mathcal{N}$ is thus the set of all such distributions:

$$\mathbf{E} = \{E \mid E : \lambda \mapsto \mathbf{R}^{+}\} \tag{2.5}$$

where $\lambda = [380, 770]$ represents wavelength in nm, and E represents a spectral distribution. The functions E will not be further defined. They need not be continuous or differentiable, for instance. A pure monochromatic stimulus would be represented as an impulse function, and mixtures of several monochromatic primaries would be represented as multi-modal distributions. The mapping $\mathcal{N}$ can then be defined as

$$\mathcal{N} : \mathbf{E} \mapsto \mathbf{C} \times \mathbf{I} \tag{2.6}$$

where $\mathbf{C}$ is an enumerable set of undefined terms, and $\mathbf{I}$ is the closed unit interval $[0, 1]$.

Informally, the preferred interpretation[6] of the model is that $\mathbf{E}$ represents the set of spectral distributions that function as input to the visual system, $\mathbf{C}$ represents a set of basic color terms, e.g. $\{white, black, red, green, yellow, blue, brown, pink, purple, orange, grey\}$, which is the set of basic color terms described in [Berlin & Kay 1969], and $\mathbf{I}$ represents the

[6]That is, the *external* interpretation of the mathematical model, not to be confused with a grounded semantic model used by a cognitive agent.

set of "confidence", "goodness", or "typicality" measures.

If we ignore the i part of the $\langle c, i\rangle \in \mathbf{C} \times \mathbf{I}$, we might think of $\mathbf{C}$ as representing a partition of the set of all possible color percepts $\mathbf{E}$ into a number of equivalence classes, one for each color term. If we do take the i part into account, we could model the response to any given stimulus $E \in \mathbf{E}$, i.e. $\mathcal{N}(E)$, as a *fuzzy partition* of the set of color terms $\mathbf{C}$ [Zadeh 1971, Kay & McDaniel 1978], which constrains the numbers i of the pairs $\langle c, i\rangle \in \mathbf{C} \times \mathbf{I}$ to sum to 1, or

$$\sum_{c \in \mathbf{C}} 2^{nd}\langle c, i\rangle = 1 \tag{2.7}$$

where 2^{nd} is the selector of the second element of an n-tuple, for any $\mathcal{N}(E)$. The mapping $\mathcal{N}$ then defines a set of membership or characteristic functions f_c, one for each color category $c \in \mathbf{C}$, on the universe $\mathbf{E}$. It is not clear what the advantage of such a model would be, however, or how well it fits the data on human color categorization.

The definition of $\mathcal{N}$ implies that I will not be concerned with recovering $R(\lambda)$ or $S(\lambda)$ from $E(\lambda)$, as noted above. A practical consequence of this is that changing the lighting of a scene may yield a different value of $\mathcal{N}$. I don't consider this a problem as long as the change is consistent with human performance on the same task.

From the brief discussion of the physiology of color vision above, it follows that if we want to model the relation between the domain of $E(\lambda)$ and a set of color terms, i.e. if we want to model *color naming*, it is not sufficient to define the extensions of color terms as intervals on the wavelength range between 380 and 770 nm. In particular, this approach could only work for pure monochromatic stimuli, which are very rare in real-world situations, and it would not explain the typical graded membership functions one finds in anthropological and linguistic research when subjects are asked to identify best examples and maximal extensions of color terms with respect to a set of color chips with known properties [Berlin & Kay 1969]. This approach would also leave out non-spectral colors like *purple* or *brown* altogether.[7] But perhaps the biggest objection against such an approach would be that it would constitute a *system-external* semantic model of color names, while our interest is in *system-internal* semantics, to be explained below. I claim that to model human *color*

[7] *Non-spectral colors* are colors that do not appear in the spectrum of white light, i.e. they are not single wavelength signals but complex signals. In the frequency domain, they are represented as non-impulse functions.

naming it is necessary to take human *color perception* into account, just as [Ronchi 1957] claims that it is necessary to take human physiology and psychology into account when studying optics, if that is defined with respect to visible light, i.e. implicitly with respect to an observer.

2.3 Outline of the Model

I will define a computational model of human color perception and color naming, i.e. construct an algorithmic mapping $\mathcal{N}$ as defined in Section 2.5, which is based in part on existing data about the neurophysiology and psychophysics of color perception, and which can explain existing anthropological and linguistic data on color naming as well. The model should allow an artificial cognitive agent (e.g. a GLAIR-agent [Lammens et al. 1994, Hexmoor et al. 1993c, Hexmoor et al. 1993b, Shapiro & Rapaport 1987]), when equipped with the necessary sensors and actuators, to

1. *Name* colors in response to a visual stimulus, and provide a confidence or "goodness of example" rating of its judgment; this requires evaluation of $\mathcal{N}(E)$ and some kind of thresholding on the resulting pairs $\langle c, i \rangle$.

2. *Point out* examples of named colors in its environment, and provide a confidence rating, and as a derivative of this capability, pick the *best example* of a named color from a set of color samples, or from its environment in general; this requires evaluation of $\mathcal{N}(E)$ over the whole visual field, and some kind of maximization on the resulting pairs $\langle c, i \rangle$.

The performance on these tasks must be consistent with human performance on the same tasks, as described in Chapter 4. In particular, this requires that

1. The model place the *foci* of basic color categories, as described in [Berlin & Kay 1969] and elsewhere, in the same regions of the color space as human subjects do

2. The model place the *boundaries* of basic color categories in the same regions of the color space (cf. Chapter 4) as human subjects do.

A more precise definition of "same region in the color space" will be given, to determine success in this area.

The model deals only to a limited extent with a number of important issues in color vision, most notably the effects of surrounds on perceived color, or in general the relation between spatial and color vision, and color constancy. These issues are dealt with only in as far as they are relevant to color naming.

The model has been integrated into a vision system capable of interacting with its environment as described above. I will refer to this system as the Color Labeling Robot (CLR),[8] after the Color Reader Robot described as a thought experiment in [Hausser 1989]. The model has also been tested on simulated data, and the results compared to known data about human color naming.

A system that exhibits the behavior described above is an example of a (partly) *embodied* [Lakoff 1987, Kay & McDaniel 1978] or *grounded* [Harnad 1990] system, or it can be seen as an instance of *situated cognition* [Suchman 1988].

[8]Thanks to Bill Rapaport for suggesting this acronym, perhaps the most important part of an AI system.

Chapter 3

Wider Significance

In this chapter, I describe what I believe to be the wider significance of my work with respect to some foundational issues in artificial intelligence and cognitive science, and to some extent also in philosophy.

3.1 Embodiment, Symbol Grounding, and Natural Language Semantics

The computational model of color perception and color naming described in this dissertation can be seen as a case study in embodiment, symbol grounding, and natural language semantics. The *symbol grounding problem* is about how to make the semantic interpretation of a formal symbol system intrinsic to the system, rather than just "parasitic on the meanings in our heads" [Harnad 1990, p. 335], i.e. only accessible to us, as designers or observers of the formal system, rather than to the formal system itself. The concept of symbol grounding is closely related to those of embodiment [Lakoff 1987] and situated action or cognition [Suchman 1988]. As mentioned before, the view of intelligence as being closely connected to the properties of the intelligent organism and the properties of the environment the organism operates in can be contrasted with a purely symbolic view, which holds that intelligence can be studied in the abstract, without reference to any organism (e.g., [Newell 1979]).

Since there is no clear definition available in the literature (Section 4.1), at least not for my purpose, I define *embodiment* as the notion that the representation, manipulation, and semantics of high-level symbolic concepts is in part determined by the physiology (the bodily functions) of an agent and in part by the interaction of the agent with the world. For instance, the semantics of color concepts is in part determined by the physiology of the color perception mechanism, and in part by the visual stimuli this mechanism interacts with. The result is the establishment of a mapping between symbolic color concepts and analog representations that reflect some properties of both the color perception mechanism and objects in the world.

My color perception and naming model grounds the color terms from the codomain of $\mathcal{N}$[1] in the perception of visual stimuli, the domain of $\mathcal{N}$, by connecting them via the mapping $\mathcal{N}$ itself. In other words, the mapping $\mathcal{N}$ constitutes a system-internal, referential semantic model of the color terms, or *embodies* the semantics of color terms for the agent that it is part of. Of course, such a model is an instance of *situated cognition* if we consider the color terms to be "mental" representations that are causally connected to the environment the robot operates in.

Note that one could simultaneously define a traditional *external*, model-theoretic semantic model of the color terms the robot is using, which might be more or less co-extensive with the internal semantic model.[2] Or a robot-psychologist could study the robot's behavior on color-related tasks and try to infer a semantic model from that. What's important about the internal model is the following:

1. Without it, the robot would not be able to perform any color-related tasks, since it could not perceive any colors at all.

2. In the internal model, there is no ambiguity about which "objects" symbols are paired with, since the relation is a causal one. As such, the model is immune to criticism of logical models as models of meaning that is based on the indeterminacy of reference or the under-determining of meaning by truth conditions, e.g., [Putnam 1981].

3. The pairing of terms with numerical "goodness" measures in the codomain of $\mathcal{N}$ al-

[1]The mapping from a color space to a set of color names, see Section 2.2.

[2]Chances are that the match will be less than perfect, however, without detailed knowledge of the internal model.

lows for both discrete, non-overlapping categories and graded, overlapping categories. To accurately model human color categorization, it is necessary to use the second kind of category [Berlin & Kay 1969, Kay & McDaniel 1978]. This property of the model reflects and makes explicit the difficulty involved in mapping a (for all practical purposes) continuous world onto a set of discrete symbols such as color terms. Subsequent processing in the symbolic domain may choose to ignore the overlapping and graded nature of the color categories by using only the term component of the pairs after applying a thresholding function to the numerical "goodness" component, but access to the "goodness" component is possible, if needed.

My thesis with respect to symbol grounding is a pragmatic one. I do not claim that, without grounding, a symbol system cannot truly "understand" the world, because that would require a consistent definition of "understanding", which is lacking to date. My thesis is that grounding, as I interpret it, enables a robotic agent to perform well on color-related tasks and that it provides a well-defined model of the agent's semantics of color terms that also adequately models the semantics of human color terms. The latter may be important for man-machine communication, as Winston has pointed out [Winston 1975, p.154].

3.2 Knowledge Representation and Semantics[3]

The work presented in this dissertation is relevant to the field of knowledge representation and reasoning (KRR), although it is not what is typically understood by that term. KRR involves a formal representation language[4] with a set of inference rules that operate on constructs of the language [B. Smith 1985]. Most KR languages can be translated relatively

[3] I am indebted to many people for the ideas presented in this section, too many to name them all. Discussions on this subject have taken place over a number of years, in settings ranging from CS department courses to the bar of a hotel in Italy. A few of the important conversation partners in this respect have been (in alphabetical order): Stevan Harnad, Henry Hexmoor, William J. Rapaport, Kenneth Regan, Stuart C. Shapiro, and Tim Smithers. The responsibility for what I've turned their ideas into is entirely mine, however.

[4] Or at least a representation language of some kind. The degree of formalism in KRR systems varies considerably, along the lines of what has been called the *neat–scruffy debate* [Rapaport 1992]. Any (symbolic) knowledge representation language uses terms of some kind, however, with an associated semantics not unlike what I describe here. My characterization of KRR is not meant to be exhaustive, however, and my remarks are addressed only to KRR systems of the kind alluded to.

easily into predicate logic of some kind,[5] so I will assume a predicate-logic-like language for the purpose of this discussion. The semantics of KR languages is typically (if at all) defined in terms of Tarskian semantics or derivatives thereof, with an interpretation function that maps terms and predicates of the language (the syntactic domain) into objects or sets of objects (or properties) in the domain of interpretation (the semantic domain), whether that domain is taken to be (part of) the real world, a possible world, or intensional objects in an agent's mind [Manin 1977, Shapiro & Rapaport 1987]. I see two problems with the practice of KRR as I have just outlined it (see also critics of traditional AI, e.g., [Harnad 1990], [Lakoff 1987], [Searle 1980], or recently [Angell 1993], to name just a few). First, the semantic models of KRR systems are purely hypothetical, in that they rarely, if ever, enter into the workings of an implemented KRR system. Second, the semantic models used in model theory for logical systems do not fit "natural" or perceptual categories at all.

3.2.1 Semantics as vaporware

Symbolic KRR systems are observer-level theories of agents' representation and reasoning capabilities, and as such should not be used as agent-level implementation vehicles, or only with extreme care. They can be used to describe, discuss, and hypothesize about agents' behavior among scientists (observers), but that is something very different from providing an actual agent with such capabilities. Analogously, we may describe the workings of a car engine using the language of differential equations, with an accompanying semantics.[6] One would not expect, however, to find an interpreter for differential equations in an engine upon opening it up, nor does it seem likely that we could build a working engine based on such an interpreter. Why then would we want to use a KRR language to implement a working agent (or agent's mind)? Granted, a car engine and a mind are very different things, at some level of description at least.

KRR systems are attempts at formalizing the way we intuitively and consciously perceive

[5]The proof of the stronger claim that *all* KRR formalisms can be translated into first-order logic is rather simple, if one considers only languages and inference mechanisms which are implemented or implementable on a Turing Machine (TM), and one further considers the TM-equivalence of first-order logic [Boolos & Jeffrey 1974]. I am not aware of any KRR formalisms which are not implementable on a TM, and I will disregard this possibility in the light of the Church-Turing thesis [Davis & Weyuker 1983].

[6]The preferred interpretation is one that relates the symbols used in the equations to such things as mass and velocity of objects, although one can doubt whether these are things "in the world" or just more abstractions derived from our perceptual experience (cf. [Heisenberg 1988], [Schrödinger 1988]).

the world to be, including the way we perceive our own mind to work, in order to endow an artificial agent with a similar understanding, and hopefully with similar capabilities to use that understanding in order to function intelligently in the world. As such it relies heavily on introspection and conscious awareness of the world and ourselves as a source of things to formalize. But in the course of studying perception on the one hand and the physical world on the other hand, we have come to realize that there is an enormous gap between the physical world as we now understand it to be and our conscious perception of that world. That gap is somehow bridged by our perceptual apparatus, and our knowledge of how that works is still spotty at best. Moreover, the perceptual apparatus is entirely "invisible" to the conscious mind,[7] to such an extent that even some scientist have proposed that we just perceive the world directly, without any intervening mechanism at all (e.g., [Luce 1954]). Strange as that may seem, it is a good indication of how convincing and "real" conscious awareness of things is, or seems to be. In AI, it has taken the concerted efforts in the field of computer vision to enable computers to see, to realize just how hard the problem is, and how well our brains manage to deceive the conscious part of us into thinking perception is something direct and easy. Even color vision, which this dissertation deals with some aspects of, and which seems so natural and effortless to us, is a far from understood problem of amazing complexity. In the light of all this, it seems limited at best to try to endow an artificial agent with intelligence by merely trying to formalize our conscious awareness of things and ignoring the very mechanisms which lead to such an awareness in the first place.

Although other kinds of models are emerging, some (like artificial neural networks) based more closely on our present understanding of how the brain works, we may feel that their level of description and operation is too low, that they therefore offer little hope for working models of mind, and thus that any viable approach has to encompass some kind of KRR capability.[8] Elsewhere we have argued for hybrid models of autonomous agents, comprising both KRR and other "lower level" mechanisms [Hexmoor et al. 1992, Hexmoor et al. 1993c, Hexmoor et al. 1993b, Lammens et al. 1994], and there are some signs that the neural network community is trying to re-incorporate results from "traditional" AI and

[7] Or "cognitively impenetrable", as [Pylyshyn 1981] calls it.

[8] One rather sobering observation about artificial neural networks concerns the size of the models, typically no more than a few hundred artificial neurons, compared to the estimated 10 billion neurons in the human brain, each of which may compute functions that are quite a bit more complex than the typical artificial neuron does. Even taking the difference in speed of the underlying technology into account, the complexity gap is huge.

KRR [Zeidenberg 1990]. If we accept the premise that using a KRR language is a valid approach to implementing (part of) an agent's mind, we should be aware of the unusual nature of trying to use an observer-level description as an agent-level implementation. One of the consequences, I believe, is that we have to treat the semantics of our descriptive language differently.

Ordinarily, it suffices that there exists a systematic interpretation (semantics), and that all observers share it (or share the preferred interpretation). We then hope that we can restrict the formal properties of the representation language in such a way that (1) we can express anything in the domain of interpretation in the formal language (representational adequacy) and (2) the interpretation of formal statements derived from others we consider to be true, by means of formal inference, does not disagree with the facts of the domain of interpretation, as we perceive them (inferential soundness). In addition, we may require that (3) anything we perceive to be true of the domain of interpretation can be derived from a set of basic premises by formal inference (inferential completeness).[9] Semantics does not enter into the inference process; it merely serves as a tool for theorists to convince themselves and each other that the formal system does what it is supposed to do. But although semantics has been effectively shut out from the inference process, it is eventually the yardstick with which the usefulness of models is measured. We hope that in turning the crank of the formal system, it will provide us with formal conclusions representing true statements about the domain of interpretation which we had not ourselves perceived before. It is a tool for *us* to understand more about the domain of interpretation.

If we want a KRR system to be part of an agent-level implementation of a mind, however, things are different. It is no longer the independent theorist who will use a semantic yardstick to measure the performance of the formal system and hope to learn more about the domain of interpretation by turning the formal crank. What we are after in implementing a mind is to give the artificial agent itself the same kind of "understanding" of the domain of interpretation as we have, to give it the same access to that domain as we do. In other words, the semantics of the model can no longer exist independently of the system itself; it has to be part and parcel of it. In terms of the subject matter of this dissertation, we do not want to find out more about color by building a formal system and turning its crank, but

[9]These definitions are rather informal; some more formal ones are discussed below.

rather we want an artificial agent to understand color the way we do, i.e., to *perceive* color the way we do. No amount of axioms and inferences in a formal system will ever give an agent that understanding if it cannot identify the referents of the terms that are supposed to be interpreted as representing colors, i.e., if it does not have an integral semantic model of color terms as part and parcel of its "mind".[10] Color perception in this case provides the required grounding for a set of terms of the KRR system. It constitutes an internal, referential semantic model of color terms, as I explained above. Even if we could provide a blind agent with enough formal representations (without the associated semantics) to not produce any statements about color which we would find in disagreement with our understanding of it, it could hardly be said to understand those statements if it has no way to identify or discriminate color in its environment at all, i.e., to make color play a role in its interaction with its environment.[11] This is, of course, what Steven Harnad has called the symbol grounding problem [Harnad 1990].

I conjecture that an ungrounded agent will never be able to adapt to its environment in an effective way, or to interact with its environment in a useful way, two things we can safely regard as prerequisites for intelligence. The reason it cannot do these things is that it is effectively cut off from its environment as long as its representations are ungrounded, i.e., as long as they are formal only. In that respect I consider the type of approach as exemplified by [Rapaport 1988] as insufficient, viz. that semantics is reducible to syntax. It seems to arise out of an exclusive preoccupation with language, and I believe one cannot deal with language without dealing with cognition in general, including perception. Language is the top of a mountain of cognition, and we cannot hope to arrive at it without climbing.

For a slightly different approach to the same problem, let's go back to the view of a KRR system as a formal language for a moment. A formal language $\mathcal{L}$ consists of an alphabet A, a syntax describing the legal strings of the language $S \subset A^*$, a set of axioms $\mathcal{A}$ that are given as true statements of the language, and a set of inference rules R that allow us

[10] Or as Cris Kobryn put it in a comp.ai electronic news article: "How does one verbally explain what the color blue is to someone who was born blind? The problem here is to explain a sensory experience (e.g. seeing 'blue') to someone lacking the corresponding sensory facility (e.g., vision)" (article `<200@kvasir.esosun.UUCP>`). Note that this formulation presupposes a person with a developed language capability, however, which makes the problem considerably less hard than in the case of an artificial agent.

[11] I do not consider producing or accepting natural language surface forms to constitute interaction; indeed, I would claim the blind agent I described cannot understand the meaning of the color words it might produce or accept.

to derive new statements from given ones [Manin 1977]. Any statement $s \in S$ that can be derived from $\mathcal{A}$ using the rules in R is a theorem, and the set of all derivable theorems plus the axioms is a theory. The inference rules R are *sound* iff given a consistent set $\mathcal{A}$, one cannot derive a contradiction, and the formal system is *complete* iff all statements that are true in any model (interpretation) of the language are theorems of the language.[12] The notion of representational adequacy described above is more difficult to formalize. Once we have defined a formal language $\mathcal{L} = (A, S, \mathcal{A}, R)$, we have once and for all defined a corresponding theory, which we may regard as being implicit in the definition and able to be made explicit by applying the inference rules to the axioms A and any previously derived theorems.[13] If we conceptualize the set A^* as a space with each string represented as a point (or a vector),[14] then the theory defined by the formal system is necessarily a subspace of that space. It is easy to see that we can change the shape and size of the theory-subspace by changing the set of axioms $\mathcal{A}$ or the set of inference rules R.

What the ungrounded symbolic approach to KRR, for instance [Rapaport 1988], is attempting to do, in my opinion, is to shape the theory-subspace such that it coincides with the space of what we as external observers would consider to be meaningful statements. It does this by introducing additional "meaning postulates" (axioms) or inference rules, whenever an undesirable result is produced. For example, when a natural language understanding (NLU) system happily produces sentences like "Young old Lucy is a male girl", additional meaning postulates are added to the underlying KRR system whose intended interpretation is that something cannot be young and old at the same time, nor male and a girl at the same time, and the inference rules used during the NL generation process are changed, if necessary, to take these constraints into account. Another example would be a meaning postulate that states that although a phrase like "reddish green" is syntactically well-formed, semantically it is not — or it is at least strange.[15] One hopes, then, that introducing enough of these meaning postulates and inference rules will eventually make

[12] Alternatively, one could say the rules are sound *iff* every derivable *wff* s is true, and the rules are complete *iff* $(\forall s) \vdash s \vee \vdash \neg s$.

[13] Strictly speaking, this applies to monotonic inference only. Since it has no bearing on the gist of the argument, I will disregard non-monotonic inference for now.

[14] Details of the construction of such a space are left as an exercise for the reader.

[15] In an informal inquiry I did not find any natural language that uses such a term on a regular basis, and I speculate (with many other researchers) that there is no such language, for a reason which will become apparent later.

the theory-space converge onto the space of meaningful statements, i.e. meaningful for the human observer.

I will refer to this as the *pruning problem*, since it amounts to pruning down the tree of theorems of the formal system. This, I believe, is in essence what [Rapaport 1988] refers to as syntactic semantics, although he does not state it in these terms. The important thing to notice is that "meaning", as we humans understand it, does not enter into the picture, except as an external yardstick used by the observer to measure how well the formal system is performing. The formal system itself is blissfully unaware of any meaning. I believe that although in principle it is possible to constrain a formal system enough in this way, there is no indication that we are anywhere near that goal, or making substantial progress. Probably the largest effort in this respect is the Cyc project at MCC [Lenat 1990], the usefulness and robustness of which for practical applications remains to be shown. One of the main problems with this approach seems to be that it is not clear where such postulates should come from (whether there is a systematic "discovery procedure" for them), nor how many one needs, nor indeed whether their number is finite at all. The Cyc project seems to have resorted to a rather haphazard way of collecting meaning postulates (and inference rules), essentially by having team members read anything they can get their hands on and figuring out which postulates have to be added to make the system work properly [Lenat 1990].[16]

What is the alternative? If we want to use some kind of symbolic knowledge representation formalism, we will have to make sure that it is grounded in perception and action, or embodied. Basically that means making sure that the agent using the KRR formalism is able to identify and interact with the referents of a set of distinguished terms, the directly grounded ones (cf. [Harnad 1990]). Other terms and constructs of the KR language, the indirectly grounded ones, will have to derive their meaning from this set of directly grounded terms. I will make these concepts more concrete in what follows. Indeed, this entire dissertation can be seen as a case study of the grounding of a particular set of terms: basic color terms (Section 4.2). As sketched in Section 2.3, and explained in detail in Chapter 5, this approach makes a referential semantic model an integral part of the agent's machinery,

[16]This is not meant to denigrate the efforts of the Cyc team. I believe that they are doing an excellent job within the parameters of their project. I merely mean to point out some fundamental problems with the approach itself, which does not reflect on the integrity of the Cyc team members or the value of their work.

rather than something that exists only on paper or in the mind of the observer. Not only can such an agent identify and interact with the referents of (a subset of) the symbols it uses, a condition I cited above for understanding, but it also provides us with a different approach to the pruning problem I mentioned before, one that looks more promising to me than the purely syntactic approach. We can prune the tree of theorems from a semantic point of view, rather than a syntactic one. That answers the question where meaning postulates come from, viz., from the constraints that are inherent in the perceptual and motor mechanisms that ground the symbols of the KR language. Embodiment really matters; one cannot study intelligence or cognition in the abstract. This approach may also provide an answer to the question how many meaning postulates are needed. I propose that they are generated on the fly, as needed, by using embodied reasoning mechanisms as well as embodied representations.

To make all of this more concrete, let's consider the grounding of basic color terms. As I will describe in section 4.3, human color perception is thought to be organized in an opponent fashion, red and green being one opponent pair, and blue and yellow another (Figure 3.1). That means that opponent color percepts cancel each other, so that we can well perceive yellowish green or reddish blue, but not reddish green or yellowish blue (the mutual cancelation results in an achromatic white or grey perception). If our model of color perception and color naming (i.e., our semantic model of basic color terms) uses an analog color space representation that conforms to that organization, our agent can employ an analog style "reasoning" process to think and speak about color, which basically amounts to mental imagery, or *envisioning* the referents of terms in the perceptual space they derive their meaning from.

Let's assume for the sake of simplicity that the semantics (meaning) of a basic color term (e.g. "red") is a function from that term to a point in the color space and that the semantics of a compound color term (e.g. "yellowish green") is compositional, i.e. it is a function of the meanings of its constituent terms.[17] In particular, let's assume that the point which is the image of the compound term under the semantic function is halfway between the points that are the images of the component terms.[18] The meaning (which I take to be just

[17] I will relax the point assumption in Chapter 5. This example also ignores possible differences in the relative saturation and brightness of colors, but without loss of generality.

[18] Using some metric on the color space, Euclidean or other; cf. [Shepard 1987].

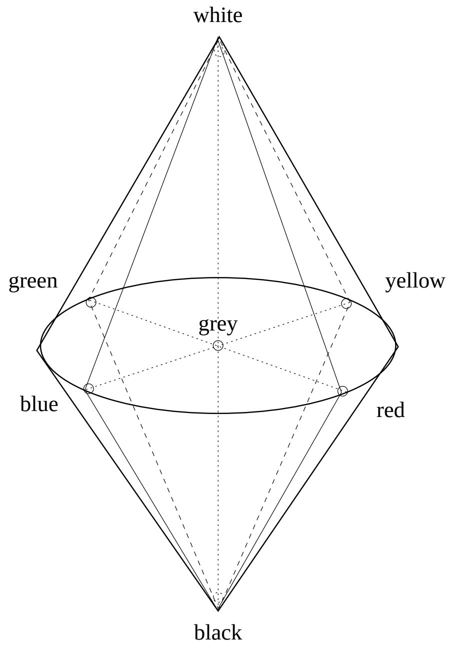


Figure 3.1: Schematic representation of a perceptual opponent color space (see text). The coordinate system is cylindrical: the vertical axis, going from black to white, represents perceived brightness and is also known as the grey axis; hue (what we commonly refer to as "color") and saturation (the "purity" of a color) are represented in polar coordinates, with the origin on the grey axis, hue as the angle between a given point and a distinguished point (usually the location of red), and saturation as the distance from the grey axis.

the image of the term under the semantic function) of "yellowish green" is then a point in the color space corresponding to the color one perceives when superimposing yellow and green lights, i.e., a yellowish green or greenish yellow color,[19] and the same for the meaning of "red-green". However, in the latter case, the resulting percept is achromatic (white or gray), i.e., not a color in the common sense.[20] Hence, there is no need for a syntactic meaning postulate specifying that there is something semantically odd about "red-green" as a color term but not about "yellowish green": it is immediately obvious from *envisioning* the meanings of these terms.

So, in a sense, having a semantic model as part of an agent's mind provides meaning postulates, but they do not have to be represented explicitly. How many such postulates are there? Given that the perceptual space is represented in an analog fashion, the number is infinite for all practical purposes, up to the limit of resolution of the perceptual space.[21] The semantics of terms used in KRR is now no longer something extraneous to the KRR system, but provides essential constraints on the reasoning process. It is not yet clear to me *how* these constraints should affect the reasoning process, only that they do. An agent with grounded color terms will not likely attempt to paint its office walls reddish green, for instance.

3.2.2 Categorial misfits

There is a more fundamental problem with Tarski-style semantics, with regard to its potential use for KRR systems. Given the discrete symbolic nature of logic, Tarski-style semantic models have always presupposed an equally discrete domain of interpretation, consisting of individuals that are the images of individual constants of the language under the interpretation function, sets of such individuals corresponding to predicates of the language, etc. It does not matter whether the domain of interpretation is taken to be the real world, a possible world, objects of thought, or something else; the discreteness assumption is universal. As I discussed in Section 1.1, the categories of human perception (and, by extension, of human cognition, since I firmly believe that perception is one of the foundations of cognition) are

[19]The limitations of language become very apparent here.

[20]The term *achromatic color* sounds somewhat contradictory to most ears, but is used as a technical term in the color vision literature.

[21]There are many other phenomena related to the graded nature of color categories that could be expressed with similar meaning postulates, but it would lead us too far to discuss those now.

not discrete. This misfit between logical categories and perceptual categories is one of the basic problems in trying to use logic-inspired KRR formalisms as agent-level mechanisms, I believe (see also [Lakoff 1987], among others). If we want to make semantic models part of the agent's machinery, as described above, they will have to take the non-discrete nature of natural categories into account. We find support for this point of view in a perhaps unexpected place, viz. a textbook on mathematical logic:

> Most natural and artificial languages are characteristically discrete and linear (one-dimensional). On the other hand, our perception of the external world is not felt by us to be either discrete or linear, although these characteristics are observed on the level of physiological mechanisms (coding by impulses in the nervous system). [...] The human brain clearly uses both principles. The perception of images as a whole, along with emotions, are more closely connected with nonlinear and non-discrete processes — perhaps of a wave nature. [Manin 1977, p. 18]

My dissertation work is a case study of how to map a continuous world onto a set of discrete symbols, viz., basic color terms. The essential characteristics of the model in this respect are that it contains an analog representation which functions as a perceptual or psychological space (cf. [Shepard 1987]) causally connected to the outside world via sensors, and a mechanism that relates regions in this space to individual terms and associated "typicality" or "goodness" values (Section 2.3). The regions representing the extensions of the terms may overlap or not, and the characteristic functions associated with the terms (if we conceive of them as something like fuzzy sets) are continuous-valued. This, I believe, represents a much more realistic attempt to characterize the semantics of a set of terms than any discrete model can hope to achieve. As logicians, we may not like the inherent fuzziness or "scruffiness" of a model with continuous numerical components, but as students of cognition we are forced to accept and deal with it.[22]

[22]Artificial neural networks have also embraced the notion of continuous-valued quantities rather than discrete symbols, and I believe this is partly responsible for their success in modeling some aspects of human perception and cognition. My own work is hybrid in nature, using both discrete symbols and continuous numerical quantities. Recent neural network research has also attempted to re-introduce symbolic representations [Zeidenberg 1990].

3.3 Some Pseudo-Philosophical Notes

The discrete symbolic nature of both logic and natural language (at least in its surface manifestations) has led many to propose discrete semantic models, as described above. If the referents of symbols or words are taken to be in the world, this leads to incompatibilities between the models and how we perceive reality to be. These incompatibilities are reflected in philosophical problems like the question of indiscernibles: "At sunset, where does the sky change from orange to yellow, yellow to ...to blue, blue to dark?". Only if one assumes that color is a discrete-valued object property would one expect it to change abruptly from one value to another at a particular place. If one assumes that color is a continuous-valued perceptual property, the question becomes ill-posed.

Another pervasive assumption in the domain of color is that color is a property of objects in the world, and our visual system perceives that property. In logical models, this would typically be represented as a predicate, so that, e.g., $Red(x)$ is taken to mean that object x has the property of being red, and in formal semantic models these predicates would map into sets of red objects. As I discussed in Section 2.1, color is not a property of objects but a response of the visual system to certain physical stimuli, and this response is only indirectly related to certain object properties and can vary in space and time with unchanging objects. In the semantic model I propose, a term like "red" maps into a region of the perceptual color space; i.e., redness is a perceptual phenomenon, not a phenomenon in the outside world. It is then easy to see that a question like "What is the color of the sky at night?" is ill-posed as well. The sky does not have any color, either during the day or at night.[23]

In logical and philosophical writings on semantics, one often runs into strange denizens like unicorns and round squares, and these poor creatures are habitually blamed for the existence of possible worlds, accessibility relations, objects of thought, and the like. The seldom explicated assumption underlying all this is that since the words exist and we can use them in thought or natural language, they must refer to some object somewhere. But where could unicorns and round squares live, other than in possible worlds or as mere objects of thought? The assumptions of traditional model theory, as described above, force one into accepting the strangest things. I believe we should abandon or review those

[23]Along the same lines, one might doubt whether there is any sky to begin with, but I will not make that case.

assumptions, rather than force the universe to conform to them. Along the lines of what [Harnad 1990] and others have proposed, I believe that meaning derives from perception and interaction with the real world, not from possible worlds or objects of thought.[24] In order to understand what a unicorn is, we first have to know what horses and horns are. If we want to ponder round squares, we'll have to know what circles and squares are first. Such knowledge comes from perception and interaction with the world, and we can consider the corresponding symbols or words to be "directly grounded". Other symbols or words, like abstracta or "impossible" objects, which we can consider to be "indirectly grounded", can only be meaningful by virtue of being related to directly grounded ones in a systematic way, and don't have to refer to strange creatures in hypothetical modes of existence. I therefore consider the most urgent task of KRR to be to investigate how symbols can be directly grounded in perception and interaction; we can worry about indirectly grounded ones later. We have to learn how to crawl before learning how to walk.

The work on color perception and color naming presented in this dissertation is an investigation into the grounding of just a small set of terms, and already the complexities are considerable. Under those circumstances, it is useless to worry about unicorns and round squares, or general purpose KRR and natural language processing, in my opinion. Or as [G. Edelman 1992] puts it while discussing Lakoff's cognitive grammar:

> The important thing to grasp is that idealized cognitive models involve conceptual embodiment and that conceptual embodiment occurs through bodily activities *prior to language*. [p. 246, italics in original]

I believe this is essentially true, and in order to arrive at true language competence for an artificial agent, it will have to go through at least some prior stages of conceptual embodiment first.

My work may also shed some light on the empiricism vs. innateness debate in cognitive science (whether "primitive" concepts or meanings are acquired through experience or inborn), with respect to the origin of "primitive" concepts. In my model, (color) concepts are grounded in the *perception* of the world, rather than in the world itself, but perception is causally connected to the world. The *mechanisms* for abstracting perceptual data into categories, in our case the neurophysiology of color vision, are to a considerable extent ge-

[24] The latter proposal is entirely circular, in my opinion.

netically determined, i.e. "innate". That does not mean that a child is born with complete color concepts, waiting to be "activated" by interaction with the world, except in a very abstract sense. Since the basic mechanism (the neuro-anatomy and physiology) is innate,[25] any normal child will develop the same or comparable color concepts (categories) when interacting with the same kind of environment (which may include cultural and linguistic factors). But no matter how innate the basic mechanism is, no concepts (categories) will actually develop without external stimuli. Also, different physiology leads to different "innate" concepts, so one might consider cats, for instance, to have different color concepts than we do, while they live in approximately the same environment, something which would be hard to explain from a purely empiricist point of view. We can regard the innate vs. empiricist position as duals; both are right in some respect, but both also miss a part of the picture as a result of methodological dogmatism. To caricature the respective positions, one might say that empiricists have overlooked the fact that human physiology is just as much part of the world as trees and stones are, and it may (and in fact does) influence concept formation, too. Nativists, on the other hand, have failed to recognize that interaction with the world is a prerequisite for any concept formation at all, and that there is a systematic correspondence between some properties of the world and some properties of categories, albeit mediated (and perhaps transformed) by perception.

[25] Even this is a simplification. As [Maturana & Varela 1987], [G. Edelman 1992] and [Zeki 1993] point out, for instance, there is considerable plasticity in even the adult brain. But for the purpose of our discussion that is not immediately relevant.

Chapter 4

Related Work

In this chapter, I review the relevant literature. Since the work presented in this dissertation draws upon a variety of disciplines (artificial intelligence, computer vision, psychology, psychophysics, biophysics, neurophysiology, anthropological linguistics, and philosophy), it is impossible to exhaustively review the color-related literature in all of them. I have therefore limited myself to a discussion of the literature that I consider most pertinent to the topic of my dissertation, and I do not claim exhaustiveness or suitability for any other purpose.

4.1 Embodiment and Symbol Grounding

In [Lakoff 1987], *conceptual embodiment* is described as the idea that the properties of certain categories are a consequence of the nature of human biological capacities and of the experience of functioning in a physical and social environment. It is contrasted with the "objectivist" idea that concepts exist independently of the bodily nature of any thinking beings and independently of their experience (p. 12). Several sources from the cognitive science literature at large are cited in [Lakoff 1987] as having contributed to the development of this idea. My definition of embodiment in the context of autonomous agents is more restricted, but more precise (Section 3.1).

According to [Harnad 1990], symbolic representations must be grounded bottom-up in non-symbolic representations of two kinds:

- *iconic* representations, which are "analogs of the proximal sensory projections of distal objects and events", and

- *categorical* representations, which are "learned and innate feature-detectors that pick out invariant features of object and event categories".

Symbolic representations, according to Harnad, are grounded in these two levels of elementary symbols. Symbolic representations consist of symbol strings describing category membership relations, e.g., *An X is a Y that is Z.* An example would be *"A zebra is a horse that is striped"*.

Part of my model of color perception and color naming (Chapter 5) is a color space defined by several dimensions derived from or modeled after the activations of the cone photoreceptors in the human retina. This is an analog (continuous) transform of the sensory "projection" of the radiation reflected off the "distal" surface of the object being imaged; therefore, it qualifies as an iconic representation. A second part of the model is a mapping from each point in the color space to one or more pairs consisting of a color term and a "confidence" or "goodness" measure. The codomain of this mapping qualifies as a categorical representation, in Harnad's framework. My model does not involve symbolic representations of the kind Harnad envisions, as it is limited to the level of terms only, not sentences.

Harnad lists five tasks a cognitive theory has to explain: discrimination, identification, manipulation, describing, and responding to descriptions of objects, events, and states of affairs in the world. A CLR (Color Labeling Robot) is in principle able to perform all these tasks:

- Discrimination: any two stimuli that result in different coordinates in the model's color space are discriminable for the CLR

- Identification: the CLR is able to identify (categorize) any stimulus as belonging to one (or more) of the categories represented in the codomain of $\mathcal{N}$, and give a confidence or "goodness" rating.

- Manipulation: an extended CLR (equipped with manipulators) that is capable of acting in its environment, in addition to perceiving it, is able to manipulate objects based on their perceived color properties.

- Describing: since the model only extends to the level of terms, describing is limited to naming the colors the agent perceives in the world, which the CLR is able to do.
- Responding to descriptions: the CLR responds to a description (color name) by pointing out an example fitting the description (a colored object in its environment).

4.2 Basic Color Terms

The notion of *basic color terms* originates in the anthropological and linguistic work of Brent Berlin and Paul Kay in the late sixties [Berlin & Kay 1969]. They studied color naming behavior with native speakers of a variety of languages, and the existing literature on the subject, and recorded two main findings:

1. There are substantial universal constraints on the shape of basic color lexicons.
2. Basic color lexicons change over time by adding basic color terms in a highly constrained, though not mechanically predictable, manner.

The research was carried out against a background of extreme linguistic relativism, also known as the Sapir-Whorf hypothesis [Kay & Kempton 1984], which holds that each language performs the coding of experience into language in a unique and arbitrary manner and that there are no semantic universals (principles of meaning that hold across all languages) in principle.

Basic color terms are defined as having the following characteristics:

1. They are "monolexemic"; i.e., their meaning is not predictable from the meaning of their parts (for English, the following do not qualify: *bluish, lemon-colored*, but *blue* and *yellow* do).
2. Their "signification" is not included in that of any other color term (for English, *crimson* or *scarlet* do not qualify, as they are both included in *red*, but *red* does).
3. Their application is not restricted to a narrow class of objects (for English, *blond* does not qualify, but *brown* does).
4. They are psychologically salient for informants, which is apparent from occurring at the beginning of elicited lists of color terms, stability of reference across informants

and occasions of use, and occurrence in idiolects of all informants (for English, *red, green, blue, yellow* are good candidates, but *chartreuse* is not).

Some additional criteria are provided for borderline cases.

A set of 329 color chips mounted on a piece of cardboard was used as stimulus material. The color chips were selected using the Munsell color system (Section 4.4), and represented 40 equally spaced hues $\times$ 8 degrees of brightness, plus 9 neutral hues ranging from white through grey to black (Figure 4.1). A constant light source was used to illuminate the

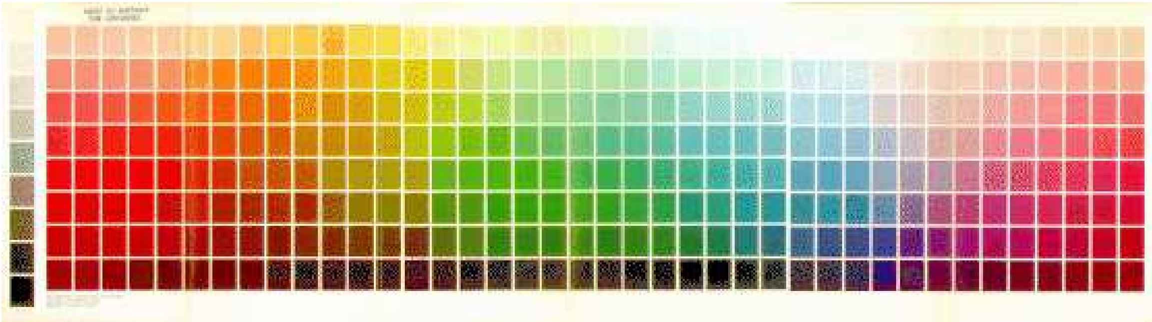


Figure 4.1: Reproduction of the color stimuli used by Berlin and Kay. In the actual experiments, the color chips were mounted on a grey background.

chips. After the basic color terms of the language in question had been elicited, subjects were instructed to indicate the *focal point* (best example) of each basic color category and its *outer boundary* in the set of color chips.

Berlin and Kay found that the *foci* of basic color terms are similar across totally unrelated languages, and that they cluster into discrete, contiguous areas in the color space. The *boundaries* between color categories, on the other hand, were found to be variable across languages and even for repeated trials with the same informants. They speculate that unless the effect is due to the experimental procedure,

> it is possible that the brain's primary storage procedure for the physical reference of color categories is concerned with points (or very small volumes) of the color solid rather than extended volumes. Secondary processes, of lower salience and inter-subjective homogeneity, would then account for the extensions of reference to points in the color solid not equivalent to (or included in) the focus. (p. 13)

This suggestion is taken up almost 20 years later by [Lakoff 1987], who considers color to be an example of a "generative cognitive category", one that consists of a focus (or foci) combined with a "complex cognitive mechanism" that generates other members of the category from the focus in a consistent but not a priori predictable manner.

Berlin and Kay also found that although different languages encode different numbers of basic color categories in their vocabularies, a total universal inventory of exactly eleven basic color categories exists, from which the eleven or fewer basic color terms of any given language are always drawn. The English terms corresponding to the eleven categories are

white, black, red, green, yellow, blue, brown, purple, pink, orange, grey.

In addition, they found that if a language encodes fewer than eleven basic color categories, then there exist strict limitations on which categories it will encode. The distributional restrictions across languages can be summarized as a sequence of "evolutionary stages":

1. All languages have terms for *white, black* (or more accurately, for light-warm and dark-cool colors).

2. If a language encodes 3 categories, it contains a term for *red*, in addition to the terms encoded in the first stage.

3. If a language encodes 4 categories, it contains a term for *green* **or** a term for *yellow*, in addition to the terms encoded in the previous stage.

4. If a language encodes 5 categories, it contains terms for *green* **and** *yellow*, in addition to the terms encoded in the previous stage.

5. If a language encodes 6 categories, it contains a term for *blue*, in addition to the terms encoded in the previous stage.

6. If a language encodes 7 categories, it contains a term for *brown*, in addition to the terms encoded in the previous stage.

7. If a language encodes 8 or more categories, it contains terms for *purple, pink, orange, grey*, or some combination of these, in addition to the terms encoded in the previous stage.

This sequence of stages defines a partial order on the set of basic color categories with six equivalence classes, which may be represented as follows:

$$\begin{bmatrix} white \\ black \end{bmatrix} < \begin{bmatrix} red \end{bmatrix} < \begin{bmatrix} green \\ yellow \end{bmatrix} < \begin{bmatrix} blue \end{bmatrix} < \begin{bmatrix} brown \end{bmatrix} < \begin{bmatrix} purple \\ pink \\ orange \\ grey \end{bmatrix}$$

where $a < b$ means that equivalence class a is present in every language in which any element of equivalence class b is present.

Berlin and Kay interpret this sequence as reflecting both distributional facts and a chronological order of lexical encoding of basic color terms in each language. This chronological order is in turn interpreted as a sequence of evolutionary stages. More recent work has revised the evolutionary sequence somewhat, e.g. [Kay et al. 1991]. Although Berlin and Kay speculate that there is a correlation between the general cultural and technological complexity of societies and the complexity of the color vocabulary (Figure 4.2), they also admit that

> [...] the particular order in which color foci universally became encoded in individual lexicons [...] is a difficult problem which is only vaguely understood at this time. (p. 17)

One may wonder why English, of all languages, turns out to have the most (11) basic color categories, and thus sits at the top of the complexity hierarchy with respect to color vocabulary. Other than reflecting the "general cultural and technological complexity of society", there might be an unintended bias in the experiment or the interpretation of the data. For example, it might be difficult to recognize a basic color term in another language for which there is no corresponding basic color term in English. Nothing in the data seems to indicate that the 11 terms of English is a theoretical maximum. Indeed, Cairo [Cairo 1977] predicts some additional potential basic color terms which have not been lexicalized in English (yet?), e.g. something akin to "khaki".

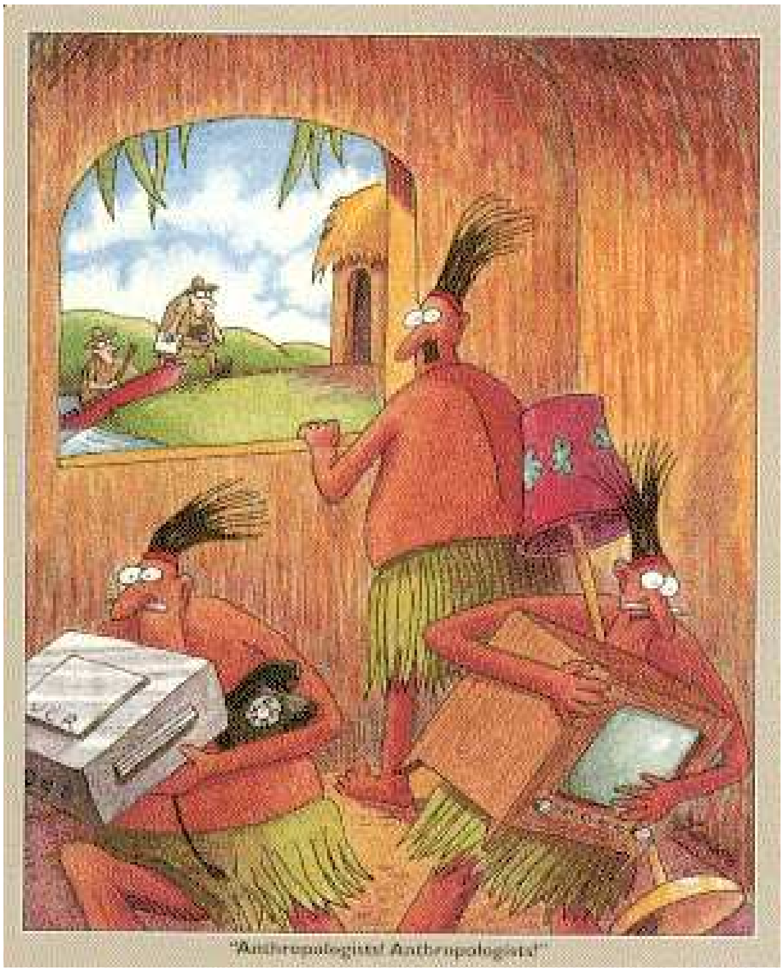


Figure 4.2: Anthropological fieldwork (Garry Larson, The Far Side).

Berlin and Kay seem to be concerned with issues of symbol grounding and embodiment too, although they don't use those terms of course (their work predates the symbol grounding and embodiment work considerably):

> The study of the biological foundations of the most peculiarly and exclusively human set of behavioral abilities — language — is just beginning [...], but sufficient evidence has already accumulated to show that such connections must exist for the linguistic realms of syntax and phonology. The findings reported here concerning the universality and evolution of basic color lexicon [sic] suggest that such connections are also to be found in the realm of semantics. (p. 110)

The research presented in this dissertation can be seen as a formal investigation into such connections.

Berlin and Kay's findings have been corroborated in numerous other studies since (see the bibliography in [Berlin & Kay 1969]). Boynton, for example, reports that other research has shown that basic color terms are listed first and used more reliably, with greater consensus and shorter response times than any other color terms [Boynton 1990, p. 240]. They also translate easily between languages and are commonly learned by the age of five, at which time very few non-basic color terms are used. None of these special attributes apply to non-basic color terms. Although Berlin and Kay discount the possibility that the boundary effect is due to their experimental procedure, Kay and McDaniel [Kay & McDaniel 1978] seem to be less convinced, as I will discuss below. Cairo [Cairo 1977] also has some doubts about their experimental procedure.

4.3 Neurophysiology of Color Vision

There is a considerable body of literature on the neurophysiology of color vision, some references to which can be found in [Boynton 1979], [Boynton 1990], [Dow 1990], and [Hubel & Livingstone 1990], for instance.

The work of De Valois and his colleagues is some of the best known in this area [De Valois et al. 1966, De Valois & Jacobs 1968, De Valois & De Valois 1975]. They made extracellular electrical recordings of LGN[1] cell responses in the Macaque monkey, while the visual system was stimulated with monochromatic light. They cite various kinds of evidence that indicate that the Macaque's vision system is very similar to that of humans, and conclude that it is safe to generalize the experimental results to human neurophysiology. They identify six types of cells, based on a statistical analysis of the responses of a large population of cells. Four of these cell types display what they call spectrally opponent responses, and two types display spectrally non-opponent responses. The opponent cell types respond with inhibition to one part of the spectrum, and with excitation to another part. The non-opponent cell types respond with either inhibition or excitation to the entire spectrum. One can further distinguish three pairs within the cell types; within each pair, the members

[1] Lateral Geniculate Nucleus, a body of cells roughly halfway between the retina and the primary visual cortex V1.

display “mirror imaged” responses: when one type displays inhibition in response to some part of the spectrum, the other displays excitation, and vice versa. De Valois et al. note that the response functions of the six cell types seem to agree well with various psychophysical findings on color perception, and are reminiscent of the opponent color response functions of [Hurvich & Jameson 1957]. I will discuss these findings in greater detail in Chapter 5.

The existence of color opponent cells in the (Macaque) LGN has been confirmed in numerous studies, for instance [Derrington et al. 1984]. The latter study finds groups of parvo-cellular (P) cells that get opposed but not equally balanced inputs from only “red” and “green” cones, which they refer to as “R-G” cells, and different groups of P-cells that receive inputs from “blue” cones almost equally opposed by some (varying) combined input from “red” and “green” cones, which they refer to as “B-(R&G)” cells. In the magno-cellular (M) layer of the LGN they find cells that are both spatially and chromatically opponent, where the “red” and “green” cones contribute differentially to the center and surround of their receptive fields. Some of these cells have inputs from “blue” cones as well.

The work presented in [Hubel & Livingstone 1990] discusses more LGN cell types than [De Valois et al. 1966] does, in particular, the different characteristics of parvo-cellular vs. magno-cellular responses. It also uses a different classification scheme, which has become widely accepted after the work of De Valois et al. was published. Some of these cell types are more sensitive to luminance contrast, others to chromatic contrast, and there are various kinds of receptive field organizations. These characteristics are not neatly separable, and it seems that for any given response dimension (like luminance or chromatic contrast) there is a range of differently sensitive cells. This organization is reminiscent of the orientation columns in the cortex that consist of cells tuned to a particular direction of edges in their receptive field, with the range of possible directions sampled at approximately regular intervals by differently sensitive cell types [Dow 1990]. In addition, there are differences in temporal response characteristics across cell types. The authors report similar response characteristics and ranges for cortical cells, in addition to the presence of motion-selective cells and other types. They also present some further evidence that the Macaque's visual system is nearly identical to the human one in terms of color perception and other dimensions.

There are no real spectral sensitivity measurements in [Hubel & Livingstone 1990], as there are in [De Valois et al. 1966], and the kind of stimuli used is different. For the purpose of classifying different cell types in detail, and tracing the pathways between the LGN and the visual cortex, the kind of stimuli used by [De Valois et al. 1966] may not be the most appropriate, but for the purpose of measuring the spectral response of groups of similar cells, they probably are. I will base what follows on the results reported in [De Valois et al. 1966], while realizing that this isolates a particular response dimension in a way that may not be fully supported by neurophysiological data. I feel this simplification is justified though, at least at this stage of modeling, since my work attempts to deal with color only, as a self-contained (set of) perceptual variable(s). It is in this context that I interpret Boynton's warning:

> The facile identification of signals recorded from cells in the LGN with various aspects of color appearance, which is now deeply rooted in many introductory textbooks, is both premature and misleading. [Boynton 1990, p. 228]

I will use these LGN recordings as a starting point to construct a color space, which in turn will give rise to measures that I will relate to color appearance, but this is by no means a "facile identification".

4.4 Color Models

A "color model", or "color space", or "color order system", is a way to organize the set of possible human color percepts in some systematic way. Four families of color models can be distinguished in the color vision literature at large (see, e.g., [Boynton 1979], [Wyszecki & Stiles 1982], and [Boynton 1990] for some references, and [Brown 1982] for an excellent overview of color models and color theory). I use these categories merely as an organizational tool; they are not entirely mutually exclusive. The four categories are:

1. Physiologically inspired models using 3 primaries, based on the three types of cones in the human retina, starting with Hering's theory of color vision, e.g., the familiar RGB models used in computer graphics hardware.

2. Colorimetric color models, based on physical measurement of spectral reflectance, usually using three primary color filters and some kind of photometer, e.g., the CIE chromaticity diagram.

3. Opponent models, based on perception experiments, using pairwise opponent primary colors (yellow and blue, red and green), starting with the Young-Helmholtz theory, e.g., [Hurvich & Jameson 1957].

4. Psychological and psychophysical color models based on the appearance of colors to human observers, with data derived either in an impressionistic way (e.g., the Munsell and Ostwald color models) or in an experimental way (e.g., the Hue-Saturation-Brightness (HSB) family of color models, or the OSA uniform color space).

I will discuss some typical representatives of each category. Mathematical derivations (where available) of a collection of color models can be found in Appendix B.

4.4.1 The Munsell and Ostwald color models

The Munsell and Ostwald color order systems represent some of the earliest attempts at systematically organizing color percepts into a space [Munsell 1946, Birren 1969a, Birren 1969b, Boynton 1979, Meyer & Greenberg 1980, Wyszecki & Stiles 1982, Rogers 1985]. Both are defined as comparative references for artists and others to use, in a fairly impressionistic way (based on subjective observation rather than on direct measurements or controlled perceptual experiments). As such, they are still in use today, especially the Munsell color order system. The Munsell system has been and continues to some extent to be used as a standard in industry, notwithstanding attempts to introduce colorimetric models to replace it. The Munsell system was also used in [Berlin & Kay 1969], as described in Section 4.2. Both the Munsell and Ostwald systems are based on reflective (subtractive) color samples. Although quantitative transforms to other color spaces have been defined, they are not typically used in computer vision and computer graphics work, which deals with additive color. The general shape of both systems (using cylindrical coordinates, with three dimensions corresponding roughly to the perceptual variables brightness (vertical), hue (angular displacement from a reference color), and saturation (distance from the central axis)) is preserved in many color models, including the ones more quantitative in nature.

4.4.2 CIE chromaticity and related models

RGB color models as described in the next section rest on a scientific foundation that has come about largely under the auspices of the Commission Internationale de l'Éclairage (CIE), or the International Lighting Committee, a Paris-based standards organization. Early in this century, this organization sponsored research into color perception which lead to a class of widely used mathematical models [Wyszecki & Stiles 1982]. The basis for all of these models is a number of color-matching experiments, where an observer judges whether two parts of a visual stimulus match in appearance, i.e., look identical or not (Figure 4.3). By varying the composition of the light projected onto either part of the field

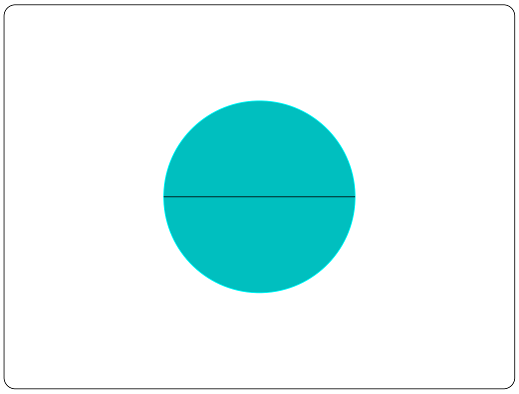


Figure 4.3: Color matching: light from two different sources is projected onto the top and bottom half of a circular field of a projection screen (or other device such as a CRT), and observers are asked to judge whether the halves match in color. The size and shape of the field may be varied, as well as the composition of the light sources.

of view, researchers can investigate properties of human color vision. For instance, it has been found that light of almost any spectral composition can be matched by mixtures of only three suitably chosen monochromatic primaries (light of a single wavelength), which is the principle behind color TV, as explained in the next section. By repeating this type of experiment with many different observers and averaging the results, and measuring the spectral composition and power of each of the light sources, the CIE has defined a number of so-called *standard observer color matching functions*. Figure 4.4 shows the color matching functions for a particular choice of monochromatic primaries with an approximately red, green, and blue appearance. Assuming that the human visual system behaves linearly,

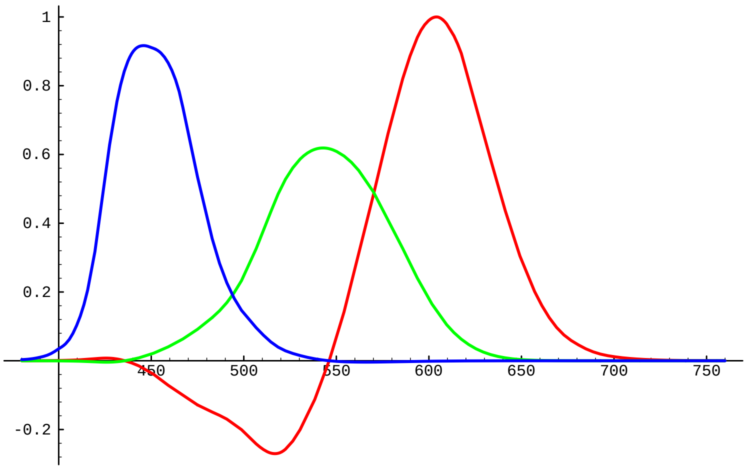


Figure 4.4: The CIE 1931 standard observer color matching functions for the set of pure monochromatic primaries (positive maxima left to right) 435.8 nm (blue), 546.1 nm (green), and 700 nm (red). On the Y axis: relative amount needed of each primary to match a monochromatic stimulus of the wavelength indicated on the X axis. Negative numbers require that the primary in question be added to the opposite side of the circular field (i.e., to the original stimulus to be matched) for both halves to match. The energies of the primaries are scaled such that a mixture of equal amounts of all three results in an achromatic white perception. After [McIlwain & Dean 1956, p. 48].

the CIE then went on to define the standard observer color matching functions in terms of so-called *virtual primaries*. This amounts to a linear transformation such that the color matching functions are all positive, which is desirable for practical applications. The resulting primaries cannot be physically realized, however. The result is usually referred to as the CIE 1931 standard observer color matching functions, and the individual functions are labeled $\overline{x}$, $\overline{y}$, and $\overline{z}$. These functions are also chosen such that $\overline{y}$ is proportional to the human photopic luminosity function, which is an experimentally determined measure of the perceived brightness of monochromatic light of different wavelengths (Figure 4.5). These functions are the basis for most quantitative work in color science to date [Wyszecki & Stiles

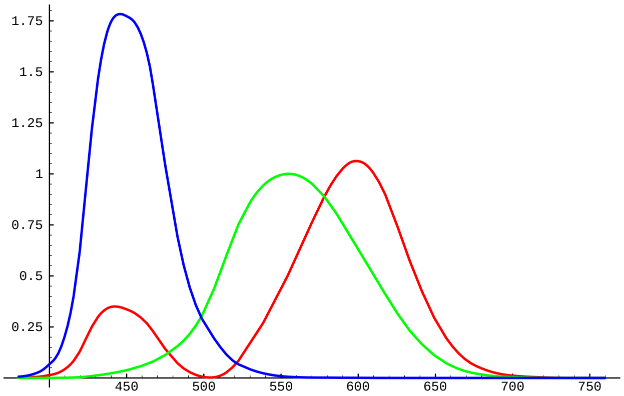


Figure 4.5: CIE 1931 standard observer color matching functions for virtual primaries (see text). Positive extrema from left to right: $\overline{z}$, $\overline{y}$, and $\overline{x}$.

1982], even though there have been several revisions since their original publication.[2] The color TV spectral sensitivity functions presented in the next section are linear transforms of the CIE functions (they are also the functions I used in Section 2.1). According to the theory, the color matching functions are linear transforms of the actual spectral sensitivity functions of the (average) human cone photoreceptors (Section 4.4.3). At the time of publication of the CIE functions, the cone spectral sensitivities were not known yet, but research done since then has shown good agreement with the predictions [Boynton 1979, Wyszecki & Stiles 1982, Boynton 1990].

If we know the spectral composition of a stimulus $E(\lambda)$, we can now determine its *chromaticity coordinates* as follows (see also Section 2.1). First, we calculate the *tristimulus values* X, Y, and Z:

$$X = \int E(\lambda)\overline{x}(\lambda)d\lambda \quad (4.1)$$

$$Y = \int E(\lambda)\overline{y}(\lambda)d\lambda \quad (4.2)$$

$$Z = \int E(\lambda)\overline{z}(\lambda)d\lambda \quad (4.3)$$

[2] For instance, the 1931 functions were measured for a 5 degree field of view, and in 1964 the CIE published similar functions for a 10 degree field of view. Other revisions have been proposed by Judd, Vos, & Walraven, and others [Wyszecki & Stiles 1982].

Next, we calculate the *chromaticity coordinates*:

$$x = \frac{X}{X+Y+Z} \tag{4.4}$$

$$y = \frac{Y}{X+Y+Z} \tag{4.5}$$

$$z = \frac{Z}{X+Y+Z} \tag{4.6}$$

Since the chromaticity coordinates are normalized, we lose all intensity information, but all colors are otherwise representable in this form. Usually the coordinates are plotted as a parametric x–y plot, with z implicit as $1-(x+y)$. Such a diagram is known as a *chromaticity diagram* (Figure 4.6, left). The chromaticity diagram has a number of interesting properties. It represents every physically realizable color as a point, within a well-defined boundary (representing the spectral colors). It has a white point at its center, with more saturated colors radiating outwards from white. When superimposing light coming from two different sources, the resulting color percept lies on a straight line between the points representing the component lights in the diagram. We can represent the range of all colors that can be produced (the *color gamut*) by means of three primaries as the triangular area of the chromaticity diagram whose vertices have coordinates defined by the chromaticities of the primaries (Figure 4.6, right). The right half of Figure 4.6, for instance, represents the gamut defined by the NTSC[3] color TV primaries. It is immediately obvious that not all physically realizable colors can be realized by the NTSC primaries. In choosing primaries, one generally tries to maximize the area of the chromaticity diagram covered, subject to technical and other constraints [McIlwain & Dean 1956]. The same holds, mutatis mutandis, for other color producing devices like printers and computer monitors. There is much more to be said about chromaticity, but this will suffice for our purpose. The interested reader can consult [Wyszecki & Stiles 1982] for more details and references.

More recently, the CIE has defined some additional color spaces, based on the notion of perceptual color difference expressed as Euclidean distance in the space (Appendix B). Color spaces with this characteristic are generally referred to as uniform color spaces. The best known examples of these are the CIE L^*, u^*, v^* (for additive light) and L^*, a^*, b^* (for

[3] National Television Standards Committee, alternatively known as the "Never The Same Color" standard.

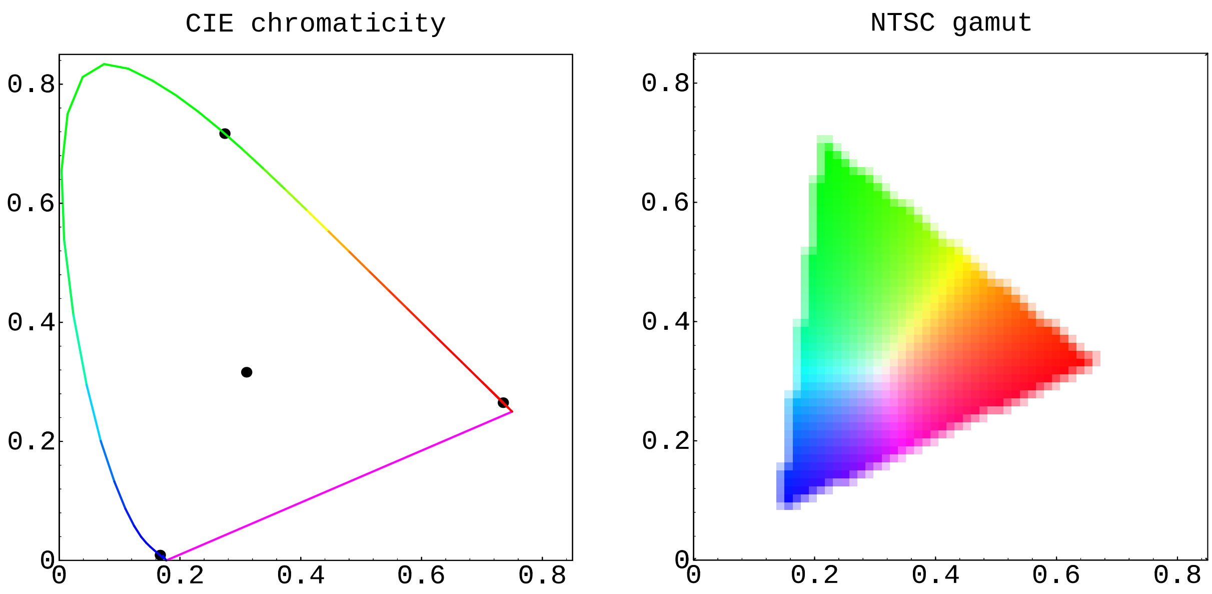


Figure 4.6: By plotting the x and y chromaticity coordinates of visual stimuli in a parametric plot, we obtain the CIE chromaticity diagram (left). The inverted U-shape defines the locations of all spectral (pure monochromatic) colors, and the straight line closing the U shape, known as the purple line, defines the most pure extra-spectral purples (mixtures of red and blue). All physically realizable colors lie within this area. For instance, the 3 points marked along the edge (clockwise, starting from the lower left corner) represent the chromaticities of the CIE spectral blue, green, and red primaries of Figure 4.4. The central point represents a reference white color (known as CIE C). The more "pure" (saturated) the color, the further away from the white point it lies. On the right, an approximate representation of the gamut defined by the NTSC color TV primaries (see text).

reflected light) [Wyszecki & Stiles 1982, Novak & Shafer 1992]. [Robertson & O'Callaghan 1986] report that linear interpolation in these spaces (for computer graphics) is superior to the more common RGB and HSL spaces. [Novak & Shafer 1992] believe that it is unlikely that even the use of CIELUV coordinates[4] will solve the fundamental problems of color image segmentation and analysis.

4.4.3 RGB color models

As [Boynton 1990] points out, it has been generally understood for nearly 200 years that the initial basis for color vision lies in the differential excitation of three different classes of cone photoreceptors in the retina (Figure 4.7; see also Section 2.1). If we represent a

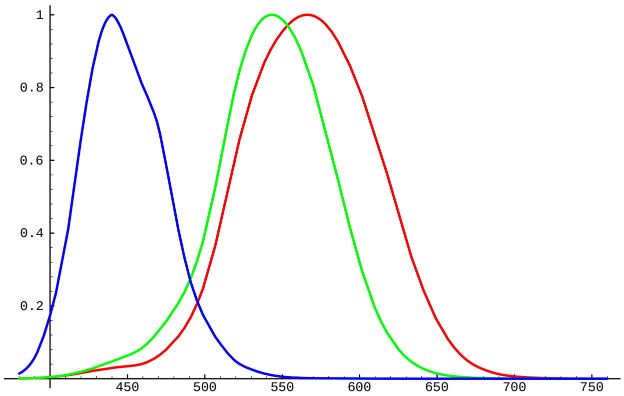


Figure 4.7: Relative spectral sensitivity of the three types of cones in the human retina, after [Wyszecki & Stiles 1982]. On the Y axis: relative response. On the X axis: wavelength of monochromatic stimulus light, in nanometers. From left to right: the S, M, and L (Short, Medium and Long wavelength sensitive) types, also known as the B, G, R (Blue, Green, and Red sensitive) types, respectively. The latter names are somewhat misleading, given the broad-band responses. The functions have been normalized for a maximum of unity, but in absolute terms the S type is much less sensitive than the M and L types.

stimulus as a spectral power distribution (SPD), which is in turn represented as a function of wavelength E, as are the cone sensitivity functions $\overline{l}$, $\overline{m}$, and $\overline{s}$, and if we further assume

[4]Another type of color space coordinates endorsed by the CIE.

that the response of the visual system is linear, we can represent the response of each of the cone types to the stimulus as follows (Section 2.1):[5]

$$L = \int E(\lambda)\bar{l}(\lambda)d\lambda \tag{4.7}$$

$$M = \int E(\lambda)\overline{m}(\lambda)d\lambda \tag{4.8}$$

$$S = \int E(\lambda)\overline{s}(\lambda)d\lambda \tag{4.9}$$

The entire SPD is thus reduced to three numbers only. Obviously there is considerable loss of information in this transformation, so that many different SPDs will result in the same set of three LMS values. Any two stimuli that result in identical LMS values cannot be distinguished from each other; we call such stimuli metameric (Figure 4.8). This reduction can be seen as a first phase of categorization. The next phase, color perception and naming, occurs after the transformation to three values, and is the main topic of this dissertation. Without serious "data reduction" and categorization, the world would be a place of "blooming, buzzing confusion", as William James put it. It is the phenomenon of metamerism, based on three different receptor sensitivities, that has made color TV and color computer graphics possible. If our visual system were sensitive to the shape of SPDs themselves, i.e., to differences in the amplitude of individual wavelengths, rather than to the weighted energy content of three broad spectral bands, it would be much harder to reproduce color in any way.

In the cases of color TV or color vision research, a scene is captured using photosensors with broad spectral sensitivities resembling those of human cones (Figure 4.9) [McIlwain & Dean 1956, Ballard & Brown 1982]. Once an image is captured, colors are defined as RGB triplets. In computer graphics work, these RGB triplets are generated directly, e.g., from an underlying imaging model [Foley & Van Dam 1982, Hill 1990]. The colors are then reproduced by using the RGB values to drive three independent electron guns in a cathode ray tube, each activating a particular kind of electroluminescent phosphor. The combined light emitted from the three phosphors will then ideally result in cone excitations that are proportional to what they would have been if the scene were viewed directly. RGB values

[5] The assumption of linearity is widely held in color science, e.g., [Wyszecki & Stiles 1982]. There may, however, be some non-linear phenomena involved in color perception, as I will discuss below.

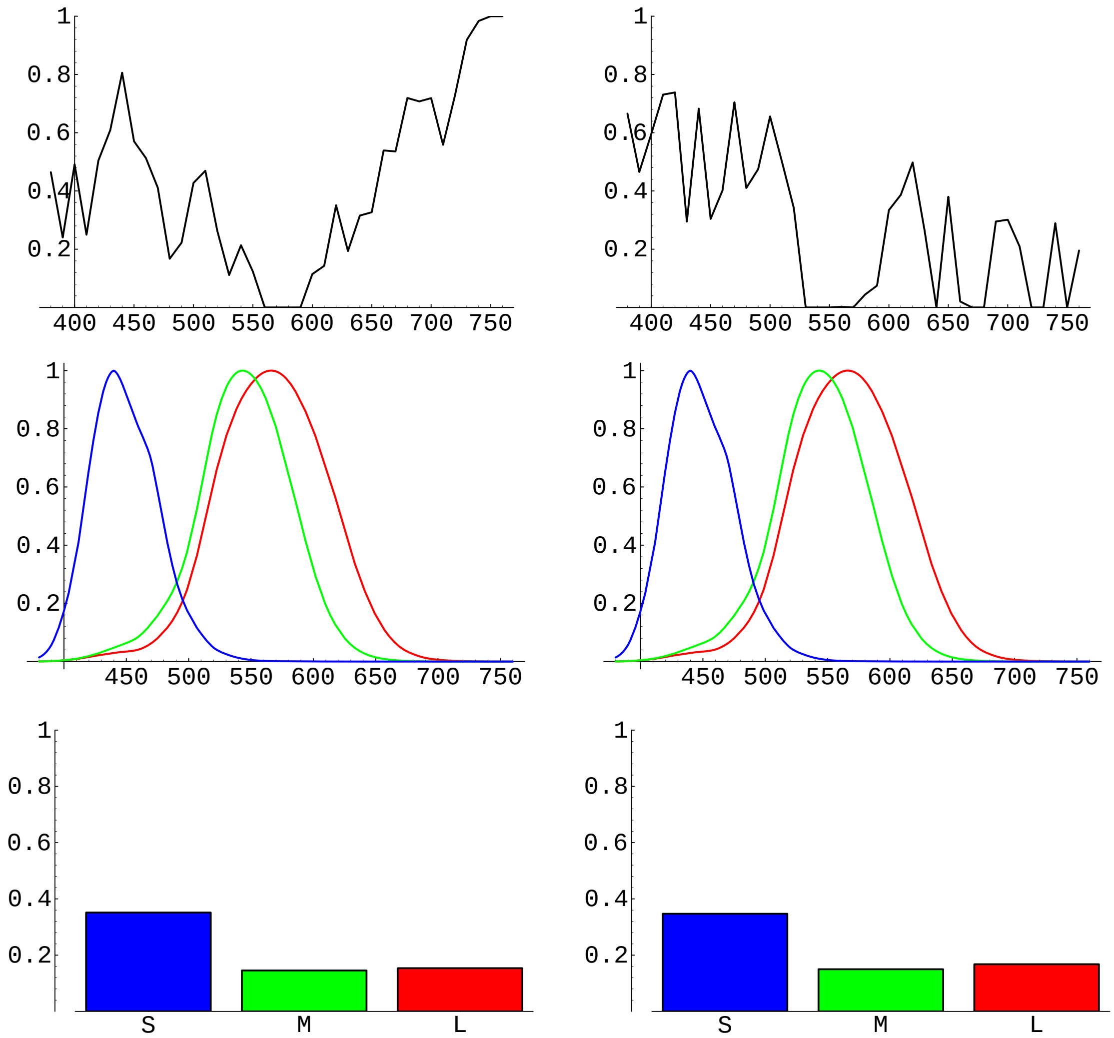


Figure 4.8: Metamerism: the two different SPDs in the top row are integrated with the same spectral sensitivity functions in the middle row, and the resulting LMS values are shown in the bottom row. Even though the SPDs are different, the LMS values are identical, and thus the perceived color is the same.

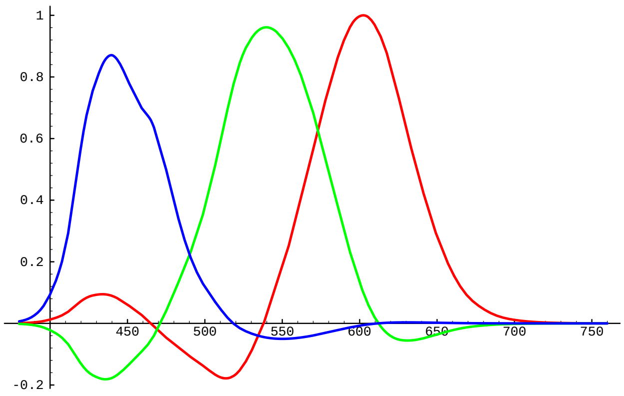


Figure 4.9: Spectral sensitivities used for color TV image capture, as defined by the NTSC. The negative components of these functions cannot be physically realized, and have to be dealt with in some way during camera design and calibration.

can also be used in color printing, after conversion to the complementary CMY (Cyan, Magenta, Yellow) colors, using subtractive rather than additive principles [Rogers 1985].

4.4.4 Computer graphics and computer vision color models

Several different color spaces have been used in computer graphics and computer vision work, see e.g. [A. Smith 1978], [GSPC-ACM 1979], [Meyer & Greenberg 1980], [Foley & Van Dam 1982], [Cowan 1983], [Rogers 1985], [Turkowski 1986], [Hill 1990], and [Novak & Shafer 1992] for some examples, and Appendix B for some mathematical definitions. Most of these spaces are derived from either the RGB or CIE XYZ class of models. These derivations can involve linear or non-linear transforms [Rogers 1985]. Many of the models involved resemble the Munsell and Ostwald color spaces (Section 4.4.1) in that they use dimensions corresponding to hue, saturation, and one of brightness (HSB), value (HSV), intensity (HSI), or lightness (HLS or LHS models). Some models have been inspired by opponent process models of color perception (Section 4.4.5).

4.4.5 Hurvich & Jameson's opponent-colors theory

Probably the best known opponent model of color perception is the *opponent process theory* of Hurvich and Jameson, described in a series of articles ([Jameson & Hurvich 1955, Hurvich & Jameson 1955, Jameson & Hurvich 1956, Hurvich & Jameson 1956, Jameson & Hurvich 1968], see also [Hurvich 1981]). The roots of this theory can be traced back to Hering's theory of color vision [Hering 1878] in terms of *opponent hues* which cancel each other when superimposed:[6] yellow and blue on the one hand, and green and red on the other. Hurvich and Jameson developed an experimental procedure which they called *hue cancelation*, which allowed them to quantitatively express the relative amounts of each of the four basic hues present in any spectral stimulus. Figure 4.10 represents an example of the resulting *chromatic response* or *chromatic valence* functions. The chromatic response functions are

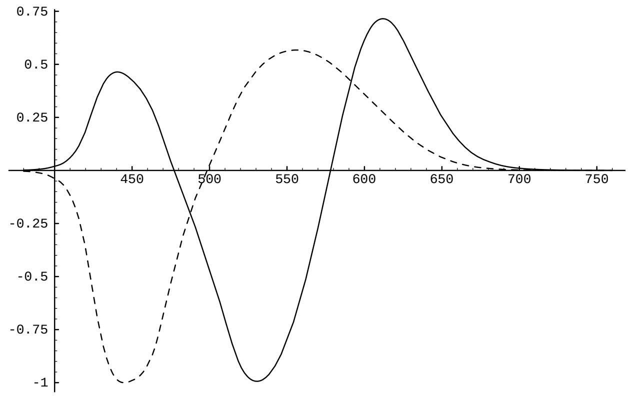


Figure 4.10: Hurvich & Jameson's chromatic response functions. For each wavelength (X axis) the functions give the relative amount of each of the four basic hues present in a pure monochromatic stimulus of that wavelength (Y axis). The members of opponent pairs are given arbitrary but opposite signs. Dashed line: blue (negative), yellow (positive). Full line: red (positive), green (negative).

determined experimentally for each individual observer, but, although there are slight interpersonal differences, the general shape of these functions remains the same. The functions presented in Figure 4.10 are derived from the CIE 1931 standard observer color matching

[6]Unless specified otherwise, I will always refer to mixtures of colored *light*, rather than dyes or pigments. The former are known as *additive* mixtures, the latter as *subtractive*.

functions, and may be taken as representative for a large number of observers. Chromatic response functions are assumed to correlate directly with hue perception. In addition to the chromatic response functions, an *achromatic response function* is determined, which represents the perceived brightness of a stimulus of a given spectral composition. Based on the chromatic response functions, Hurvich and Jameson also derived *hue coefficient functions* (Figure 4.11) and *saturation coefficient functions* (Figure 4.12). The hue coefficient

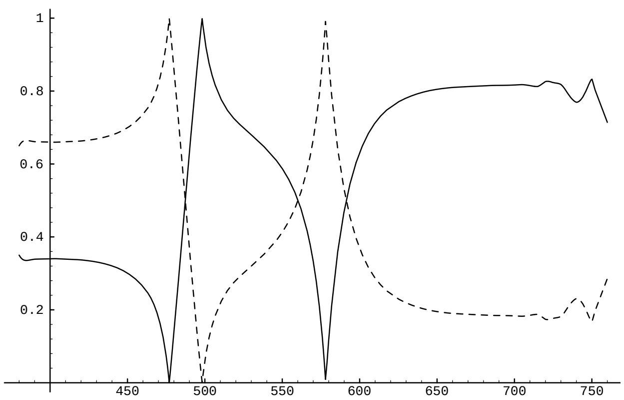


Figure 4.11: Hurvich & Jameson's hue coefficient functions. For each wavelength (X axis), the functions give the ratio of each of the chromatic response functions to the sum of all chromatic responses (Y axis). Dashed line: blue ($< 500nm$), yellow ($> 500nm$). Full line: red ($< 480nm$ and $> 580nm$), green ($480 - 580nm$). Note the instability of the functions at the extreme low and high end of the spectrum, because of the division by very small values.

functions express hue as the ratio of each chromatic response to the sum of all chromatic responses, at each wavelength. Thus, each of the four basic or *physiologically primary* hues has associated with it a function which varies between 0 and 1 over the visible wavelength spectrum. The wavelengths where these functions reach 1 are the loci of the *unique* or *pure hues*. Only unique blue, green, and yellow exist as spectral colors, but not red, as shown in Figure 4.11. The saturation coefficient functions give the ratio of the sum of the chromatic responses to the achromatic response at each wavelength, and are supposed to reflect the perceptual saturation of monochromatic stimuli of different wavelengths. Hurvich and Jameson showed that their model could account well for the stimulus attributes of spectral lightness, hue, and saturation, and their associated psycho-physical functions. They also showed how their model could account for some changes in perceived brightness,

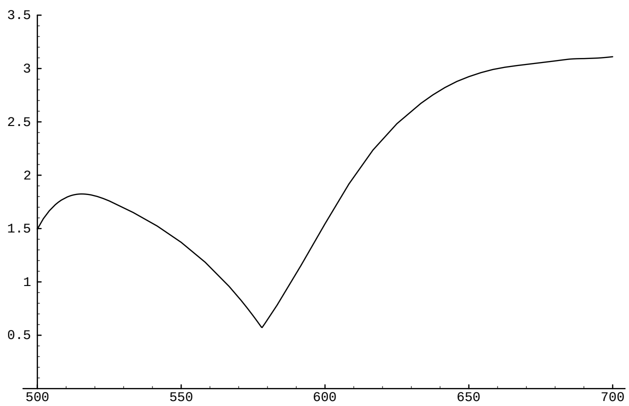


Figure 4.12: Hurvich & Jameson's saturation coefficient function. For each wavelength (X axis), the function gives the ratio of the sum of the chromatic responses to the achromatic response (not shown). Note that the function is only useful within a narrow frequency band.

saturation, and hue with chromatic adaptation, and to some extent for color constancy phenomena. They finally defined a "psychological color specification system", which is a hue/saturation/brightness (HSB) color space based on hue and saturation coefficients in a polar coordinate system, and achromatic response as the brightness dimension.

Hurvich and Jameson's model is the first quantitative color opponent model based on psychophysical experimentation, and as such deserves a lot of credit. They demonstrate that an impressive range of psychophysical phenomena can be explained elegantly in terms of their model. However, the model has a number of drawbacks for use as a computational model of color perception and color naming, which I now outline briefly. One could interpret the hue coefficient functions as *categorial perception* functions, but it is not clear how to extend these to colors other than the four physiologically unique hues (as Hurvich and Jameson call them). All their data deal only with some (moderate) luminance levels, and there seems to be no systematic way to extend it to arbitrary luminance and adaptation levels. Their discussion of adaptation phenomena is also limited to some specific steady-state adaptation states, which require the spectral composition of the adapting light to be known beforehand; i.e., the model is not useful as a computational model of color perception if it is to operate in real-world environments where lighting characteristics are not known. In addition, their model says nothing about the dynamics of adaptation with respect to time.

With the requirements for computational models of color perception in mind, Hurvich and Jameson's model is to some extent a descriptive rather than an explanatory model. Another problem with their discussion of adaptation is that it implicitly assumes that the different photoreceptor types (the cones in the retina with different spectral sensitivities) can change their adaptation states independently of each other. In view of their assumption that adaptation is proportional to receptor output, and the large degree of overlap of the spectral sensitivities of the photoreceptor types they propose (or even the ones currently accepted), this independence is very unlikely. [Wyszecki & Stiles 1982] discuss problems with another assumption underlying Hurvich and Jameson's work, viz., the linearity laws for opponent hue cancelation. These laws seem to hold for the red/green responses, but not under all circumstances for the yellow/blue responses. The color space defined by Hurvich and Jameson is only partial; there is no representation of an actual 3D color solid, only selected planar sections through it at fixed luminance levels. The regular spacing of hues in the polar coordinate system (red diametrically opposite green, and yellow to blue, with the axes perpendicular to each other) is unlike that of, e.g., the Munsell color model. As I mentioned above, the categorial boundaries of the four basic hues (red, green, yellow, and blue) are "built in" to the color space, and it is not clear how other perceptual color categories would fit in. The basic organization of my own color space using opponent color pairs is of course similar to that of Hurvich and Jameson.

When compared to neurophysiological data, particularly [De Valois et al. 1966], the red response in the short wavelength region is strikingly absent, although the overall similarity is clear. It is not so clear how this discrepancy can be explained, other than to postulate some unknown higher level neural mechanism that would be responsible.

4.4.6 De Valois & De Valois's color model

In [De Valois & De Valois 1975], the authors suggest that the six types of LGN cells found (see Section 4.3) can be grouped into three dimensions, when mirror-image pairs are combined into one dimension. That results in red-green, blue-yellow, and black-white dimensions, which can be arranged in a "double cone"-type color space. Hue would be coded in a circular fashion (ranging through blue, green, yellow, red, and back to blue), saturation as distance from the center of the hue circle (making hue and saturation specifiable with a polar coordinate system in the plane), and brightness along an axis perpendicular to the hue circle. This kind of color space is well known from early work of, e.g., Munsell [Munsell 1946, Birren 1969a] and Ostwald [Birren 1969b]. De Valois and De Valois also mention that this kind of color model can be "semiquantitatively" tested, e.g., through color naming experiments. My color model, presented in Chapter 5, is very close in spirit to the model they suggest, but as their adjective use suggests, theirs is not specified in any quantitative detail. Their model is certainly not specified well enough to be usable in the context of computer vision.

4.4.7 Valberg et al.'s equidistant color space

In [Valberg et al. 1986], the authors investigate the relationship between a psychophysical color space they have defined in earlier work and the responses of color-sensitive cells in the Macaque LGN. They propose that a combination of the responses of various types of spectrally opponent cells of the Macaque LGN may be related to the equidistant color space they defined, using their SVF color difference formula. They mathematically simulate the responses of some cell types as linear combinations of hyperbolic response functions of cone outputs, and compute the coordinates of equidistant points in the Munsell and the OSA-UCS color spaces. They conclude that neural correlates to elementary hues must probably be sought in the visual cortex, not the LGN. In other words, some kind of second-stage processing is necessary for the LGN cell responses to be turned into something that can explain elementary hue sensations (red, green, yellow, and blue). The color model I present in Chapter 5 can be thought of as such a second-stage model. Valberg et al.'s models of LGN cell responses are not used for computer vision work. Their work examines some properties of these responses in isolation, while my work attempts to construct a complete color space based on the responses, usable for computer vision work.

4.5 Models of Color Perception and Color Naming

As outlined in the introduction, the working hypothesis of this dissertation is that the characteristics of the underlying neurophysiological color vision process explain phenomena like the universality of basic color category foci and graded membership of color samples in color categories. In other words, both universal basic color foci and graded membership properties must follow directly from the properties of the used color model, if we want the color model to explain human color naming behavior. It is not sufficient for a color model to merely *categorize* all possible color percepts in a systematic way; it must also explain why some color percepts are more salient than others, and why category membership judgments are graded. I hypothesize that when learning basic color names, some points in the learning space are more salient than others and act as "attractors" for category foci. These properties must follow from the color model, and it is against these criteria that I will judge the usefulness of any particular color model. None of the existing color models meets these criteria.

In the light of the requirements for color vision models set forth above, we can immediately disqualify a number of simplistic models of color naming that might be intuitively appealing:

- Extensions of color names as intervals on the wavelength line: Clearly, this kind of model does not accommodate or explain the existence of universal foci or graded membership functions. In addition, this model does not allow for non-spectral colors like purple or brown at all. It is this kind of arbitrary model that Berlin and Kay (successfully) sought to disprove in their study [Berlin & Kay 1969].

- RGB-based models: RGB models are loosely based on the three types of photoreceptors and their spectral sensitivities in the human retina (Section 4.4.3). As such, they model a very early stage in the human color vision process, probably too early to be useful for our purpose. Indeed, already at the LGN level, RGB-like signals have been transformed into an opponent coding (yellow–blue and red–green), which continues to the cortical level [Dow 1990]. As [Hurlbert 1991] points out, RGB codes perceived colors neither uniquely nor efficiently. Since the cones adapt to the intensity and spectral composition of the ambient light (including surrounding image areas), RGB values do not uniquely specify colors. In addition, the ranges of wavelengths that the three cone types are sensitive to overlap greatly, especially for the R and G cones. This makes for rather inefficient coding of color. According to Hurlbert, the brain has evolved a second-stage mechanism to transform cone triplets into more reliable and discriminatory code, which discounts redundancies and enhances differences in cone responses.

She claims that few color scientists would disagree that this color-opponent processing occurs, but there is disagreement on exactly which directions the opponent-color axes take in color space.[7] To summarize, although opponent representations are usually thought of as "merely" linear transforms of RGB space, the opponent representation is much more suitable for modeling perceived color than RGB is.

From a psychological point of view, the inappropriateness of RGB as a color model is clearly demonstrated by the difficulty people experience in manipulating RGB color coding in a consistent and intuitive way, and by the recent interest in different kinds of color models (among which are opponent models) in computer graphics research [Meyer & Greenberg 1980, Naiman 1985, Rogers 1985, Turkowski 1986, Robertson & O'Callaghan 1986, Scheifler & Gettys 1992]. [Rogers 1985] points out that RGB and all of the linear transforms thereof (CIE XYZ, YIQ, CMY) are difficult for users to specify subjective color concepts in. A closer look at the properties of RGB representations provides independent evidence. The RGB color solid has the shape of a unit cube, with the achromatic (grey level) dimension going from $\langle 0,0,0\rangle$ to $\langle 1,1,1\rangle$, and the maximally saturated colors red, green, and blue at a lower brightness level than the maximally saturated colors yellow, cyan, and purple, all of which are situated at corners of the cube (Figure 4.13). This kind of organization does not match the color naming data very well. As [Shepard 1987] points out, the representation of stimulus data is of major importance for the interpretation of experimental psychological results. He cites the example of generalization across stimuli, which can exhibit a non-monotonic increase between stimuli separated by certain special intervals in a metric space, for example, between tones separated by an octave, between hues at opposite ends of the visible spectrum (red and violet), and between shapes differing by particular angles related to inherent symmetries of those shapes. If we use a circular hue representation rather than a wavelength interval representation, as in the Munsell color space, for instance, red and violet become neighbors rather than opposites, and the apparent nonmonotonic effect disappears. Similar conceptions of psychological spaces exist for pitch [Patterson 1986] and shape [Walters 1987].

[7] The reddish-green and yellowish-blue reported to be perceived in some experiments [Crane & Piantanida 1983], and which apparently contradicts opponent process theories of color vision, seems to be a function of lateral interactions in the cortex, and occurs only under very special circumstances, mostly in the absence of a corresponding retinal stimulus. As such it is not a major objection to opponent process theories, since they deal with ordinary stimuli in the main visual pathway.

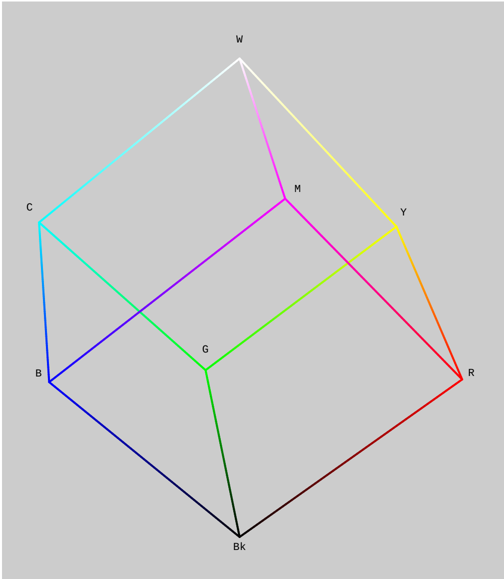


Figure 4.13: The RGB intensity color cube. The coordinate system is tilted so that the black-white axis is displayed in vertical position. The vertices of the cube correspond to minima and maxima along the RGB dimensions: Bk=Black=(0,0,0), R=Red=(1,0,0), G=Green=(0,1,0), B=Blue=(0,0,1), C=Cyan=(0,1,1), M=Magenta=(1,0,1), Y=Yellow=(1,1,0), and W=White=(1,1,1).

The same holds, mutatis mutandis, for lookup tables specifying RGB values indexed by color names (as found in the X-windows system, for instance), or the other way around. This kind of "model" does not even provide a name for every point in the space, and graded membership or learning names in a productive way is even harder to model in this context, not to mention the existence of universal color foci.

Other kinds of existing color models (CIE chromaticity, opponent models, psycho-physical and colorimetric models in general) all suffer from the same disadvantage of not being able to explain the existence of universal foci, and to a lesser extent, the graded category membership functions.

The only existing models of color naming based explicitly on the neurophysiology of color vision and attempting to explain the universality of foci and graded membership functions are [Cairo 1977] and [Kay & McDaniel 1978]. Apart from not being defined or implemented as full-fledged computational models, both of these have important drawbacks. Kay and McDaniel's model interprets (stylized versions of) the response functions of four types of color-sensitive cells in the LGN [De Valois et al. 1966] as characteristic functions of four fuzzy sets corresponding to the categories red, green, yellow, and blue. As such, the model explains the existence of universal foci (which correspond to maxima of the characteristic functions) and the graded membership functions (which correspond directly to the characteristic functions) of these four basic color categories. But the model has to be tweaked to account for other basic color categories (requiring the introduction of new and ad hoc fuzzy set operations and a nonlinear compression of the wavelength dimension that is without apparent external motivation), and it is not clear at all how non-spectral basic color categories like brown or purple are to be dealt with, nor how to model the learning of color names in this model. It also does not adequately explain the hierarchy of languages with respect to the lexical encoding of basic color categories. Another objection that can be brought against the model is that it predicts that a flat-spectrum white stimulus light is a good example of *every* basic color category, since there is no opponent mechanism that cancels out opponent primaries. This prediction is clearly not born out by experience or by psychophysical experimentation.

Cairo's model of color naming [Cairo 1977] is also based on findings in the physiology of the pre-cortical visual system. It is four-dimensional: wavelength, intensity, purity, and a fourth dimension representing the adaptation state of the retina. All of these dimensions are defined as physical parameters of the stimulus, rather than as perceptual dimensions. Cairo introduces a "data-reduction model" of analog-to-digital conversion in the pre-cortical visual system, and represents Berlin and Kay's eleven basic color categories as specific combinations of quantized values on the four dimensions. In addition, he predicts four potential new basic color categories beyond Berlin and Kay's eleven, which he calls sky-blue, turquoise, lime, and khaki.

Although this model is interesting for its attempt to take adaptation into account, it suffers from a number of important drawbacks. For instance, the discrete nature of the model and the use of wavelength as one of the dimensions forces complex stimuli to be treated as if they were monochromatic, and this is done using the "dominant wavelength" to represent an entire spectrum. While this may work in limited circumstances, it is clearly a gross simplification which cannot hold in general (e.g., which is the "dominant wavelength" of the spectra of Figure 4.8?). In addition, Cairo claims that Berlin and Kay's finding of universal foci of categories is a mere artifact introduced by the use of the Munsell color system for the stimulus material used in the experiment. This conclusion does not seem to be supported by the wealth of psychological and linguistic research done since Berlin and Kay's initial study (cf. the bibliography of recent research in [Berlin & Kay 1969]). On the contrary, it seems that it is rather Cairo himself who is forced to deny the evidence because of the discrete nature of his model, which does not allow for graded categories. The discretization of the model itself seems to be fueled by a desire to make color notation the subject of algebraic analysis, requiring a complete ordering of a discrete number of possible color values. While that may be a noble cause, it also requires most of the data on actual color naming behavior to be ignored, which is too much of a price to pay for a convenient analysis. Berlin and Kay's hierarchy of languages is explained in a rather circular fashion by Cairo in terms of the "perceptual salience" of the four underlying dimensions, but without explaining where their salience derives from. In general, this model seems overly biased by an extreme information processing view of psychology that regards cognition quite literally as a digital computational process.[8]

[Harnad et al. 1991] present a rather different approach to the categorization problem, using artificial neural networks. They consider the problem of separating lines into two classes. After training an auto-associator network (one that essentially reproduces its input as its output), they add a categorization task to the same net. The result is that the "similarity space" derived from the net's hidden nodes' activations is warped to reflect categorial boundaries. The analysis of the network representations is interesting, and the authors provide a good discussion of network representations, but the applicability of the approach seems limited. An interesting thing to note is that the representations used by this network seem just as ungrounded as the symbolic representations that the first author criticizes in some of his other work [Harnad 1990].

[8]Of course it is easy to see this with the 20/20 vision of hindsight. At the time Cairo wrote his dissertation, these models were just as "hip" as graded cognitive models are today. Although his assumptions are now relatively widely regarded as wrong, his work nevertheless provides some valuable insights.

Chapter 5

From Visual Stimuli to Color Space

In order to construct a mapping $\mathcal{N}$, rather than just define it extensionally as I did in Section 2.3 (p. 24), we need a theory of how to transform a function $E \in \mathbf{E}$ into one or more pairs $\langle c, i \rangle \in \mathbf{C} \times \mathbf{I}$.[1] Recall that E represents the spectral energy distribution of a stimulus, as a function of wavelength; $\mathbf{E}$ represents the set of all such distributions; $\mathbf{C}$ represents a set of color terms; $\mathbf{I}$ represents the closed interval $[0, 1]$; and $\mathcal{N}$ represents a mapping $\mathbf{E} \rightarrow \mathbf{C} \times \mathbf{I}$, i.e., a color-naming mapping. I will conceptualize the mapping $\mathcal{N}$ as being a composition of two mappings: one that takes us from $\mathbf{E}$ to what I will refer to as a *color space*, and one that takes us from that color space to $\mathbf{C} \times \mathbf{I}$. This chapter discusses the first one of those mappings.

5.1 General

The first "half" of the mapping $\mathcal{N}$ needs to transform a visual stimulus $E \in \mathbf{E}$ into a point in a color space. In Shepard's terms, this is a psychophysical function that maps physical parameter space into psychological space [Shepard 1987]. The reasons for using a

[1]The term *theory* is used here in the sense of David Marr's distinction of the computational theory, algorithm, and implementation levels [Marr 1982].

color-space transformation are many:

1. The psychophysical literature on color perception commonly uses a color-space model to represent color percepts; see Chapter 4 (p. 42). If we want to evaluate the psychological validity of our model, using similar representations allows us to compare results easily.

2. The color-space model reflects important constraints on color perception; e.g. the constraint that similar spectra by and large evoke similar percepts is reflected in the mapping of similar spectra onto nearby points in the color space. This is of course a special case of the *continuity* constraint introduced by [Marr 1982].

3. The color-space model allows us to easily model some of the tasks a cognitive theory has to explain (Section 4.1 p. 42), e.g.,

    - *Discrimination*: Two stimuli are discriminable if and only if they map into distinct points in the color space (data on the minimum differences people can discriminate under varying circumstances allow us to adjust the "resolution" of the space to match human performance, if desired).

    - *Similarity judgment*: We can straightforwardly model similarity between different stimuli as the distance between the corresponding points in the color space (the geometry of the space need not be Euclidean, however).

    - *Identification*: we can associate regions of the color space with foci and boundaries of color categories, as shown in [Boynton & Olson 1987], which is of course a consequence of the continuity constraint mentioned above.

The new color space I will describe is based as closely as closely as possible on known data about the neurophysiology of color vision. In particular, an opponent model of color vision as described in [De Valois et al. 1966, De Valois & Jacobs 1968, Dow 1990] seems to be appropriate, since it conforms with widely accepted psychophysical theories of opponent color systems, particularly the Young/Hering theory, cf. [Boynton 1979, Boynton 1990].

5.2 Assumptions and Limitations

In modeling color perception, I will make some limiting assumptions to make the problem tractable:

1. I am concerned with single-point determination of perceived color only; hence, I disregard spatial adaptation and interactions in color perception [Boynton 1979]: I will not be concerned with such phenomena as induced colors[2] or Mach bands[3] [Boynton 1979]. Context is only taken into account implicitly to the extent that it is necessary for the discrimination and identification of basic color categories. For example, a color like *brown* is not distinguishable from *orange* in the so-called aperture mode of viewing (in isolation, through a small hole in a screen),[4] because *brown* is a "dark color" which is only defined with respect to brighter colors in its surroundings. When we examine the spacing of basic color categories in a color space, we implicitly take this into account by comparing the location of *brown* to that of *orange* or *white*, for instance.

2. I assume a fixed adaptation state of the visual system; i.e., I am not concerned with issues of color constancy (invariant perception under varying light sources), pigment bleaching, temporally induced colors,[5] and the like [Boynton 1979].

3. I assume foveal photoreceptors as sensors, and their spectral sensitivities as defined in the CIE 1931 standard observer data [Wyszecki & Stiles 1982]. Although there is some evidence that there are inaccuracies in these data, and several revisions have been proposed and published, the standard is so well established that I have chosen to adhere to it. From a practical point of view, it is also the basis of the NTSC color TV standards and corresponding camera specifications. Moreover, the existing revisions like Judd 1951, CIE 1968, and Vos 1978 [Wyszecki & Stiles 1982] are not so different that they would have much effect on the relative locations of basic color categories in the color space.

[2] Apparent colors that are perceived in proximity to certain other colors.

[3] Narrow bands of lighter or darker shades of a color perceived around the boundary separating two bands of different colors.

[4] Thanks to Chris Brown for pointing this out to me at the outset of my work. I wasn't convinced until he told me to look for a brown light.

[5] Apparent colors that are perceived when looking at a temporally alternating series of colors.

Although these assumptions and limitations represent a considerable simplification in some ways, they do not render my work useless for practical applications, as I will show below.

5.3 Modeling Neurophysiological Data

I have based the color space which I will refer to as the *NPP space*[6] on the neurophysiological data published in [De Valois et al. 1966]. Section 4.3 (p. 49) discusses this data and the motivation for choosing it to base a color space on.

5.3.1 Reconstruction of 3D response functions

The data in [De Valois et al. 1966] consist of the averaged responses of six types of cells in the Macaque LGN to monochromatic stimulus light, sampled at three radiance levels and 12 to 13 wavelengths for each cell type. The cell types are designated according to the approximate color of the light that results in a maximum excitatory (+) or inhibitory (−) response: the four opponent types $+R-G$, $+G-R$, $+Y-B$ and $+B-Y$, and the two non-opponent types referred to as excitators and inhibitors, which I will refer to as $+L$ and $-L$, respectively (the symbols are R for red, G for green, Y for yellow, B for blue, and L for any light). Figure 5.1 shows the six data sets. The inhibitory (negative) phases of the responses are generally much smaller than the excitatory (positive) phases, because they are relative to the spontaneous firing rate, which is typically 5–10 spikes/s, and negative firing rates are obviously not possible. The columns of Figure 5.1 show pairs of cell types that are complementary in nature, with responses that are a kind of "mirror image" of each other (when disregarding the absolute sizes of the phases): in a wavelength range where one type is excited, the other is inhibited (and the other way around), and the cross-over wavelengths between excitation and inhibition (if any) are virtually identical. The use of these mirror-imaged channels is common in the vertebrate visual system, and may be a way to encode a larger dynamic range than would be possible with a single channel [Slaughter 1990].

[6] Neuro-Psycho-Physical color space; thanks to K.N. Leibovic for the suggestion.

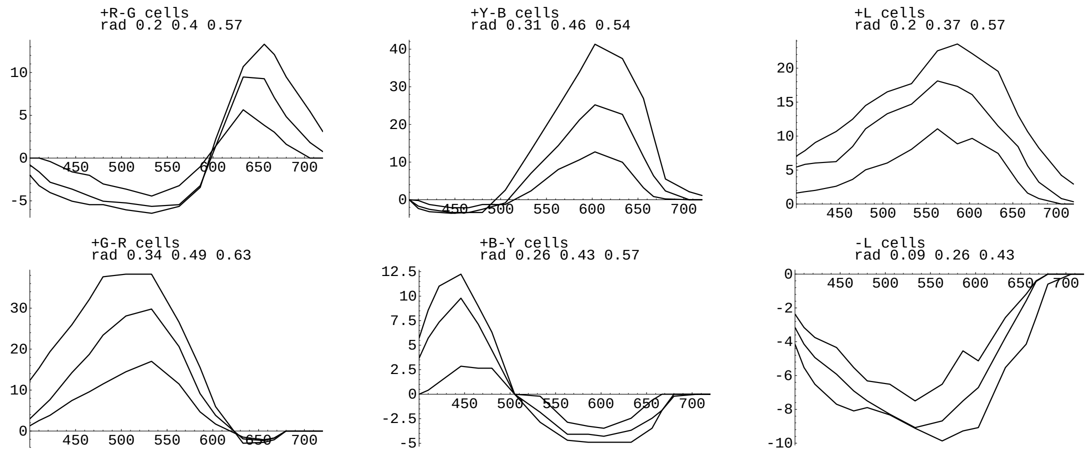


Figure 5.1: Data sets for six LGN cell types, after [De Valois et al. 1966]. On the Y axis: response relative to spontaneous firing rate (spikes/s). Note that the Y scales are different. On the X axis: wavelength of monochromatic stimulus light (nm). Data points are connected with straight line segments for clarity. Each plot shows three data sets at the relative radiances listed above the plot, higher radiances correspond to larger excursions (see text).

To reconstruct the 3D response functions[7] from this relatively sparse data set, I started by fitting a model to the data in the radiance domain, followed by interpolation in the wavelength domain. The reason for starting in the radiance domain is that a lot is known about the response characteristics of photoreceptors and other cells in the visual system with respect to changing stimulus intensity, and this knowledge provides important constraints on the model. Based on the literature (e.g., [Leibovic 1990a, Slaughter 1990, Frumkes 1990]), I have assumed a sigmoid-shaped function as the basic response model in the radiance domain:

$$S(r) = (1 + e^{\frac{h-r}{t}})^{-1} \tag{5.1}$$

$$S(h) = \frac{1}{2} \tag{5.2}$$

$$\frac{dS}{dr} = \frac{1}{4t}, r = h \tag{5.3}$$

where r represents radiance, h the half-response radiance, and t is a parameter that de-

[7]Response is a function of both the wavelength and the radiance of the stimulus.

termines the first derivative around the half-response point, sometimes referred to as the *temperature* parameter. Figure 5.2 shows a plot of this function for $h = 0.5$ and $t = 0.1$. In

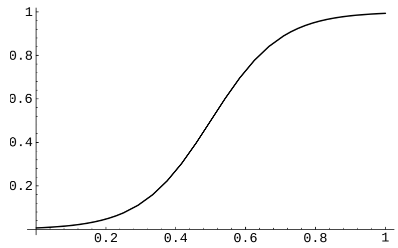


Figure 5.2: A typical sigmoid function plot for $h = 0.5$ and $t = 0.1$ (see text).

practice I use a scaled version of the sigmoid function:

$$S(r) = c(1 + e^{\frac{h-r}{t}})^{-1} \tag{5.4}$$

where c is a scaling factor that is usually 1, but takes on different values on some occasions (see below).

This type of function is particularly appropriate for modeling, e.g., the operating curve of photoreceptors that shifts along an absolute radiance axis as a result of adaptation, as background radiance increases or decreases [Leibovic 1990a]. The visual system's response is always relative to its adaptation state, and if we assume a fixed adaptation state, a function like the one in Figure 5.2 can be used to model the response. The parameter h can be interpreted as a function of the adaptation state. Near $r = h$, the sigmoid function is almost linear, but moving away from $r = h$ towards $-\infty$ and $+\infty$ it approaches the limits 0 and 1, respectively. This can be interpreted as approaching threshold response and saturation response, respectively. The sigmoid function is also commonly used as a node activation function in artificial neural networks research (e.g., [Rumelhart et al. 1986]).

To model the response of an LGN cell (type) as a function of radiance (for a fixed wavelength) with a sigmoid function, we need to interpret the data points in terms of r and $S(r)$, and then determine appropriate values of h and t that minimize the error of fit. To make the computations easier, I will use the $[0, 1]$ interval as the "useful" range of r corresponding to a value $S(r)$ going from near-zero (threshold) to near-one (saturation). The radiances in [De Valois et al. 1966] are specified as log units below some maximum used radiance. Log transformations of physical intensity are generally used in psychophysics to express perceived intensities, since they roughly correspond to a perceptually linear scale, over a significant part of the useful range.[8] Since these radiances are relative, we need to decide where they should be situated along the radiance axis. I made the following assumptions:

1. The dynamic range of the responses from threshold to saturation is 3.5 log units of input radiance. This accords well with estimates of 3 log units for the "useful" (near-linear) range given in [Leibovic 1990b] for photoreceptors, extended slightly to allow for response threshold and saturation. This value works well for fitting the sigmoid intensity response model to the data, as explained below. Since we know the difference between the highest and lowest radiance level used in the experiments[9], and all measurements are made at a constant adaptation state[10], we can derive a rough estimate of the maximum level used relative to the background light level, as

$$R_{max} = \frac{R_h - R_l}{f_{avg}(R_h) - f_{avg}(R_l)} f_{avg}(R_h) \tag{5.5}$$

   where R_{max} is the maximum level used in log units relative to the background light level, R_h is the highest and R_l is the lowest stimulus radiance used relative to the maximum available, $f_{avg}(R_h)$ is the average firing rate across the spectrum at the highest and $f_{avg}(R_l)$ at the lowest radiance used. I used the average responses to minimize measurement errors. Of course this equation assumes linear response as a function of log radiance, which we know not to be the case, but to be approximately true around the half-response radiance. Table 5.1 summarizes the results. In some cases (viz. +R-G, +B-Y, and especially -L), the estimated maximum is smaller than

[8] This is also true for perceived loudness (the decibel (dB)) or perceived pitch (semitones, octaves), for instance.

[9] The stimulus radiance levels are given as attenuation relative to the maximum available, in log units.

[10] The article states that "the animal was maintained under a low level of light adaptation", and the pauses between 1-second stimuli were 30 to 60 seconds, which should be long enough to re-establish the previous adaptation state if it were at all affected by the preceding stimulus.

type	R_h	R_l	R_{max}
+G-R	-0.2	0.8	1.43
+R-G	0.0	1.3	0.87
+B-Y	0.0	1.1	0.99
+Y-B	0.1	0.9	1.01
+L	0.0	1.3	1.92
-L	0.5	1.7	3.02
		average	1.54

Table 5.1: Estimates of the maximum stimulus radiance used for each cell type, as log units relative to the background light level. Symbols as in text.

the maximum attenuation used, which would imply light levels below that of the background adaptation light. The article is not clear on whether or not this was the case, and I have not been able to determine this any other way. The -L data are all inhibitory, i.e., negative with respect to the spontaneous firing rate, and thus has a small range. There may hence be a considerable measurement error involved for this data set. In any case, it is safe to assume that the total useful range of input radiances must be at least as large as the average of the values R_{max} as given above. In fact it must be at least as large as the maximum value of R_l (the maximum attenuation used), viz. 1.7 log units, since the cells are still responsive at that level. The latter argument would give us a range of about 2 log units. Since the measurements were most likely done at radiance levels well below saturation, the total range must be larger than this, which is consistent with the figure of 3.5 log units mentioned above.

2. The preceding argument gives us a radiance range of about 2 log units for the experimental data, with a maximum of 3.5. The radiance of the brightest stimuli is therefore about 1.5 log units below saturation level. The radiance levels for each of the data sets are shown in Figure 5.1 (p. 76), normalized to $[0, 1]$.

3. The responses of all six cell types will eventually saturate at approximately the same level if the stimulus radiance is turned up high enough, since this is more a function of the biochemistry of the cell than of the absolute stimulus intensity. To scale the responses of the cell types, I used a global maximum (saturation level) of 150 spikes/sec above the spontaneous firing rate. This is somewhat lower than typical LGN cell

saturation levels of 2-300 spikes/sec,[11] but the spontaneous rate in these data of 5-10 spikes/sec is also lower than the typical rates of about 30. The fact that the experimental subjects were anesthetized may have something to do with this general lowering of the firing rates. The maximum occuring firing rate in the data of about 50 spikes/sec for the +Y-B cell type at about 610 nm for an estimated stimulus radiance of about 2 log units would support a saturation level of about 150 spikes/sec for a total of 3.5 log units, taking the non-linearity of the response at high radiances into account.

4. Since I measure responses relative to the spontaneous firing rate, I also need to determine a maximum negative response. These responses represent inhibition, and the absolute firing rate can obviously not drop below 0. Since the spontaneous rates in De Valois' data does not exceed 11 spikes/sec, I've set the negative maximum to -12.

Although the estimates of the total useful stimulus intensity range and saturation levels are somewhat speculative (since there is not enough data to support better estimates), they are generally not in disagreement with known characteristics of LGN cell responses. In addition, the exact numbers chosen are not all that important, since using different values amounts basically to a linear scaling of the (non-linear) response functions, as I will show below.

With these assumptions in place, we can now fit the sigmoid models to the data. I manually extrapolated the data in the lower and upper wavelength ranges, so each constant-radiance data set starts and ends with a zero point. In addition, I added a zero point to each constant-wavelength set of data points, to constrain the sigmoid fit better. I used the following algorithm to do the fit for each constant-wavelength data set:

```
1.  let data = sign(data)*data
1.1 c = {Abs[NegMax/PosMax], sign < 0
        { 1.0 , otherwise
2.  compute least-squares linear fit of data as function of r: a=mr+b
3.  let h1 = solution(mr+b=0.5 c) solving for r
4.  let t1 = solution(S'(h1,h1,t,c)=m) solving for t
```

[11] K.N. Leibovic, personal communication.

```
5. minimize(SE2(r,h1,t1,c)) solving for h1 and t1
6.  return sign(data)*S(r,h1,t1,c)
```

Since the sigmoid is really a function of three variables $S(r, h, t)$ (with radiance r, half-response radiance h, and temperature t, cf. equation 5.1 p. 76), we need to optimize the fit with respect to one, r, by varying the other two, h and t. The data form a list of ordered pairs $\langle radiance, response\rangle$ for a fixed wavelength and cell type, including the added $\langle 0,0\rangle$ point. Step 1 makes sure that all response values are positive. In principle, a data set may contain both negative and positive response values, especially around the cross-over point between inhibition and excitation, although in practice this is rare. This step is more useful for the inhibitory phases of the response, which are represented as negative values since I measure responses relative to the spontaneous rate. I define the sign of the data set as the sign of the sum of the response values in the set:

$$sign(D) = sign\left[\sum_{i\in D} 2nd(\langle r,a\rangle_i)\right] \tag{5.6}$$

$$sign(x) = \begin{cases} -1 & ,x < 0 \\ 0 & ,x = 0 \\ 1 & ,x > 0 \end{cases} \tag{5.7}$$

where D is the data set, $2nd$ is the function that selects the second element from an ordered pair, r is the relative radiance, and a is the response (activation). The sigmoid function has to be either entirely positive or entirely negative, and this step chooses between those. Note that this also turns a set of data points whose a's sum to zero into all-zero a's. This can only happen around the cross-over point between inhibition and excitation, and is not wrong, even desirable, because varying the radiance alone should not affect the sign of the response, so the data are probably unreliable in that case. Step 1.1 chooses the scaling factor for the sigmoid function, 1 for positive data and the ratio of the maximum negative to the maximum positive response for negative data.[12] To find initial values for h and t,

[12] Modeling the negative (inhibitory) responses with a sigmoid function may be questionable, but in view of the small inhibitory responses relative to the excitatory ones, it does not make much difference anyway. The unified approach to modeling both inhibitory and excitatory responses is to be prefered, in my opinion.

we first compute a linear fit to the data set in step 2, using Mathematica's least-squares Fit function. Solving for a linear equation value of $0.5c$ gives us an initial estimate $h1$ for h in step 3 (recall that h is the half-response radiance, and the sigmoid function values are normalized to $[0, 1]$). Step 4 gives us an estimate $t1$ for t such that the first derivative of the sigmoid function at $r = h1$ is equal to the slope of the linear equation. This is a reasonable estimate, since the sigmoid function is near-linear around the half response point, and the data points cover approximately the first half of the total radiance range. Step 5 does the actual fit, using Mathematica's FindMinimum function (which finds local minima using a steepest gradient descent algorithm) on SE2, which is the summed squared error of fit over the data set. The last step returns the sigmoid function model for the data set, restoring the sign. The algorithm works well in practice. Figure 5.3 illustrates the procedure, and Figure 5.4 shows some examples of the computed sigmoid fits. Actually, steps 2 through 4

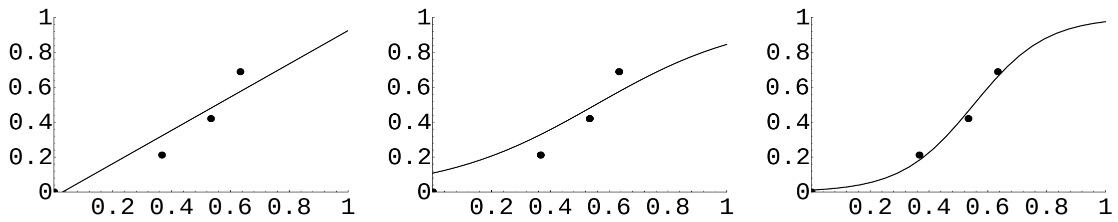


Figure 5.3: The procedure for fitting a sigmoid function to a constant-wavelength data set. On the X axes: relative radiance, on the Y axes: response. From left to right: the data points with a superimposed plot of the linear fit (step 2), the sigmoid fit with initial estimates for h and t (step 4), and the final fit after error minimization (step 5). The data set is for the $+Y - B$ function at a wavelength of 603 nm (cp. Figure 5.1 p. 76).

can be omitted and the minimization started with initial estimates of $0.5c$ for both h and t, but in that case it may take longer to converge. A constraint $t \geq 0.1$ is imposed on the fit; if it is violated, a different iterative error minimization procedure is used that keeps $t = 0.1$. This constraint serves to prevent slopes of the sigmoid function that are too steep to serve as realistic models of neural activation, but in practice this is seldom necessary.

After computing an array of sigmoid fit functions (indexed by wavelength) for each sampled wavelength and each of the six cell types, we can compute the response of each type to an arbitrary wavelength and radiance by interpolating between the values given by

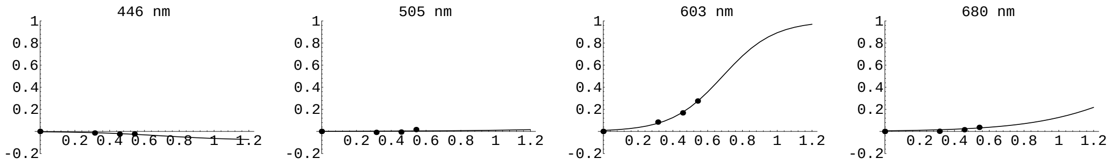


Figure 5.4: Some examples of computed sigmoid fits superimposed on the data points for the $+Y - B$ cell type, at the wavelengths listed above the plots. On the X axes: relative radiance, on the Y axes: response. The second plot is for a wavelength very close to the cross-over point between inhibition and excitation (cp. Figure 5.1 p. 76)

the sigmoid functions of radiance that are closest to the given wavelength:

$$i = Min_{j \in A} \left[Abs(\lambda_j - \lambda) \right] \tag{5.8}$$

where i is the index into the array of sigmoid functions A that we are looking for, λ is the wavelength for which we want to compute a response, and λ_j is the j-th wavelength for which we have computed a sigmoid function. Note that before this interpolation is applied, there is no smoothness constraint imposed on the parameters of the sigmoid fit from one sampled wavelength to the next, so in principle there could be serious discontinuities. In practice one would expect the parameters to change smoothly, and it turns out they do, to a considerable extent at least. I used Mathematica's standard cubic spline Interpolation function on the 7 sigmoid function values returned by the series of functions centered around i, adding additional zero points above and below the sampled wavelength range (and specifying 0 first and second derivatives for the very lowest and highest, to constrain the interpolation better):

$$a = Interpolation[\{\langle r_{i-3}, S_{i-3}(r)\rangle, \ldots, \langle r_{i+3}, S_{i+3}(r)\rangle\}, \lambda] \tag{5.9}$$

where a is the computed response (activation), $Interpolation[\{\ldots\}, x]$ computes a cubic spline interpolation function for the points in list $\{\ldots\}$ and evaluates it at x, r_i is the i-th radiance level and S_i the corresponding sigmoid function we have previously computed, and r is the relative radiance and λ the wavelength for which we want to compute the response. The resulting functions are shown in Figure 5.5 for the same relative radiances as in Figure 5.1 (p. 76), and additionally for maximum relative radiance $r = 1$.[13] We can

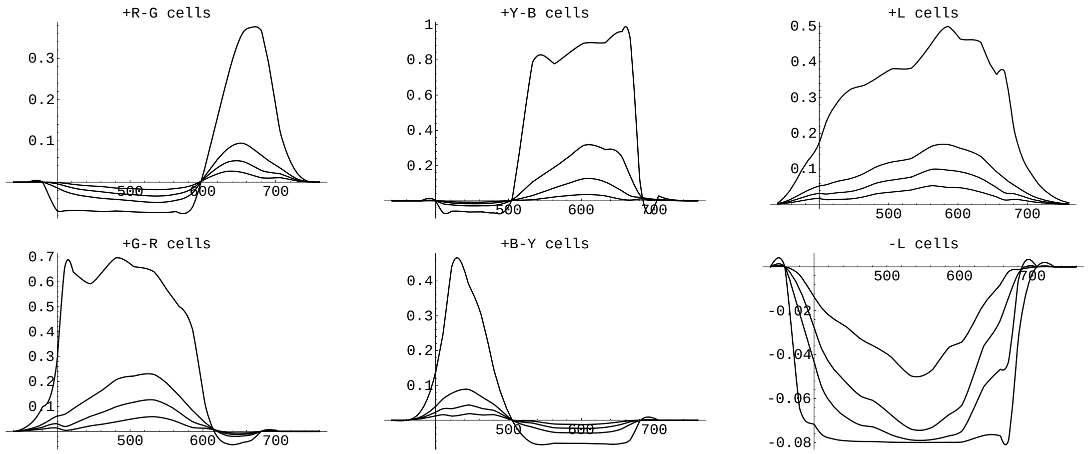


Figure 5.5: Activation functions for six types of cells based on the data from Figure 5.1 (p. 76). The radiance levels plotted for each type are the same as in that figure, with the addition of the maximum $r = 1$ plot. Note again that the Y scales are different.

see that the functions follow the data sets closely at the sampled wavelengths and radiances, but they are continuous and extend beyond the sampled data in both domains. We can see some irregularities in the maximum radiance level responses, which results from extrapolating the data considerably, combined with the absence of smoothness constraints as explained above. These irregularities will largely disappear in the next processing step, as explained in Section 5.3.4 (p. 90). It is interesting to note the clear saturation effect in most inhibitory response phases, and in the excitatory phases of the $+Y - B$ and $+G - R$ functions, due to the use of the sigmoid model. If we continued to compute function values at ever higher radiances, saturation would eventually also set in in the other functions. It is not clear whether the radiance range should be extended further to allow all functions to reach saturation levels. Alternatively, the radiance axis could be normalized for each function individually, or perhaps for opponent pairs of functions, but that is not in agreement with the assumptions I outlined above. Saturation effects are clearly in agreement with neurophysiological data, but to my knowledge there is no other color model available that incorporates them.

[13] Although I could have used the Interpolation function directly on the data set in two dimensions, that would not have resulted in the typical sigmoid behavior in the radiance domain, which I consider important.

5.3.2 Saturation levels revisited

In Section 5.3.1 (p. 75), I motivated my choice of 150 spikes/sec as the absolute saturation level response for all cell types. I will now show that choosing a different value basically amounts to a linear scaling of the response functions, so the actual value chosen does not matter all that much, as long as it is reasonable. The value should not be too much higher than the maximum represented in the data sets (less than 50), or the extrapolation will become too unreliable.

Following the procedure outlined in Section 5.3.1 (p. 75), I computed activation functions for the six cell types using a saturation response of 300 spikes/sec, or twice the value used before (the "negative saturation" level was likewise doubled). I then fitted the functions in the second set linearly to their counterparts in the first set, for instance for the +R-G functions:

$$R_{300} = mR_{150} + b \tag{5.10}$$

where R_{300} represents the +R-G function with a saturation level of 300 spikes/sec, R_{150} the same but with a saturation level of 150 spikes/sec, and m and b are scalars. The Root Mean Square (RMS) error of fit over a data set of 231 points spaced in a regular grid (11 in the radiance domain and 21 in the wavelength domain) was minimized using the steepest gradient descent algorithm of Mathematica's FindMinimum function, as a function of m and b. Table 5.2 summarizes the results, and Figures 5.6 through 5.11 graphically show the results and the errors of fit.

type	m	b	$RMSerror$
$+G-R$	2.13	0.0046	0.016
$+R-G$	1.77	0.0073	0.034
$+B-Y$	2.37	0.0039	0.017
$+Y-B$	1.33	0.0194	0.073
$+L$	1.87	0.0055	0.018
$-L$	1.24	-0.0067	0.010

Table 5.2: Linear coefficients and RMS error for linearly fitting 300 spikes/sec functions to their 150 spikes/sec counterparts (see text).

As is apparent from Table 5.2, the fit generally amounts to just scaling by a factor in each case (the offset b is negligible), and the error of fit is relatively low, with a typical RMS

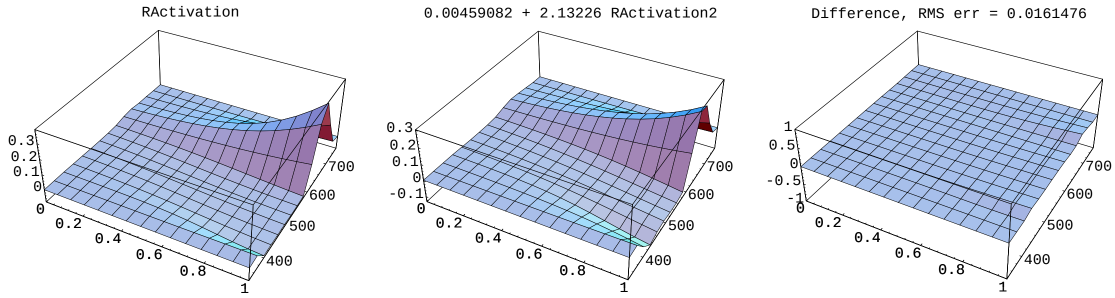


Figure 5.6: Linear fitting of R_{300} to R_{150} (see text). Left to right: R_{150}, $b + mR_{300}$, difference (error signal).

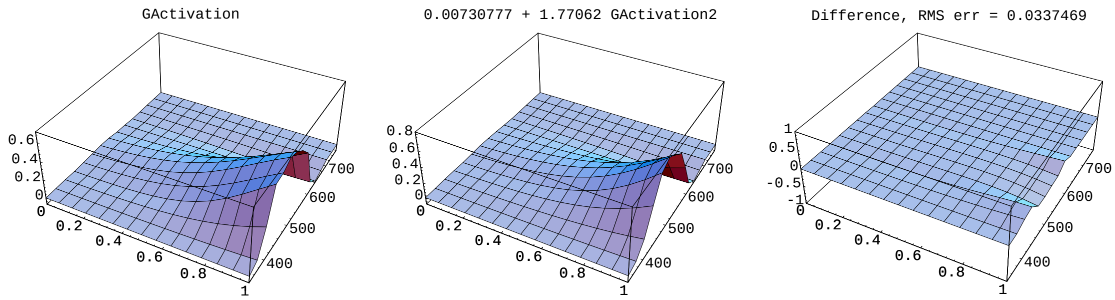


Figure 5.7: Linear fitting of G_{300} to G_{150} (see text). Left to right: G_{150}, $b + mG_{300}$, difference (error signal).

error of 1–2%. The worst fit is for the Y function, probably because it operates closer to saturation levels than any other, but it is still quite reasonable. The scaling factors do vary for the different functions, but that is not important (subsequent normalization will cancel the effect of different factors anyway).

5.3.3 Reducing six dimensions to three

As discussed in Section 4.4 (p. 51), [De Valois & De Valois 1975] suggest that the six types of LGN cells found can be grouped into three dimensions, when mirror image pairs[14] are combined into one dimension. That results in a red-green, a blue-yellow, and a brightness dimension which can be arranged in a "double cone"-type color space. We now turn to

[14]The terminology of "mirror images" should be interpreted loosely, with respect to the overall shape of the response profiles. The functions are by no means exact mirror images of each other.

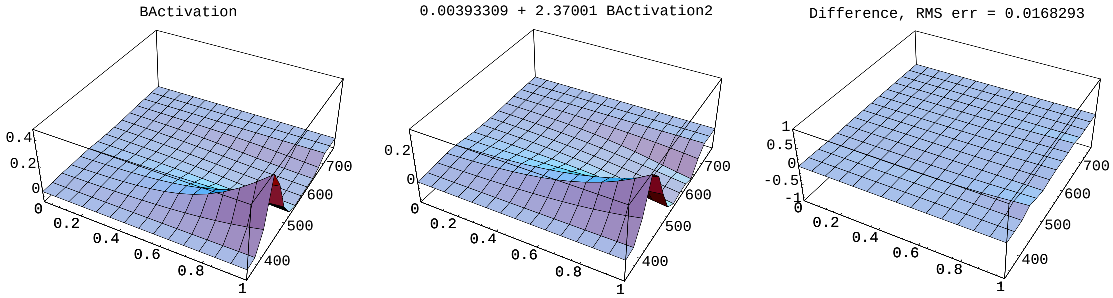


Figure 5.8: Linear fitting of B_{300} to B_{150} (see text). Left to right: B_{150}, $b + mB_{300}$, difference (error signal).

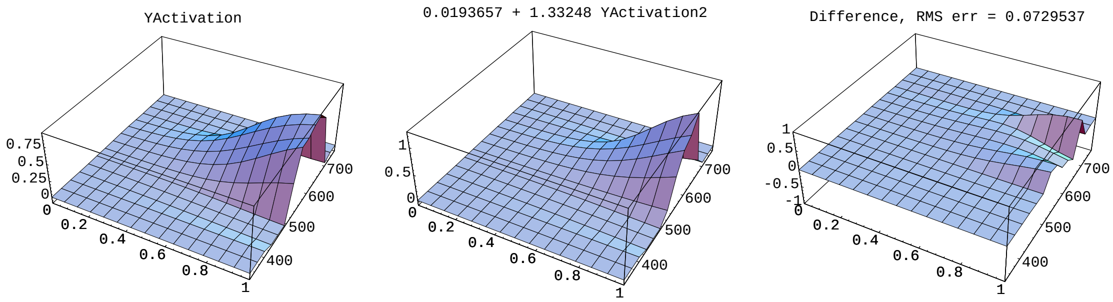


Figure 5.9: Linear fitting of Y_{300} to Y_{150} (see text). Left to right: Y_{150}, $b + mY_{300}$, difference (error signal).

the construction of such a 3-dimensional color space based on the six response functions described above. In view of the hypothesis that mirror-image coding of signals in the nervous system is a way to increase the dynamic range of the signal [Slaughter 1990], and following the suggestion of [De Valois & De Valois 1975], I will assume that the responses of mirror-image pairs of functions can be added to give one composite function:

$$GR = R - G \tag{5.11}$$

$$BY = Y - B \tag{5.12}$$

$$Br = L - D \tag{5.13}$$

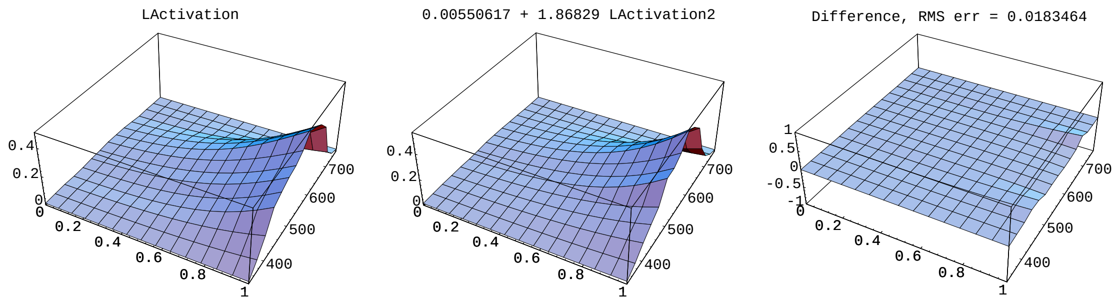


Figure 5.10: Linear fitting of L_{300} to L_{150} (see text). Left to right: L_{150}, $b + mL_{300}$, difference (error signal).

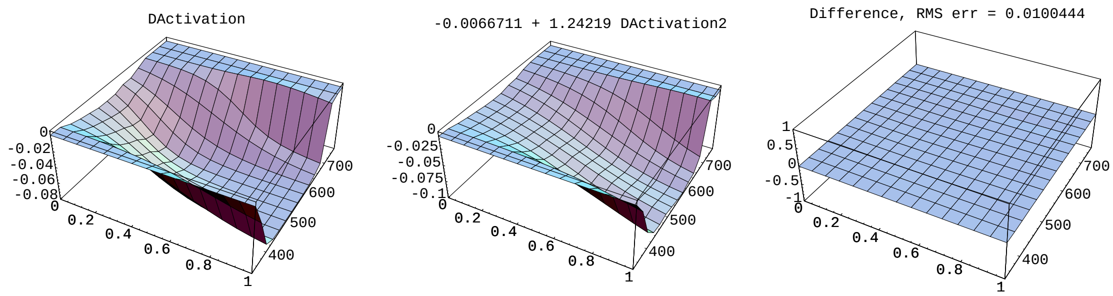


Figure 5.11: Linear fitting of D_{300} to D_{150} (see text). Left to right: D_{150}, $b + mD_{300}$, difference (error signal).

where GR is a new composite green–red opponent function, BY a new composite blue–yellow opponent function, Br a new composite brightness function (non-opponent), and R, G, Y, B, L, D represent the six response functions $+R - G$, $+G - R$, $+Y - B$, $+B - Y$, $+L$, and $-L$, respectively. The members of the pairs have to be subtracted rather than added because they are 180 degrees out of phase relative to each other, and we want the corresponding phases to add up rather than cancel each other. The order of the terms determines the sign of the phases of the composite functions, and is arbitrary. I will always use the order of equations 5.11–5.13 as the convention. Figure 5.12 (upper half) shows the resulting opponent functions at a relative radiance of 0.5. Some interesting properties of these functions are:

1. The GR zero crossing (upper left) at 605 nm is close to the maximum of the Y (positive) phase of the BY function (upper middle) at 609 nm, and the zero crossing

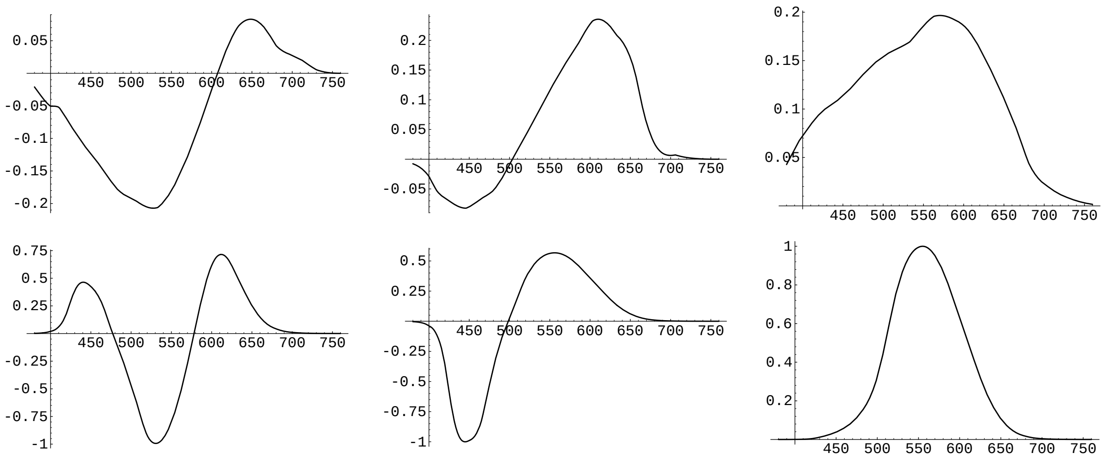


Figure 5.12: Upper: composite functions derived from the six neural response functions (see text). From left to right: GR (green–red), BY (blue–yellow), Br (brightness). Lower: the opponent functions and luminosity function defined by [Jameson & Hurvich 1955], see Section 4.4.5 (p. 62). On the X axes: wavelength, on the Y axes: response. The upper functions are plotted at a relative radiance of 0.5, the lower ones are undefined with respect to radiance.

of the BY function at 503 nm is close to the maximum of the G (negative) phase of the GR function at 528 nm.[15] If we interpret the composite functions as perceptual opponent functions in the style of [Jameson & Hurvich 1955] (Figure 5.12 lower), these zero crossing wavelengths accord relatively well with their estimates of the wavelengths of unique yellow (578 nm) and unique green (498 nm), i.e., wavelengths at which one of the two opponent functions is zero and the other has a non-zero response.[16] By this definition, there is no unique red or unique blue based on our opponent functions, since the other opponent function is non-zero over the entire red and blue phases. This differs from the model of [Jameson & Hurvich 1955] where there is no unique red, but there is unique blue.

2. The wavelength of maximum response of the Br function, 563 nm, is close to the wavelength of the CIE Y (photopic luminosity) function maximum, 555 nm.

[15] The zero crossings actually vary slightly with radiance, which may be related to the Bezold-Brücke effect [Wyszecki & Stiles 1982], but we can ignore that for the purpose of the present discussion.

[16] Of course, their model is merely that, and it is not without problems of its own. The existence of unique yellow and green wavelengths is well established, however. As with all psychophysical measurements, there is some interpersonal variability in the exact numbers, but not in the general trend.

3. The absolute maximum response of the green phase of the GR function is identical to the absolute maximum response of the yellow phase of the BY function, and the absolute maximum response of the red phase of the GR function is very similar to the absolute maximum response of the blue phase of the BY function. This is without any scaling on the 6 component functions or the resulting composite functions, other than the assumption stated above that all cell types will saturate at the same absolute firing rate, and the responses being scaled relative to that rate (150 spikes/sec). These maxima are different from the Br function maximum, but since we consider that to be an independent channel from the two color opponent functions, that is no problem.

In my opinion, these observations lend support to the assumption of one global maximum firing rate, and to the method of combining the six component functions into three. I therefore feel confident in using the three composite functions obtained as a neurophysiological basis for a 3-dimensional color space with a color opponent organization, interpreting the GR and BY functions as color opponent dimensions and the Br function as the brightness dimension. Later I will investigate the usefulness of this color space for the color naming problem. For now, I would like to note that this approach may provide an interesting bridge between the neurophysiology and the psychology of color perception (see Section 5.5, p. 106).

5.3.4 The sigmoid-of-linear activation model

The usual way to compute the response of a sensor to a certain stimulus of known spectral composition is by means of equations like 2.2 (p. 21) from Section 2.1:

$$a(E) = \int E(\lambda)\overline{s}(\lambda)d\lambda \tag{5.14}$$

where $a(E)$ is the response of the sensor, $E(\lambda)$ is the power spectrum of the stimulus, $\overline{s}(\lambda)$ is the sensor spectral sensitivity, and λ is wavelength. Since this depends crucially on the linearity of $\overline{s}(\lambda)$, we cannot apply this equation to the basis functions we derived in Section 5.3.3 (p. 86) because they are essentially non-linear. The solution I have adopted for this problem is what I will call the *sigmoid-of-linear* (SL) model. This model rests on the hypothesis that the basis functions of Section 5.3.3 can be modeled as the composition of a linear function and a sigmoid function:

$$a(E) = S(a'(E)) = S\left[\int E(\lambda)\overline{s'}(\lambda)d\lambda\right] \tag{5.15}$$

where a' is the linear function, S is the usual sigmoid function, and $\overline{s'}$ is the spectral sensitivity of a'. I believe this is a reasonable hypothesis, since saturation (which is modeled by the sigmoid function) is a function of the biochemical machinery of cells, and is determined by total absorbed energy (total "quantum catch") rather than by the spectral composition of the stimulus. In Section 5.3.1 (p. 75), I treated each constant-wavelength data set as independent from the others, and hence applied the sigmoid model at this level, but in the SL model I separate the wavelength dependent component of the response from the radiance dependent component. For a', we can use the activation functions from Section 5.3.3 (p. 86) at relative radiance 1.[17] That gives us the following equations for the SL basis functions:

$$\begin{aligned} GR_{sl}(r,\lambda) &= S'\left[r\, GR(1,\lambda), h_{GR}, t_{GR}\right] && (5.16)\\ BY_{sl}(r,\lambda) &= S'\left[r\, BY(1,\lambda), h_{BY}, t_{BY}\right] && (5.17)\\ Br_{sl}(r,\lambda) &= S'\left[r\, Br(1,\lambda), h_{Br}, t_{Br}\right] && (5.18)\\ S'(a,h,t) &= \frac{sign(a)}{1+e^{\frac{h-abs(a)}{t}}} - \frac{sign(a)}{1+e^{\frac{h}{t}}} && (5.19) \end{aligned}$$

where GR_{sl}, BY_{sl}, Br_{sl} are the SL basis functions derived from the previous set GR, BY, and Br; r is relative radiance in $[0,1]$; λ is wavelength; and h_{GR} and t_{GR}, etc., are the sigmoid parameters for GR_{sl}, etc. The sign function is as defined before, returning −1, 0, or 1, depending on the sign of its argument. If GR were linear in r, then $\int r\, GR(1,\lambda) = \int GR(r,\lambda)$, and likewise for the other two. The function S' is a straightforward relative of the usual sigmoid function, meant to deal with both positive and negative values, and subtracting the zero-offset (the value of the function at $a=0$) of the sigmoid with the given parameters. The latter is a technicality introduced to prevent discontinuities around the zero-crossings of opponent functions.

[17]The technical reason for choosing $r=1$ is that the ratio of the integral of the positive to the integral of the negative phase of the functions is greatest at that radiance, which results in a more "regular" shape of the normalized color space.

All that is left to do now is to determine values for the constants h_{GR}, t_{GR}, h_{BY}, t_{BY}, h_{Br}, t_{Br}. To do this, I fitted the SL functions GR_{sl}, BY_{sl}, and Br_{sl} to their non-linear counterparts GR, BY, and Br, minimizing the mean square error of fit as a function of h and t, over a regular grid of 441 points (21 coordinates in the radiance and 21 in the wavelength domain), using the steepest gradient descent algorithm of Mathematica's FindMinimum function. The results are shown in Table 5.3 and Figures 5.13 to 5.15.[18] With an RMS error of fit of 2–5% over the data set, the model fits rather well. Some of

function	h	t	*RMS error*
GR_{sl}	0.527639	0.178883	0.0352013
BY_{sl}	0.59439	0.158255	0.0483502
Br_{sl}	0.422241	0.202455	0.0188912

Table 5.3: Sigmoid parameters and RMS error for the three SL basis functions (see text).

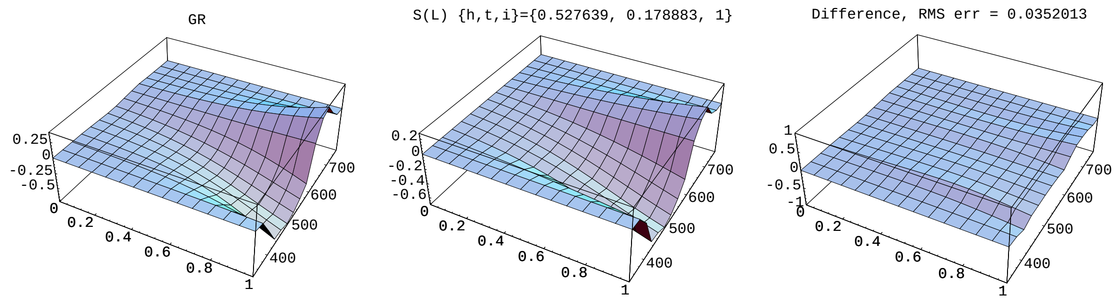


Figure 5.13: Fitting the SL function GR_{sl} to its non-linear counterpart GR (see text). From left to right: the non-linear function, the SL model fitted to it, and the difference between the two (error signal).

the error is no doubt attributable to the higher intensity ranges of the functions (as evident from the plots of the difference functions), which is the most extrapolated in the original non-linear functions, relative to the data sets they themselves were based on. I will therefore use the SL functions as the basis functions for the NPP color space, because they allow us to continue our exploration of the color-space properties in an analytical way.

[18] One could apply equation 5.15 to the 6 component functions before they are combined into opponent functions. I have tried this, but the resulting fits to the non-linear functions are no better than when applying the equation to the opponent functions directly.

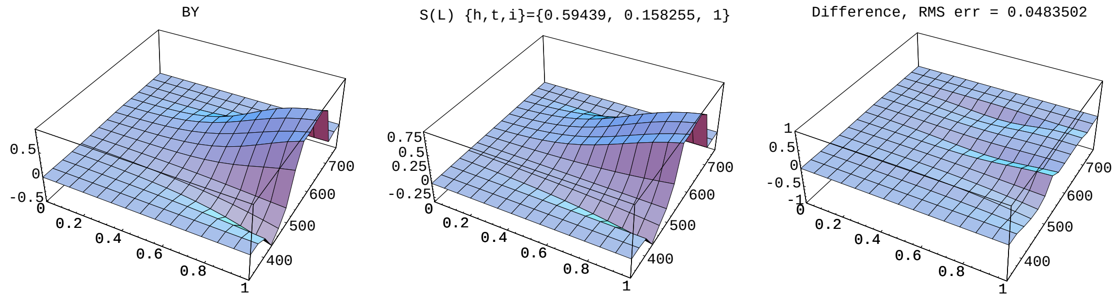


Figure 5.14: Fitting the SL function BY_{sl} to its non-linear counterpart BY (see text). From left to right: the non-linear function, the SL model fitted to it, and the difference between the two (error signal).

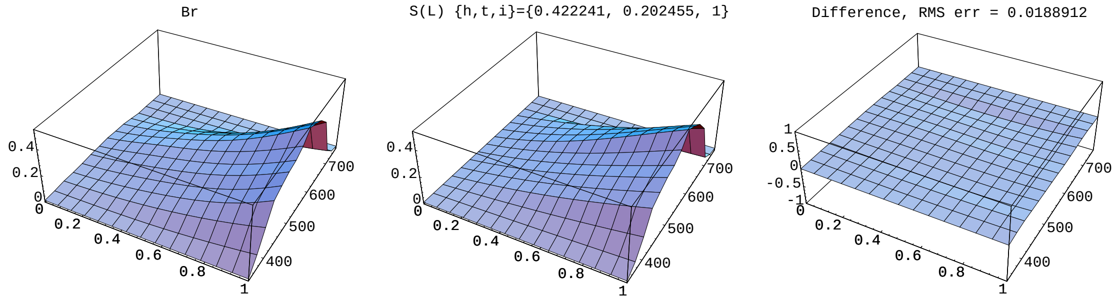


Figure 5.15: Fitting the SL function Br_{sl} to its non-linear counterpart Br (see text). From left to right: the non-linear function, the SL model fitted to it, and the difference between the two (error signal).

5.3.5 Normalization of the basis functions

We need one more step to turn the three SL functions into a convenient basis for a color space. Since I consider the GR_{sl} and BY_{sl} functions to be color opponent functions, I would like each function's two phases to cancel each other out in response to "white" (flat spectrum) light, i.e., to give a zero response. This corresponds to perceptual experience: white light does not seem to contain any color, either red, green, blue, yellow (the four primary colors), or any other one. I want the color model to reflect this directly. From Figure 5.12 (p. 89), we can see that the phases of the opponent functions are not equal in size.[19] I therefore apply a normalization step as follows:

$$A'_{sl}(r,\lambda) = \frac{A_{sl}(r,\lambda)}{Abs\left(S'\left[c_A \int^{+} A(1,\lambda)d\lambda\right]\right)} \tag{5.20}$$

where A_{sl} is any of the six SOL response functions as discussed in the previous section, A'_{sl} is its normalized version, r is relative radiance, λ is wavelength as before, S' is the sigmoidification function as in equations 5.16ff (p. 91ff), c_A is an energy equalization constant, and $\int^{+} g(x)dx$ is the integral over positive values of $g(x)$ only.[20] The constant c_A is discussed in more detail in Section 5.4 (p. 95). This normalization makes sure that the color-space dimensions are "square", since the maximum response of each function is obtained in response to a stimulus like

$$\begin{cases} \sigma(\lambda) &= 0,\ \lambda < \lambda_0 \\ \sigma(\lambda) &= 1,\ \lambda > \lambda_0 \end{cases} \tag{5.21}$$

where λ_0 is the zero-crossing wavelength of the function, i.e., the wavelength at which the response changes from inhibition to excitation.[21] The normalization step makes sure that these maximum responses are the same for all functions. When the normalized functions are added pairwise, we get the normalized versions of the SL basis functions that I will refer to as the SLN functions :

$$GR_{slN} = GR_0\left(R'_{sl} - G'_{sl}\right) \tag{5.22}$$

$$BY_{slN} = BY_0\left(Y'_{sl} - B'_{sl}\right) \tag{5.23}$$

$$Br_{slN} = Br_0\left[L'_{sl} - Abs\left(\frac{N_{max}}{P_{max}}\right) D'_{sl}\right] \tag{5.24}$$

with symbols as above, and N_{max} and P_{max} representing the maximum negative and positive response, respectively (see Section 5.3.1 p. 75). The normalization applied to the L and D

[19] The relevant measure is the area "under" the curve with respect to the Y axis, or the integral of the functions with respect to wavelength.

[20] For the $-L$ function $\int^{-}$ is used, or the integral over the negative values only, since that function is an inhibitor with an all-negative response.

[21] The equation given is for functions with inhibitive responses in the lower wavelength range, and excitative responses in the higher wavelength range. For the functions with the reverse profile, the inequalities have to be exchanged. For non-opponent functions, λ_0 can be thought of as ∞ for inhibitors and $-\infty$ for excitators.

functions is slightly different because the –L (or D) function is an inhibitor only, and has a much more limited range than the other five. The overall scaling factors GR_0, BY_0, and Br_0 are set such that the maximum response of the Br_{slN} function is 2, and the maximum for each opponent function is 1. The response to an equal-energy "white" spectrum (defined by $\sigma(\lambda) = c$) must therefore be close to $\langle 0, 0, 2 \rangle$. It is not exactly equal to that because of the effect of the inhibitory phases of the opponent functions. The gray axis of the color space (the path defined by the coordinates of equal-energy spectra of increasing radiance) must therefore be close to the line through $\langle 0, 0, 0 \rangle$ (black) and $\langle 0, 0, b \rangle$ (white). Later I will examine the shape of the gray axis in more detail. Figure 5.16 shows 3D plots of the resulting functions.

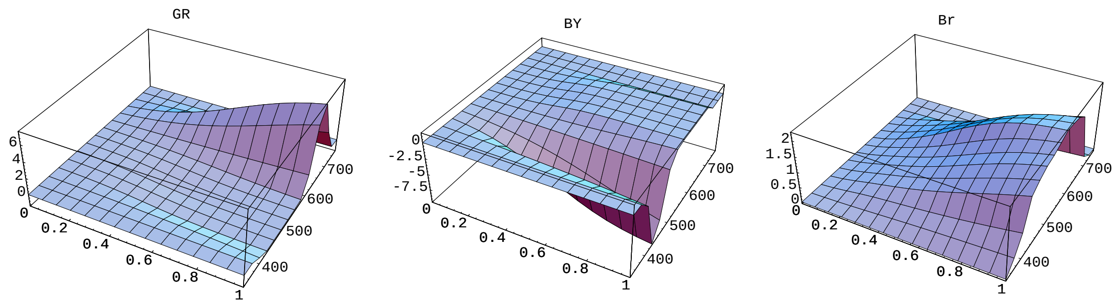


Figure 5.16: The normalized basis functions for the NPP color space (see text). From left to right: the GR_{SLN} (green–red) and BY_{SLN} (blue–yellow) opponent functions, and the Br_{SLN} (brightness) function. On the X axes: wavelength in nm, on the Y axes: relative radiance, on the Z axes: response (activation).

5.4 Visualizing Color Space Properties

To visualize the general "shape" of the NPP color space, I have computed the shape of the Optimal Color Stimuli (OCS) Surface in NPP space. We can represent all physically possible surface-spectral reflectance functions[22] in a solid known as the Object Color Solid. The surface of this solid represents the limit of physically realizable surface colors, known as Optimal Color Stimuli, and can be generated by computing the response of a given set

[22]Or transmittance functions, which makes no difference for our purpose.

of sensors to a continuum of two kinds of spectra:

> The spectral reflectance $\sigma(\lambda)$ is either zero or unity, and in moving through the visible spectrum, there are generally not more than two transitions between these values. Optimal color stimuli are imaginary stimuli in the sense that no actual object surfaces have reflectance curves with abrupt transitions of this kind. However, they are of considerable interest because they represent limiting cases of all (non-fluorescent) object-color stimuli. [...] Two types of curves must be distinguished; the first has zero reflectance (or transmittance) at wavelengths $\lambda < \lambda_1$ and $\lambda > \lambda_2$, the second at wavelengths $\lambda_1 < \lambda < \lambda_2$. [Wyszecki & Stiles 1982, p.181 ff.]

I have used the following differentiable approximations to these two types of reflectance functions:

$$\sigma_0(\lambda,\gamma,\theta) = \begin{cases} \frac{0.5}{(1-\lambda+\theta)^2}, & \lambda \leq \theta \\ 1-\frac{0.5}{(1+\lambda-\theta)^2}, & \theta < \lambda \leq \theta+\frac{\gamma}{2} \\ 1-\frac{0.5}{(1-\lambda+\theta+\gamma)^2}, & \theta+\frac{\gamma}{2} < \lambda \leq \theta+\gamma \\ \frac{0.5}{(1+\lambda-\theta-\gamma)^2}, & \theta+\gamma < \lambda \end{cases} \tag{5.25}$$

$$\sigma_1(\lambda,\gamma,\theta) = \begin{cases} 1-\frac{0.5}{(1-\lambda+\theta)^2}, & \lambda \leq \theta \\ \frac{0.5}{(1+\lambda-\theta)^2}, & \theta < \lambda \leq \theta+\frac{\gamma}{2} \\ \frac{0.5}{(1-\lambda+\theta+\gamma)^2}, & \theta+\frac{\gamma}{2} < \lambda \leq \theta+\gamma \\ 1-\frac{0.5}{(1+\lambda-\theta-\gamma)^2}, & \theta+\gamma < \lambda \end{cases} \tag{5.26}$$

where λ is wavelength in nm as usual, γ is the width of the "gap" in nm, and θ is the start of the "gap" in nm. Some typical examples of reflectance spectra generated by these functions are shown in Figure 5.17. If we assume a flat-spectrum (white) light source, defined by $\sigma(\lambda) = 1$, the light reaching the sensors has a spectrum identical to the reflectance

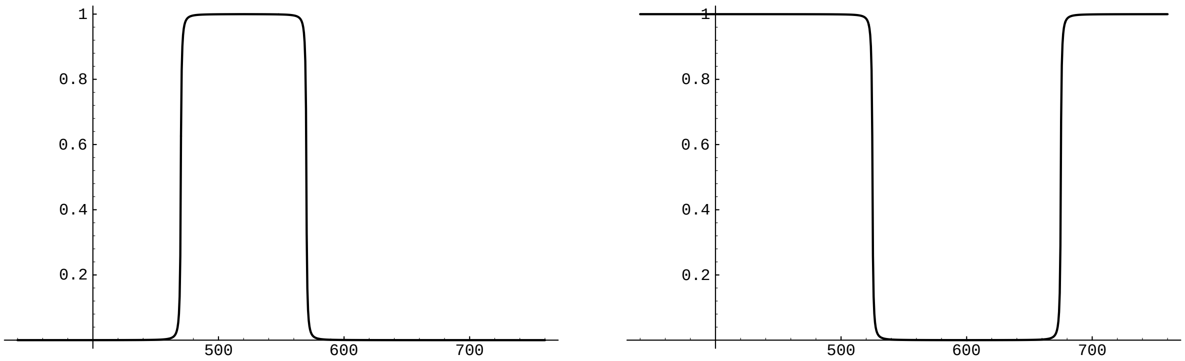


Figure 5.17: Some examples of the two kinds of spectra needed to generate Optimal Color Stimuli. Left: $\sigma_0(\lambda, 100, 470)$, right: $\sigma_1(\lambda, 150, 525)$. On the X axes: wavelength (nm), on the Y axes: relative reflectance. (After [Wyszecki & Stiles 1982, p. 181].)

function,[23] and we can compute Optimal Color Stimulus coordinates for *linearly* responding sensors as follows:

$$p_0 \quad = \quad List_{i=1}^{N}\left[\int_{\lambda_l}^{\lambda_u} \sigma_0(\lambda,\gamma,\theta)\,\overline{s_i}(\lambda)\,d\lambda\right] \tag{5.27}$$

$$p_1 \quad = \quad List_{i=1}^{N}\left[\int_{\lambda_l}^{\lambda_u} \sigma_1(\lambda,\gamma,\theta)\,\overline{s_i}(\lambda)\,d\lambda\right] \tag{5.28}$$

where $List_{i=1}^{N}$ is a list of expressions with index variable i ranging from 1 to N, N is the number of dimensions (basis functions) of the color space, $\overline{s_i}$ is the spectral sensitivity of each of the basis functions, and λ_l and λ_u represent the lower and upper limit of sensitivity for the sensors used, typically in the neighborhood of 300 and 800 for the human visual system, respectively. By varying γ and θ over the visible wavelength range, and plotting the resulting points p_0 and p_1, we can compute the shape of the OCS surface. It is made up of two "halves" that fit together like clam shells, corresponding to the set of points p_0 and p_1.

Now we need to choose sets of basis functions $\overline{s_i}$. If we choose the standard CIE XYZ functions (Section 2.1 p. 19), we get the result shown in Figure 5.18. This is the typical

[23] Again ignoring other factors such as atmospheric absorption and scattering, viewing angle, etc., as I did in equation 2.1 (p. 20).

"torpedo-like shape" that [Wyszecki & Stiles 1982] refer to. For the actual computations involved in creating figures 5.18ff, I used a computationally more efficient technique than suggested by equations 5.27ff, making use of the special properties of the functions σ_0 and σ_1 and using a list of partial integrals as a kind of cache. The surface color[24] in Figure 5.18

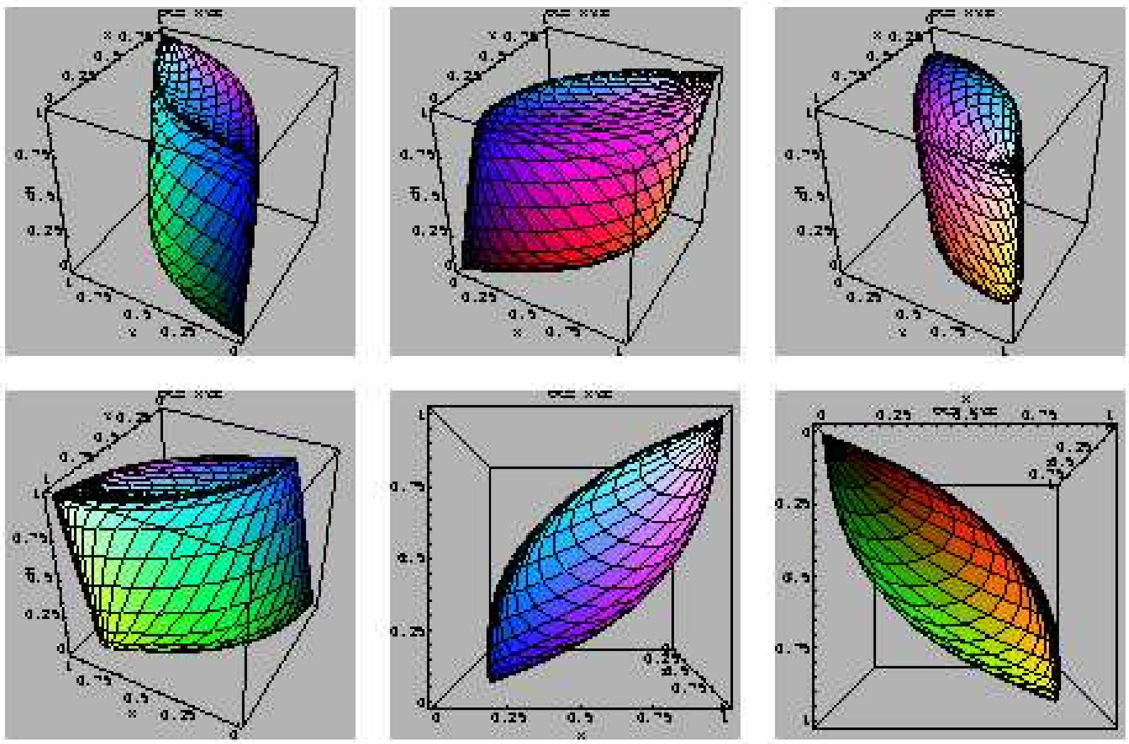


Figure 5.18: The Optimal Color Stimuli surface in the CIE XYZ color space (see text). The color rendering is only approximate, derived by using a transform from XYZ to RGB for typical computer monitors, and re-normalizing the result. In English reading order: four views rotating counter-clockwise around the surface, and a view from above and below.

is (necessarily) only an approximation, derived as follows:

$$\{r, g, b\} \quad = \quad Lt\left(\frac{\mathbf{M_{XR}} \cdot \{x, y, z\}}{\mathbf{C_X}}\right) \tag{5.29}$$

[24]For those having a color print available, or viewing this document on a color monitor. It is interesting to view this and related figures with a PostScript previewer like ghostview, where one can see the surface being built up in 3D. This gives a better appreciation of the shape, probably as good as possible short of an animated plot.

$$Lt(x) = \begin{cases} 0, & x < 0 \\ x, & otherwise \end{cases} \tag{5.30}$$

$$\mathbf{M_{XR}} = \left\{ \begin{array}{ccc} 1.26497 & -0.569038 & -0.198262 \\ -0.42012 & 0.797003 & 0.0196274 \\ 0.0275845 & -0.111426 & 0.46304 \end{array} \right\} \tag{5.31}$$

$$\mathbf{C_X} = \mathbf{M_{XR}} \cdot List_{i=1}^{3} \left[\int_{\lambda_l}^{\lambda_u} \sigma_1(\lambda, 0, 0)\, \overline{s_i}(\lambda)\, d\lambda \right] \tag{5.32}$$

where $\{r, g, b\}$ are $[0, 1]$ normalized RGB coordinates, Lt is a limiting function serving to limit RGB coordinates to the gamut of the display device, M_{XR} is a linear transform from XYZ to "typical computer monitor" RGB coordinates such as the ones given in [Rogers 1985] or [Hill 1990], $\{x, y, z\}$ are the CIE XYZ coordinates computed with equations 5.27ff (p. 97), and $\mathbf{C_X}$ is the set of RGB coordinates corresponding to a maximum radiance flat spectrum (white). The latter is used as a normalization factor for display purposes, which is basically Von Kries adaptation [Wyszecki & Stiles 1982]. It is clear from these equations that the displayed color has to be approximate, because of the limitations of the gamut of the display device, the "typical" transform used, and the inability to control for such things as gamma correction. Nevertheless, the rendered color is a reasonable indication of the "real" color corresponding to that particular point on the OCS surface.[25]

It is interesting to note that only a relatively small volume of the XYZ cube actually corresponds to physically realizable colors. The same is true of course if we compute the OCS surface in RGB coordinates, but the intersection of the volume enclosed by the OCS surface with the all-positive (i.e., displayable) RGB subspace is even smaller than in the XYZ case (Figure 5.19).

Choosing different basis functions results in different shapes of the OCS surface. The computation for the SL functions is somewhat more involved then for the CIE XYZ functions, because of the nonlinearities. The equations for computing points on the surface

[25] Were it not that there is no such thing as a real color, of course.

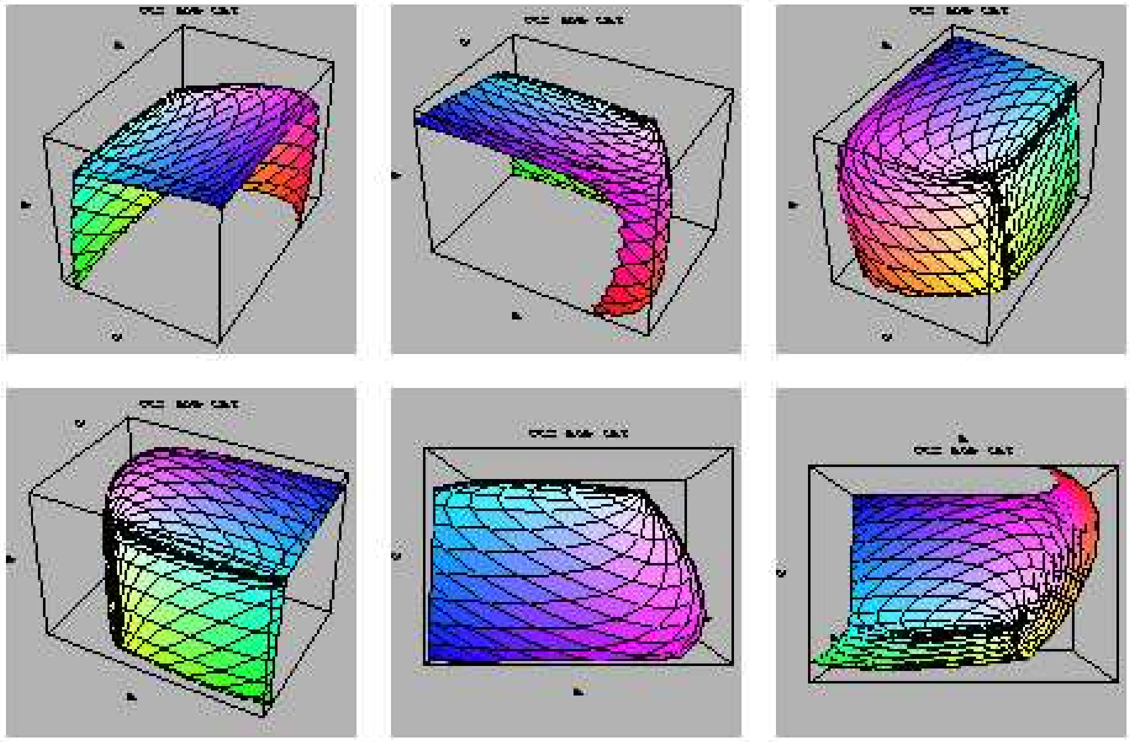


Figure 5.19: The OCS surface transformed into RGB space using the transform $\mathbf{M_{XR}}$ from equation 5.31 (p. 99). Viewpoints as in Figure 5.18. The parts of the surface that lie outside of the positive octant (i.e., outside of the gamut of a typical RGB computer monitor) have been clipped.

are

$$p_0 \quad = \quad List_{i=1}^{N}\left[sig_i\left(c_i \int_{\lambda_l}^{\lambda_u} \sigma_0(\lambda, \gamma, \theta)\,\overline{lin_i}(\lambda)\,d\lambda\right)\right] \tag{5.33}$$

$$p_1 \quad = \quad List_{i=1}^{N}\left[sig_i\left(c_i \int_{\lambda_l}^{\lambda_u} \sigma_1(\lambda, \gamma, \theta)\,\overline{lin_i}(\lambda)\,d\lambda\right)\right] \tag{5.34}$$

where symbols are as in equations 5.27ff (p. 97), sig_i being the sigmoid component, and lin_i being the linear component of the basis functions. Since the value of the sigmoid function is a non-linear function of its input, there is an issue here with respect to scaling the linear responses that was not relevant for the linear CIE XYZ functions. We have to determine values for the constants c_i. The method I have used is essentially scaling with respect to the equal-energy (EE) or "white" response. The argument goes as follows. We want to scale the brightness (Br) dimension to range from perceived black to white, without affecting the adaptive state of the visual system. Once I have determined a scaling factor for the Br dimension I will use it to scale the other dimensions as well. It is justifiable to use the same scaling factor for all color-space dimensions, since the functions I derived in sections 5.3.3ff (p. 86ff) preserve the ratios of responses among the cell types involved. Assuming (as usual) that the perception of "white" arises from the viewing of a stimulus with a flat equal-energy spectrum $\sigma_w(\lambda) = 1$, we can normalize responses with respect to this type of stimulus.[26] Since "white" is at the top of the Br dimension (always within the adaptive range), a flat spectrum "white" (maximum relative radiance) stimulus must result in the maximum response for the Br function, which is the function value at the wavelength of maximum response and at maximum relative radiance:

$$Br_{lin}^{max} = Max_{\lambda}[Br_{lin}(1, \lambda)] \tag{5.35}$$

The response of the linear Br function to a flat-spectrum "white" stimulus is given by[27]

$$Br_{lin}^{w} = \int Br_{lin}(1, \lambda)\, \sigma_w(\lambda)\, d\lambda \tag{5.36}$$

And the desired scaling factor is then just the quotient of the two:

$$c = \frac{Br_{lin}^{max}}{Br_{lin}^{w}} \tag{5.37}$$

The normalized response of a linear function to a stimulus $\sigma(\lambda)$ is then simply

$$f_{lin}^{n}(\sigma) = c \int f_{lin}(1, \lambda)\, \sigma(\lambda)\, d\lambda \tag{5.38}$$

[26] Sometimes a non-flat spectrum is used as the white reference, e.g. the CIE standard light sources A to D, but the differences are not important for our purpose.

[27] For integrating the linear response functions any relative radiance will do, as long as it is the same for all functions, since the result is scaled before sigmoidification. I have chosen a relative radiance of 1 for convenience.

The final SL normalized function value is

$$f_{sig}\left[f_{lin}^{n}(\sigma)\right] \tag{5.39}$$

as discussed above. Applying this normalization to the SL functions and computing the OCS surface results in the shape shown in Figure 5.20.

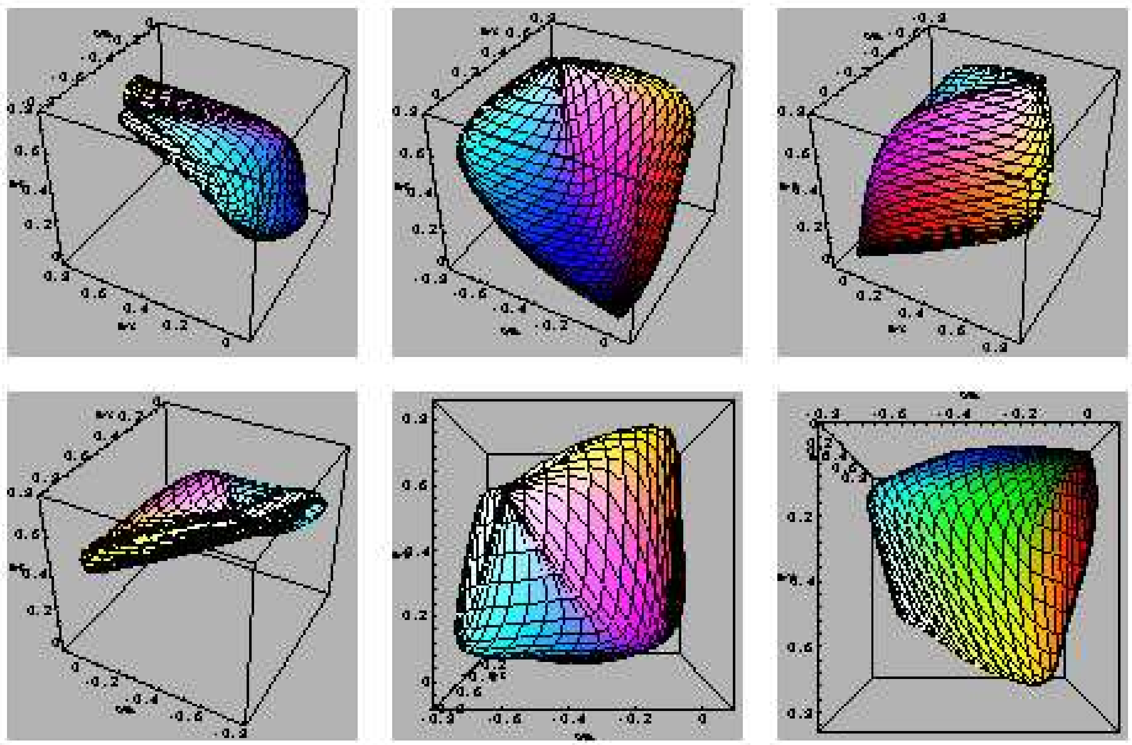


Figure 5.20: The Optimal Color Stimuli surface in the NPP color space, using the SL basis functions (see text). The color rendering is only approximate, computed the same way as for Figure 5.18. In English reading order: four views rotating counter-clockwise around the surface, and a view from above and below.

For the SLN functions the normalization process is as described for the SL functions, except the scaling factor is computed using the $+L$ function, and each of the six response functions is normalized individually before being combined into 3 dimensions. The resulting OCS surface is shown in Figure 5.21.

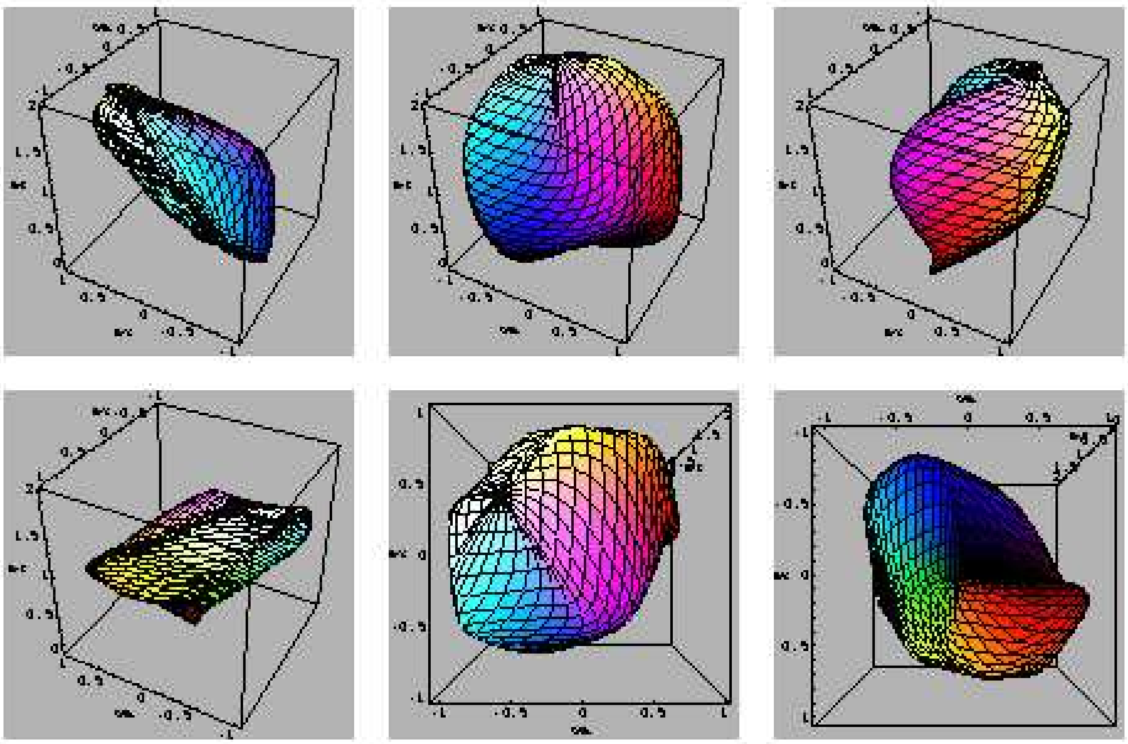


Figure 5.21: The Optimal Color Stimuli surface in the NPP color space, using the SLN basis functions (see text). The color rendering is only approximate, computed the same way as for Figure 5.18. In English reading order: four views rotating counter-clockwise around the surface, and a view from above and below.

Since the white point is slightly off-center (i.e., not on the $\langle 0, 0, Br\rangle$ axis) in the SLN space as shown above, I will apply a final linear transform to the color-space coordinates to compensate for this. Since the SLN coordinates are in the range $\langle[-1,1],[-1,1],[0,2]\rangle$ (Section 5.3.5 p. 93), the transform we want is given by

$$M = \left\{\begin{array}{ccc} 2 & 0 & gr_w \\ 0 & 2 & by_w \\ 0 & 0 & 2 \end{array}\right\}^{-1} = \left\{\begin{array}{ccc} 0.5 & 0 & -0.25gr_w \\ 0 & 0.5 & -0.25by_w \\ 0 & 0 & 0.5 \end{array}\right\} \tag{5.40}$$

where gr_w and by_w are the GR_{slN} and BY_{slN} coordinates of the white point, respectively. I will refer to the transformed coordinates as the SLN_M coordinates, and the corresponding color space as the NPP color space. The OCS surface in SLN_M coordinates is shown in Figure 5.22. The gray axis is now perfectly vertical, the coordinates of black being $\langle 0, 0, 0 \rangle$,

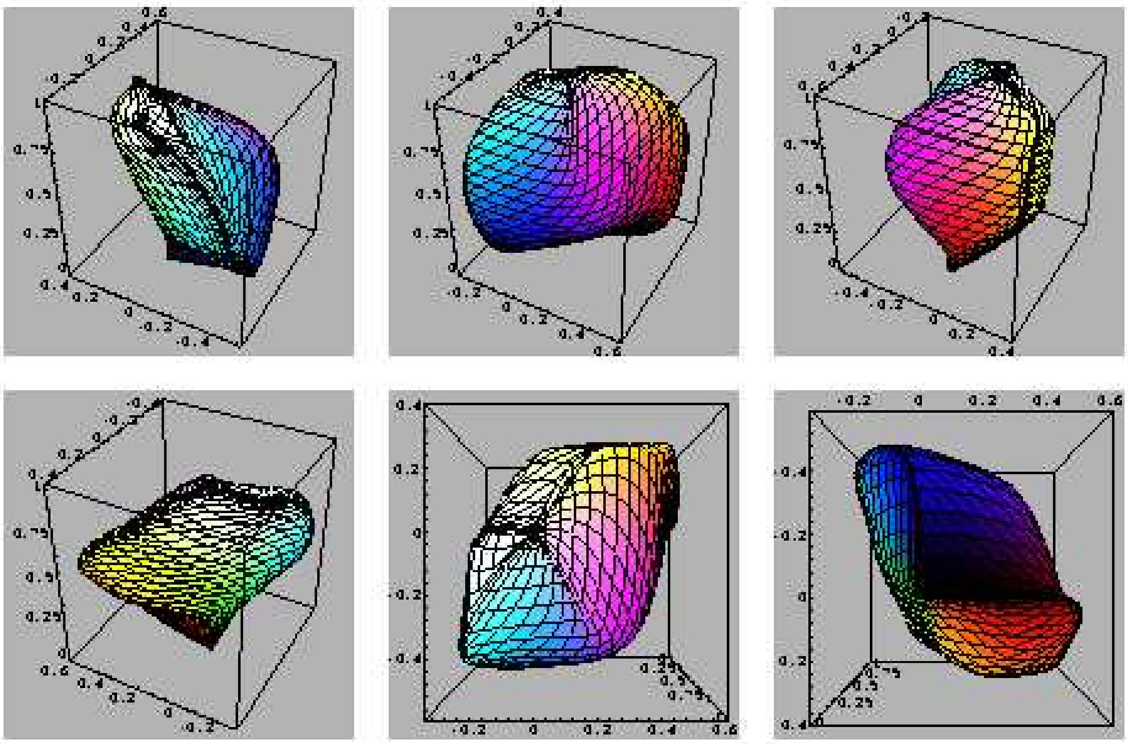


Figure 5.22: The OCS surface in SLN_M coordinates (see text).

and white being $\langle 0, 0, 1 \rangle$. The extrema of the OCS surface coordinates have now changed from $\langle [-1, 1], [-1, 1], [0, 2] \rangle$ to $\langle [-0.4, 0.6], [-0.6, 0.4], [0, 1] \rangle$, which makes the complete color stimuli solid fit inside a unit magnitude cube.

For comparative purposes, the OCS surface is represented in CIE L*a*b* coordinates (Figure 5.23), which represents an attempt to create a perceptually equidistant color space for reflected light (Section 4.4.2 p. 53). The L*a*b* model is based on psychophysical principles only, however, not on neurophysiological data as the NPP space is. Although the order of hues around the NPP and L*a*b* spaces is the same, there are marked differences in the overall shapes (e.g. the sharp protrusion of the L*a*b* space in the blue region is absent in the NPP space), and the relative positions and areas that certain colors occupy

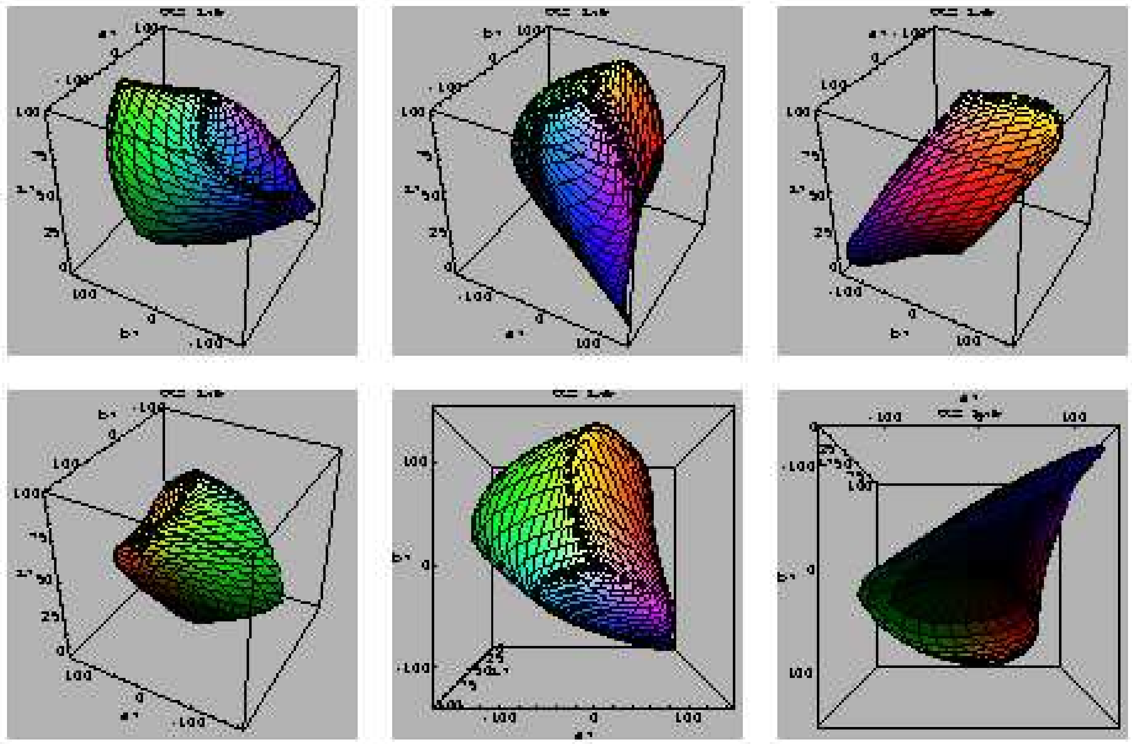


Figure 5.23: The OCS surface in CIE L*a*b* coordinates (see text).

on the respective surfaces.[28] These differences and their implications for theories of color perception remain to be investigated in detail, but that is outside of the scope of this dissertation. In the next section we will look at similarities between the NPP and other spaces to the Munsell system, another often-used psychological color-order system.

In [Wyszecki & Stiles 1982], only black and white hand-drawn approximations of the OCS shape (in different color spaces, but not NPP of course) are shown, and I am not aware of any attempts to define the complete shape and its surface color analytically as I have done. Later, I will use the OCS surface as a frame of reference to investigate the distribution of basic color categories. It is well suited for that purpose, since it represents the limit of all physically possible surface reflectances (giving rise to color perception when viewed under an appropriate light source), and I can represent it in the neurophysiologically-based NPP

[28]A note of caution: the color rendering of Figure 5.23 and the like is only approximate, as explained before, so we should not give too much weight to the visual appearance of the OCS surface in these figures. Nevertheless, the color rendering is a reasonable approximation, the algorithm used is exactly the same for all spaces shown, and a picture is certainly worth a thousand words for our purpose.

space.

5.5 Some Interesting Properties of the NPP Space

The OCS surface in NPP space as shown in Figure 5.22 (p. 104) is similar in shape to the outer surface of the Munsell color solid, a psychological color-order system (or color-*appearance* system) based on the principles of color perception [Wyszecki & Stiles 1982, p. 510]. This is remarkable in view of the fact that the Munsell system is a purely psychological or "impressionistic" color model used especially by artists, and the NPP space is defined in an entirely analytic way, based on neurophysiological measurements and physical object surface properties alone. As I mentioned in Section 5.3.3 (p. 86), the NPP space thus provides a tentative bridge between the psychology and the neurophysiology of (an aspect of) color perception, which has proved difficult to obtain to date [Boynton 1990].

A direct quantitative comparison between the NPP and Munsell spaces is not possible, because the Munsell space is not defined mathematically, but rather as a set of example color chips arranged in a certain way. We can do some qualitative comparisons however:

- The vertical dimension in the Munsell space is the gray axis, ranging from black at the bottom to white at the top. This is also true for the NPP space (Figure 5.24). For all practical purposes, we can consider the NPP gray axis to be straight (Section 5.6 p. 108).

- Hue varies as a polar-coordinate system around the gray axis in Munsell, and can be expressed the same way in NPP. The spacing of the hues around the hue circle is approximately the same in both spaces (Figure 5.25).

- The overall shape of the NPP color space[29] is remarkably similar to the Munsell space as well, e.g., when comparing planes of constant Munsell hue with planar sections through the NPP space (Figure 5.26).

[29] Whenever I say "the shape of the color space" I really mean the shape of the OCS surface as represented in the color space, of course. The color space as such has no shape. The OCS surface is very well suited for this purpose, since it represents the limit of physically realizable colors. The Munsell color system is also based on the *appearance* of surface colors. The combination of the OCS surface and the color space can be thought of as the representation of physical or "outside world" constraints in the internal perceptual mechanism.

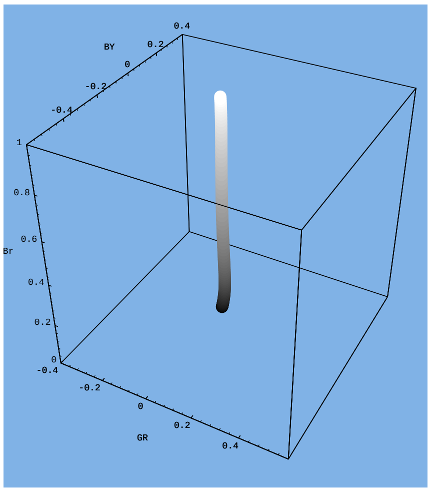


Figure 5.24: The gray axis in NPP color space, using the SLN_M functions (see text). As in Munsell space, the gray axis is (as good as) straight.

For comparative purposes I have made similar sections through the OCS surface in XYZ space[30] (Figure 5.27) and in L*a*b* space (Figure 5.28). In XYZ space, both of the sections are much more symmetrical in shape than the Munsell diagram shows, which is also true for the "green-red" section in L*a*b* space. The "blue-yellow" section in L*a*b* space is more similar to the Munsell diagram than the one in XYZ space, but not quite as similar as the NPP section is. The asymmetry of the sections through Munsell and NPP space is relevant, as it corresponds to greater saturation ranges for some colors versus others, which is a psychological property of object colors that is not reflected to the same extent in the XYZ or L*a*b* spaces.

- Saturation is expressed as distance from the gray axis in the Munsell system, and can be expressed the same way in NPP (Figure 5.25).

In Section 5.6, I will discuss how we can derive precise measures for these three variables from the NPP space.

The OCS surface in NPP color space shows some clear "lobes" corresponding to the locations of blue and red, and to a lesser extent yellow and green (Figure 5.21 p. 103). The locations of white and black are also clearly marked, being respectively at the top and

[30]For this purpose, the XYZ space has been rotated appropriately to make the gray axis vertical and to position the four primary colors in positions similar to the ones they occupy in the NPP space.

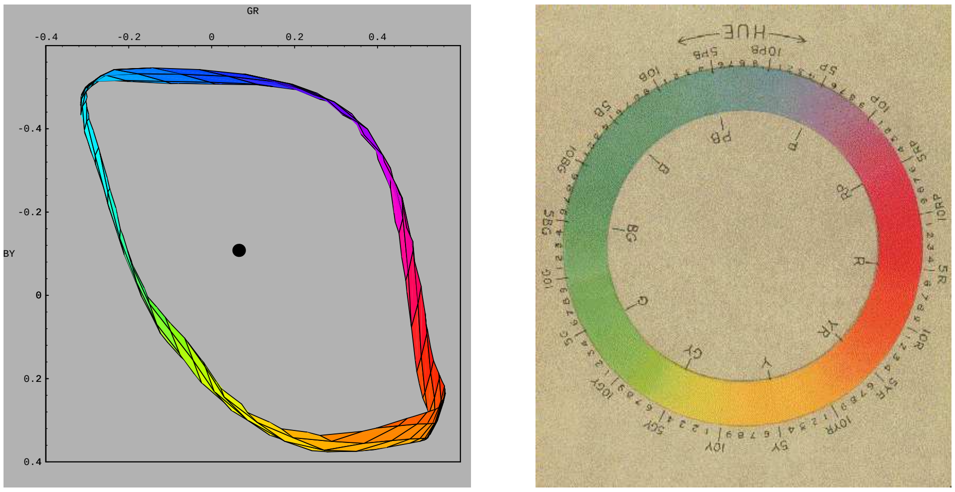


Figure 5.25: Spacing of hues around the NPP color space is similar to hue spacing in the Munsell psychological color space. Left: a horizontal section through NPP space (using the SLN_M functions) at Br=0.5 (half way up the brightness axis). The black dot in the middle indicates the position of the gray axis. Right: the Munsell hue circle as rendered artistically in [Munsell 1946, Plate I].

bottom of the hull. These observations lend some preliminary support to the hypothesis that there is a neurophysiological basis for the "basicness" of basic colors like the six primary colors just mentioned. We will look into these matters in more detail later.

5.6 Psycho-Physical Variables

[De Valois & De Valois 1975] suggest that in the 3D color space derived from the six LGN cell response functions, hue would be coded in a circular fashion (ranging through blue, green, yellow, red, and back to blue), saturation as distance from the center of the hue circle (making hue and saturation specifiable with a polar-coordinate system in the plane), and brightness along an axis perpendicular to the hue circle. This kind of color space is well known from early work of, e.g., Munsell [Munsell 1946, Birren 1969a] and Ostwald [Birren 1969b], and, as I have noted in the preceding section, the NPP space is similar to those in it overall organization. It is relatively easy to define psycho-physical measures on the NPP space, if one considers the axes to be orthogonal and one uses the standard Euclidean distance metric.

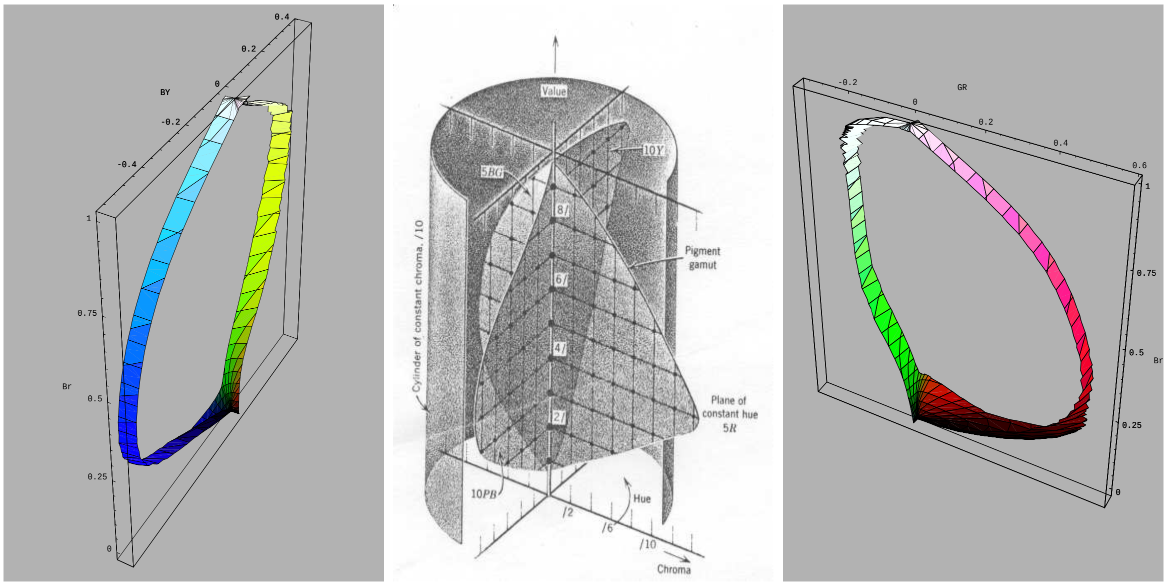


Figure 5.26: Comparison of the shape of the Munsell color solid with the shape of the NPP color space (see text). Center: schematic diagram of the Munsell color solid showing four planes of constant Munsell hue (from [Wyszecki & Stiles 1982, p. 510]). Left and right: a planar section through NPP space at $GR_{slN_M} = 0$, and one at $BY_{slN_M} = 0$, respectively. Note the remarkable similarity in shape and color of the latter two with the planes outlined in the Munsell diagram (to compare the Munsell color coding with the rendered NPP colors, refer to Figure 5.25 (p. 108).

I define saturation in the NPP space in the traditional way, as the distance between a color point and the gray axis in an equal-brightness plane going through the point:

$$S(gr, by, br) = \sqrt{(gr - 1st[gray(br)])^2 + (by - 2nd[gray(br)])^2} \tag{5.41}$$

where $S(gr, by, br)$ represents the saturation of the color represented by NPP coordinates $\langle gr, by, br\rangle$, and $gray(br)$ is a parametric equation defining the gray axis in SLN coordinates (before transformation via transform M given in equation 5.40 p. 103):

$$gray(br) = \langle -0.295935\, br, -5.55112\, 10^{-17} + 0.211884\, br, br\rangle \tag{5.42}$$

These linear equations produce an RMS error of fit of $\langle 0.0760, 0.1139, 0\rangle$ over 11 equally-spaced points along the gray axis, or $\langle 0.0380, 0.0569, 0\rangle$ normalized to a unit cube, which is

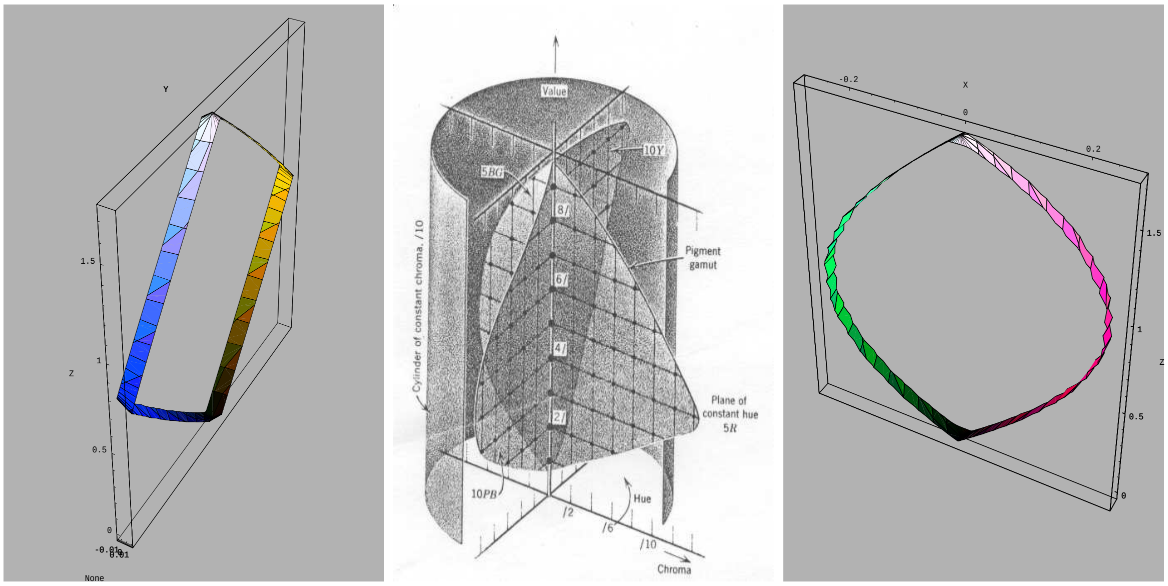


Figure 5.27: Planar sections through the OCS surface in (rotated) XYZ space, parallel to the (rotated) X and Y axes. Compare to Figure 5.26.

quite reasonable. When using a second order polynomial fit:

$$gray(br) = \begin{array}{l} \langle \quad 0.0259402 - 0.104432\, br - 0.0921772\, br^2, \\ \quad -0.0321185 - 0.0894883\, br + 0.141793\, br^2, \\ \qquad\qquad\qquad br \qquad\qquad\qquad \rangle \end{array} \tag{5.43}$$

the RMS error is reduced to $\langle 0.0238, 0.0341, 0\rangle$, or $\langle 0.0119, 0.0170, 0\rangle$ when normalized to a unit cube.

I define hue also in the usual way, as the angle between a color point and some reference point, with the gray axis at the origin, in a projective plane of constant brightness. The reference hue is defined as 0 degrees. As the reference point, I have chosen the hue of unique green, i.e., that wavelength at which the BY function is zero, and the GR function has a non-zero green response. This is a good reference, since it is easy to define in an unambiguous way, and the corresponding wavelength does not change with intensity. Unique green in SLN coordinates corresponds to a wavelength of 504.934 nm, which corresponds to NPP coordinates $\langle -0.279332, -0.0959707, Br\rangle$ (Figure 5.29). The equation for hue in the NPP

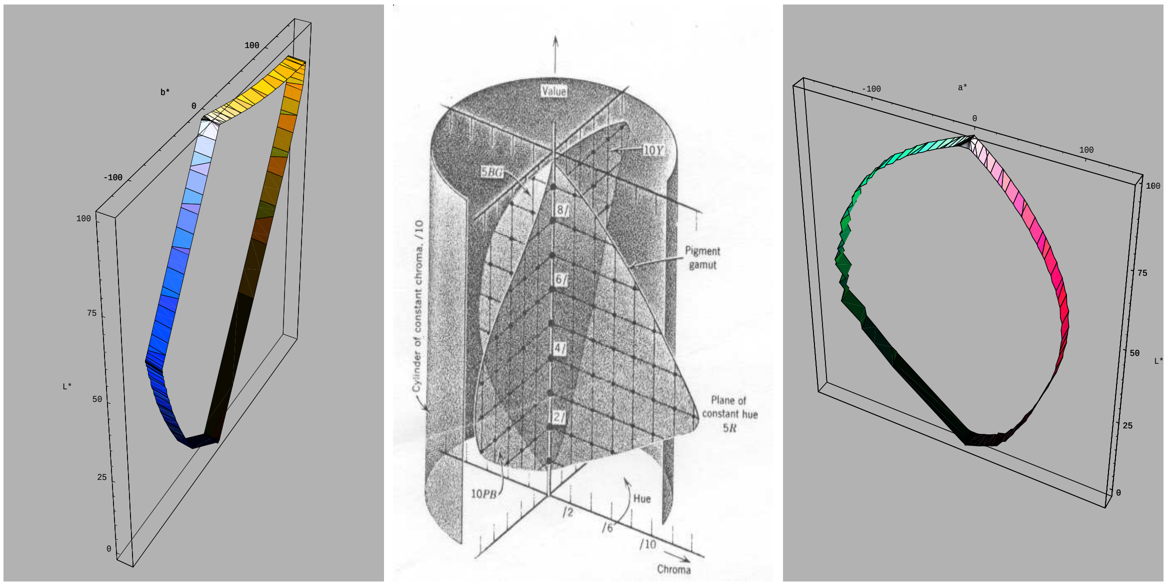


Figure 5.28: Planar sections through the OCS surface in L*a*b* space, parallel to the a* and b* axes. Compare to Figure 5.26.

space is thus:

$$H(gr,by,br) = \begin{cases} (2\pi - H_r + sin^{-1}[\frac{by-2nd[gray(br)]}{S'(gr,by,br)}]) \ mod \ 2\pi, & gr \geq 1st[gray(br)] \\ (3\pi - H_r - sin^{-1}[\frac{by-2nd[gray(br)]}{S'(gr,by,br)}]) \ mod \ 2\pi, & gr < 1st[gray(br)] \end{cases} \quad (5.44)$$

where $H(gr, by, br)$ is the hue of the color point $\langle gr, by, br\rangle$, and H_r is the reference hue. The resulting values are in the range $[0, 2\pi]$.

Brightness, finally, is just the third dimension of the NPP space:

$$B(gr, by, br) = 3rd\,[M \cdot [gr, by, br]] \quad (5.45)$$

which gives us a complete transform from NPP space to a psychophysical HSB (Hue-Saturation-Brightness) space based on it. Since the computation of hue is inherently unreliable with low saturation values, a threshold is applied to saturation, below which all hues are considered undefined.[31]

[31] I have used a threshold of $\frac{1}{256}$, which corresponds to the smallest representable value with an 8-bit

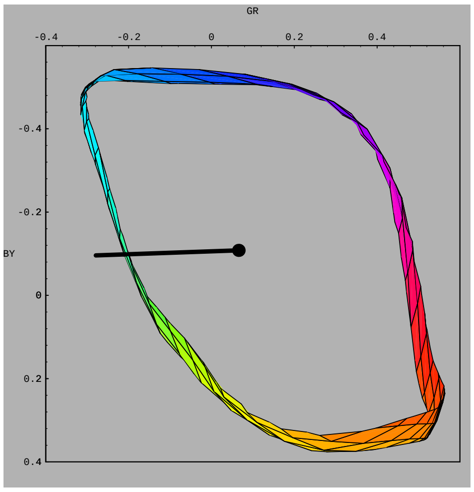


Figure 5.29: The unique green hue reference direction in the NPP color space (black line, see text).

5.7 Learning a Color Space Transformation

Now that we have constructed the NPP space, we need to be able to transform data coming from color sensors (typically an RGB color camera) into it, if we want the NPP space to be usable for robotic agents (Chapter A p. 179) or for general computer vision work. Since all color cameras in use today are based on the CIE XYZ standard, we need to transform XYZ coordinates to NPP coordinates. This will proceed in two steps: a transform from XYZ to the 6 linear functions from Section 5.3.4 (p. 90), followed by sigmoidification and a final linear transform to make the gray axis vertical, as described in the same section. Although it might be possible in principle to go directly from XYZ to NPP coordinates, I have chosen the method just described because it follows the theoretical construction of the NPP space more closely, and it provides additional advantages for the learning (optimization) procedure described below.[32]

Since it is not possible to determine a simple linear transform from XYZ to the 6 linear functions within a reasonable margin of error[33], I have used the error back-propagation

resolution.

[32]Whether one sees the artificial neural network technique described below as learning or as optimization depends largely on one's background and one's theoretical likes and dislikes. I will freely use "learning" in the remainder of this section because that term is traditionally used in the neural networks literature, but the reader should feel free to substitute "optimization" if (s)he finds the other term offensive. Please contact the author for an Emacs lisp function to enforce the usage of choice.

[33]As determined by some experimental attempts to do so.

algorithm to determine such a transform, as commonly used in artificial neural networks research [Rumelhart et al. 1986]. In contrast with many applications of the backpropagation technique I did not use the network as a classifier, but rather to learn six simultaneous functions of three real variables (the CIE X, Y, and Z coordinates), implementing a transform from one space to another. Using the networks this way imposes much stricter requirements on the obtained transform than using them as a classifier does, since both the domain and range are continuous-valued, and there are no "bins" in the range within which differences in function values are not important, as is the case in classifier nets.

The error back-propagation algorithm is defined for N-layer[34] feed-forward networks with consecutive layers fully connected, no connections going to nodes in layers other than the neighboring one(s), and using the sigmoid node activation function (Figures 5.30 and 5.31).

Since this kind of network is strictly feed-forward, the computation at each layer amounts to the composition of a linear transform with the sigmoid node activation function:

$$\mathbf{O} = S(\mathbf{M}_j \cdot \mathbf{I}) \tag{5.46}$$

where $\mathbf{M}_j$ represents the weight matrix for stage j, $\mathbf{I}$ represents the vector of inputs to this stage, i.e., the activations of the nodes at stage $j-1$, $\mathbf{O}$ represents the vector of outputs of this stage, i.e., the activations of the nodes at stage j, and S represents the usual sigmoid activation function from equation 5.1 (p. 76). After the network has been trained we can collect the matrices $\mathbf{M}_j$ and implement the transform via equation 5.46, disregarding the neural network simulation machinery. If we keep the networks "rectangular", i.e., with the same number of nodes at each layer, we can also determine the inverse transform by inverting the matrices $\mathbf{M}_j$ (if possible) and the sigmoid function S. This turns out to be possible for the transforms I have computed, so I have indeed used rectangular nets, padding the input and/or output with small fixed random values if necessary. This padding does not seem to have any noticeable effect on the convergence of the backpropagation algorithm or the quality of the computed transform (as determined by the procedure outlined below).

[34]There is some confusion as to whether N should refer to the number of layers of *nodes* or *connections*. If the layer of input nodes is not counted, these measures are identical. I will refer to the number of layers of *nodes* by N, however.

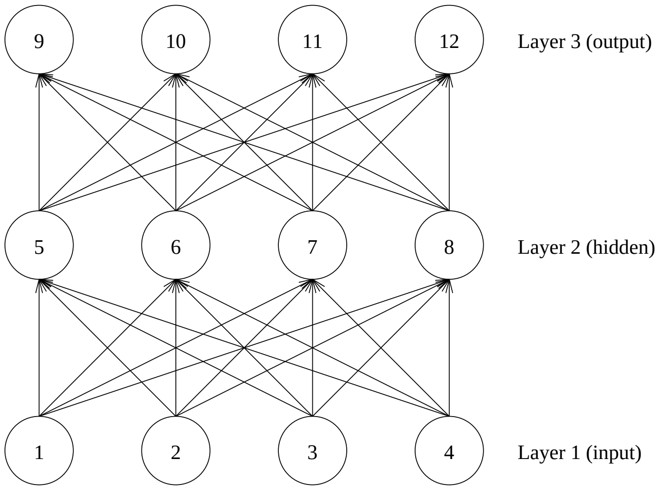


Figure 5.30: Typical feed-forward network topology used with the error back-propagation algorithm, for N (number of layers) equal to 3: the input layer, one hidden layer, and the output layer. For $N \geq 2$ there are $N - 2$ hidden layers. Each node in layer i is connected via a feed-forward link (indicated by the arrows) to every node in layer $i + 1$, if any, with layers being numbered starting from the input layer. This particular network has a completely regular structure with m nodes per layer, for a total of $N \times m$ nodes, which need not be the case in general. Computation proceeds by setting the *activation* level of each input node to a certain value, and propagating the values forward through the weighted links from one layer to the next, until they reach the output nodes (Figure 5.31). The output of the network is the vector of activation values of the output nodes after the feed-forward cycle is complete, at which time the next input vector can be presented to the net. During the error back-propagation stage, the error between the output vector and a vector of desired outputs (the teaching vector) is propagated backwards through the layers, and link weights are adjusted proportional to their relative contribution to the error. See [Rumelhart et al. 1986] for details.

One of the most critical problems in using the error backpropagation algorithm to learn a particular transform is the selection of a proper training set. Since this learning technique is a supervised one, one needs to construct a set of input-output pairs that covers the n-dimensional input and output subspaces (assuming n input and output nodes) sufficiently well to produce a smooth transform, without significant discontinuities and generalizing well to new input/output pairs. In our particular case, the natural thing to do might seem to be to construct a training set based on the pure frequency responses of the XYZ and

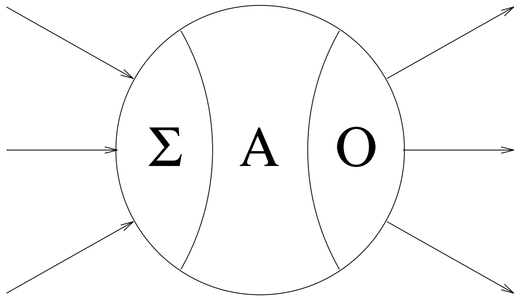


Figure 5.31: A typical backpropagation network node is made up of three parts: a function Σ combining all inputs to the node, an activation function A determining the node's activation as a function of the combined inputs, and an output function O determining the output value that is propagated along outgoing links, as a function of the node activation. In our case, Σ computes the sum of weighted inputs to node i minus an offset or bias value: $\sum_j w_{ji}a_j - \theta_i$ (w_{ji} is the weight of the link going from node j to node i, and a_j is the activation of node j), A is the sigmoid function from equation 5.1 (p. 76), and O is simply the identity function. See [Rumelhart et al. 1986] for details.

NPP functions, by varying the frequency over the visible wavelength range:

$$in(\lambda) = \langle \overline{x}(\lambda), \overline{y}(\lambda), \overline{z}(\lambda) \rangle \tag{5.47}$$

$$out(\lambda) = List_{i=1}^{6} \left[\overline{lin_i}(\lambda) \right] \tag{5.48}$$

with symbols as in Section 4.4.2 (p. 53) and equation 5.33 (p. 100). This type of training set turns out not to work well, and some analysis reveals why: it does not contain any training points on the gray axis (since no pure frequency signal results in an equal response of all three XYZ functions) or any points in the purple region of the color space (since purple is a "mixture of red and blue" which cannot be obtained from a pure frequency signal), for instance. It is therefore no surprise that the transform does not work well in these (and other) areas of the color space. What we need to do is to create a training set containing points corresponding to complex stimulus spectra, including grays, purples, and all other kinds. Since spectra can be thought of as real-valued functions of wavelength (usually, though not necessarily, continuous and differentiable), the set of possible spectra is infinite. Fortunately we can exploit some constraints on the physically possible surface reflectance functions to generate a number of representative spectra, analogous to the way we constructed the OCS surface in Section 5.4 (p. 95). We know that all physically possible spectra must result in points enclosed by that surface (including points on the gray axis), so these are the spectra of interest for constructing a training set. Based on the technique described in Section 5.4 (p. 95), I have created an extended one to generate a set of spectra covering both the OCS surface and the space contained within it. The spectra are given by the following function:

$$\sigma(\lambda, \gamma, \theta, \rho_1, \rho_2) = \begin{cases} \rho_1, & \lambda \leq \theta \\ \rho_2, & \theta < \lambda \leq \theta + \gamma \\ \rho_1, & otherwise \end{cases} \tag{5.49}$$

where λ is wavelength in nm as usual, γ is the width of the "gap" in nm, θ is the start of the "gap" in nm (as in Section 5.4 p. 95), and ρ_1 and ρ_2 are two reflectance levels in $[0, 1]$. By varying γ and θ over the visible wavelength range and ρ_1 and ρ_2 over the $[0, 1]$ interval, we can generate a subset of all possible reflectance spectra, some of which are shown in Figure 5.32.[35] By varying γ in 10 equal steps from 0 to 400, θ in 10 equal steps

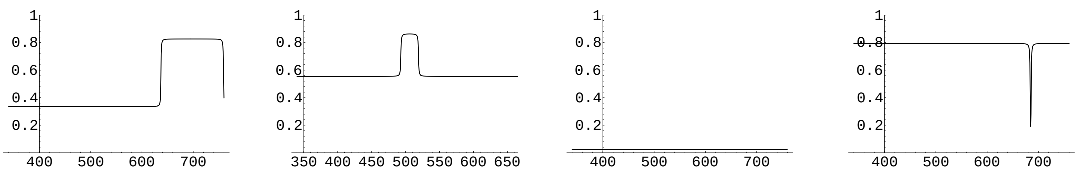


Figure 5.32: Some example spectra generated by varying the parameters γ, θ, ρ_1 and ρ_2 in equation 5.49. From left to right, parameter values are $\langle 122, 637, 0.34, 0.83\rangle, \langle 26, 493, 0.56, 0.87\rangle, \langle 192, 760, 0.02, 0.03\rangle$, and $\langle 2, 684, 0.80, 0.10\rangle$ (see text). On the X axis: wavelength in nm, on the Y axis: relative spectral reflectance.

from 380 to 780, and ρ_1 and ρ_2 each in 7 equal steps from 0 to 1, we obtain a set of 4900 spectra that cover the OCS surface and the space enclosed by it fairly well, as shown in Figure 5.33. In this figure the XYZ coordinates corresponding to the spectra are computed

[35] This subset is itself infinite since the parameters are real-valued, but its cardinality is obviously less than that of the set of all possible spectra. By discretizing the domains of each of the parameters we can bring the cardinality down to a finite integer number.

as in equation 5.27 (p. 97) ff, and these coordinates are the input values for the training set. The output values for the training set are computed analogously using the 6 linear NPP functions. Comparing Figures 5.18 (p. 98) and 5.33 we can see that there are no points

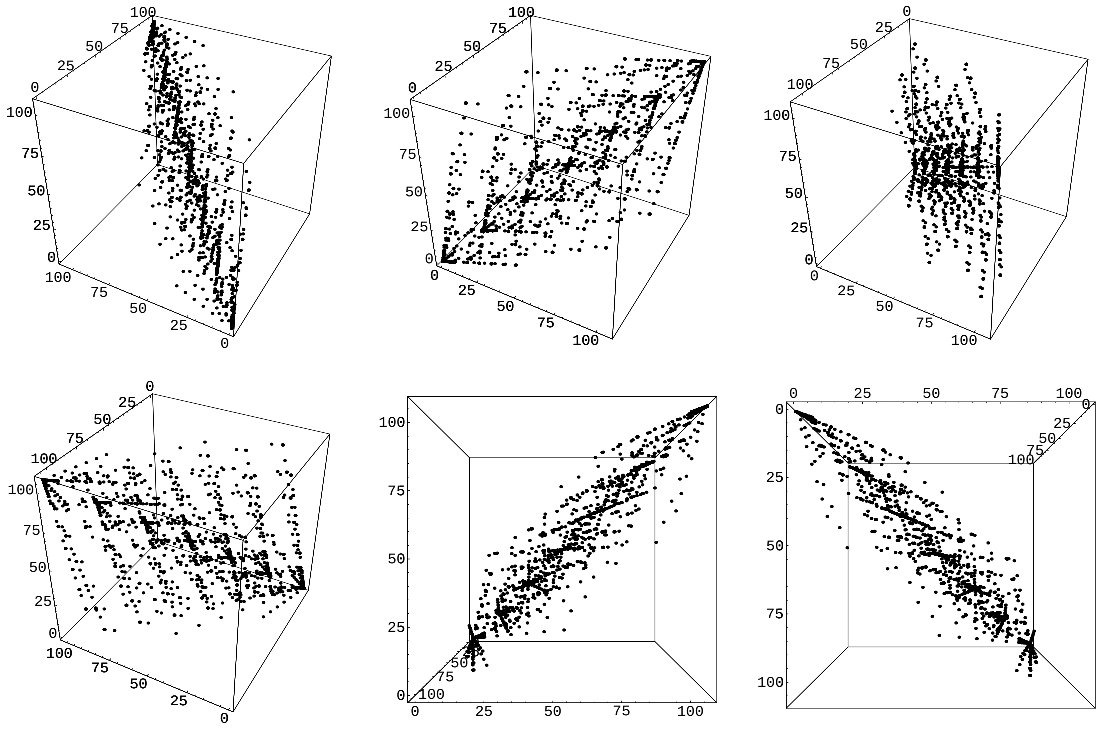


Figure 5.33: Backpropagation training set input points. Each point corresponds to a spectrum defined by equation 5.49. Comparing to Figure 5.18 (p. 98), we can see that the points cover both the OCS surface and the space enclosed by it.

outside the OCS surface, which is expected of course. Such points do not correspond to physically possible reflectance spectra, and hence are of no interest to us.

As a measure of convergence of the backpropagation algorithm, I have computed the RMS error vector between the teaching vectors and the output vectors over the complete training set. The results for two, three, and four-layer networks are shown in Figure 5.34. I have used the Rochester Connectionist Simulator for these computations. The results can be influenced by some additional parameters such as a momentum term (preventing the weights from changing too abruptly in the course of learning), a temperature parameter (determining the derivative of the sigmoid node activation function), and the learning rate

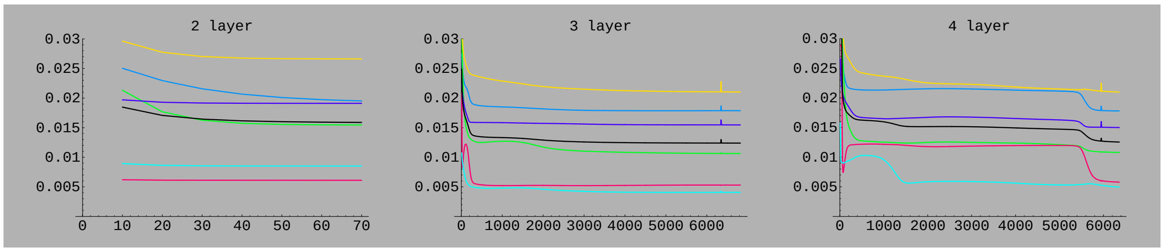


Figure 5.34: RMS error vector between the teaching vectors and the output vectors over the complete training set (see text). On the X axis: iterations, on the Y axis: RMS error. Values for each of the six output nodes are color-coded, the average is shown in black.

(determining how fast the gradient descent learning proceeds). These parameters were set by trial and error to 0.66, 0.44, and 0.03, respectively. In general, deciding when to stop the learning is a relative of the general halting problem, and thus not a computable function. Since we are doing optimization for one particular problem, we can hand-pick a local minimum in the error landscape by examining the error plots over a long enough period, but we can never really be sure that we have obtained a global minimum. Since the magnitude of the error derivative tends to approach zero over the course of learning, we can be reasonably confident that whatever local minimum we pick is not too far from the global minimum, although networks with more layers tend to display sudden changes in the error functions, as evident for the 4 layer network in Figure 5.34 (between 5000 and 6000 iterations).

Of course the quality of the obtained transform is more important than the RMS error over the training set. To evaluate this, I will first compare the position of the gray axis in the NPP space as derived in Section 5.3 (p. 75) to the gray axis as computed by the network-derived transforms (Figure 5.35). Visual inspection reveals that the gray axis extends into the negative brightness values for the 2-layer network transform, which is of course undesirable, but not for the 3 or 4-layer transforms. In all cases, the transformed gray axis is vertical, and somewhat more curved than the original. The exact black and white coordinates are $\langle\langle 0,0,0\rangle,\langle 0,0,1\rangle\rangle$ for the NPP space, and $\langle\langle -0.019, 0.019, -0.11\rangle, \langle -0.019, -0.016, 0.98\rangle\rangle$ for the 2-layer network transform, $\langle\langle 0.019, -0.0035, 0.0074\rangle, \langle -0.014, 0.0038, 0.98\rangle\rangle$ for the 3-layer, and $\langle\langle 0.017, -0.0025, -0.0016\rangle, \langle -0.0034, 0.00088, 0.99\rangle\rangle$ for the 4-layer network transform (to 2 significant digits). For 11 equally spaced points along the NPP gray axis,

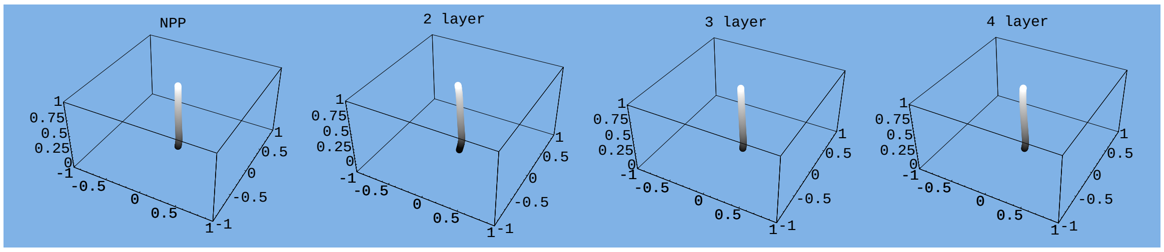


Figure 5.35: The gray axis in NPP space (left) and as computed via the XYZ to NPP transform using a 2-layer (second from left), 3-layer (third from left) or 4-layer network (right). On the X axes: Green-Red, Y: Blue-Yellow, Z: Brightness.

the RMS error vectors relative to the transformed points are $\langle 0.0149, 0.0128, 0.0368\rangle$ for the 2-layer net, $\langle 0.0110, 0.00241, 0.0105\rangle$ for the 3-layer, and $\langle 0.0139, 0.00515, 0.00964\rangle$ for the 4-layer net, which confirms the aberration in the brightness dimension for the 2-layer net. The other two are close enough to make the difference insignificant.

Another important measure of the quality of the transform is what it does to the OCS surface, which I will continue to use as a frame of reference for studying the spacing of basic color category foci in the NPP and other spaces. Figure 5.36 shows the OCS surface as transformed by the 4-layer network, the one with the lowest overall RMS error. It should be compared to Figure 5.22 (p. 104), which shows the same surface in the (directly computed) NPP space. We can see that the general shape of the transformed space (Figure 5.36) is close to that of the directly computed space (Figure 5.22 p. 104), with a vertical gray axis, similar spacing of hues around the perimeter of the surface, and comparable overall dimensions. There are some differences too, e.g. the (approximately) red-purple region of the transformed space is stretched and seems to bulge more relative to the directly computed version. All in all, I consider this result satisfactory enough to be used in practice. We should also be aware that the directly computed version is based on measurements of the Macaque visual system, and the transformed one on a transform of human standard observer functions, which may account for some of the observed differences.[36] The exact parameters of the transform are given in Appendix B.

[36] In addition, I have used the most common 5 degree 1931 CIE functions, and there are several modified versions as well as 10 degree versions available. I have not investigated the effects of using any of these sets of functions on the accurateness of the transform.

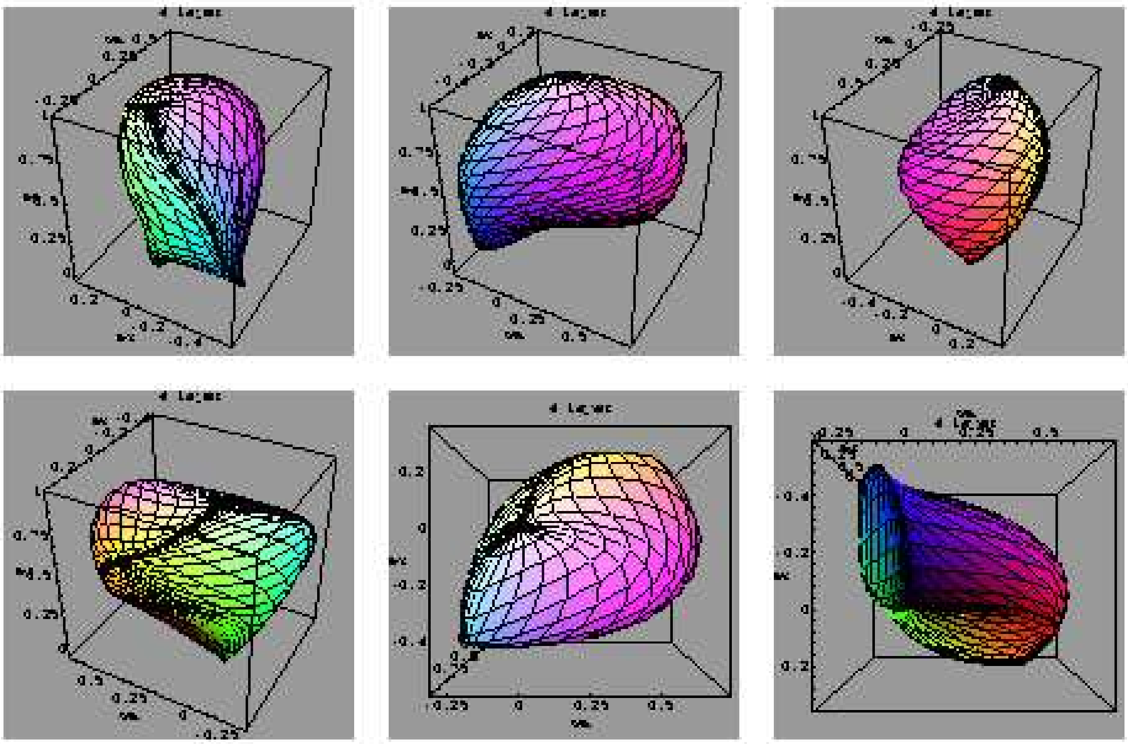


Figure 5.36: The OCS surface as computed via the 4-layer network XYZ to NPP transform (see text). Compare to Figure 5.22 (p. 104), which shows the directly computed version.

Chapter 6

From Color Space to Color Names

The second part of the mapping $\mathcal{N}$ takes us from a color space to a set of (basic) color names, each with a "typicality" or "goodness" measure associated. In Shepard's terminology, each color name corresponds to a "consequential region" in psychological space. Categorizing a stimulus then amounts to inferring the consequential region to which it belongs [Shepard 1987], together with its goodness value. In this chapter I describe the procedure I used for fitting a particular category model to the experimental color naming data of [Berlin & Kay 1969], present a theoretical evaluation of the model, and outline a model for learning color names.

6.1 The Normalized Gaussian Category Model

[Shepard 1987] provides an elegant argument for his claim that the probability of generalization of an existing (known) category to a new (unknown) stimulus is a monotonic function of the normalized distance in psychological space of the unknown stimulus to known stimuli belonging to the category. He further specifies that this function can be approximated by a simple exponential decay or, under certain circumstances, a Gaussian function. The distance metric is either Euclidean, resulting in circular (or spheroid) contours of equal generalization, or a slight variant that results in elliptic (or ellipsoid) contours of equal generalization.

Following Shepard's suggestion, I have used a variant of the Gaussian normal distribution as a category model, which I will refer to as the *normalized Gaussian* model. The usual normal curve in one variable (as used in probability theory) is given by

$$G(x) = \frac{1}{\sigma\sqrt{2\Pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{6.1}$$

where σ is the standard deviation (determining the "width" of the curve), and μ is the mean or expected value (determining the location of the maximum) (Fig. 6.1). Since the normal

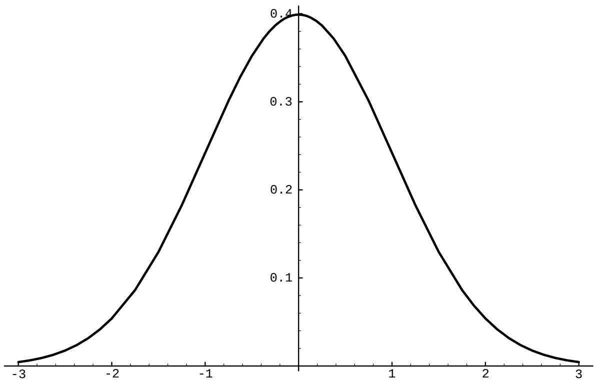


Figure 6.1: A plot of the normal curve in one variable, with $\mu = 0$ and $\sigma = 1$ (see text).

curve is used as a probability density function, it has the special property that $\int G(x) = 1$. The term $x - \mu$ in equation 6.1 is the Euclidean distance of the one-dimensional point x to the mean μ. To derive the normalized Gaussian function, we drop the factor $\frac{1}{\sigma\sqrt{2\Pi}}$, since we don't need the interpretation as a probability density function, and we substitute the general N-dimensional Euclidean distance function for the distance term, which gives us

$$G_n(x) = e^{-\frac{1}{2}\left(\frac{\sqrt{\sum_{i=1}^{N}(x_i-\mu_i)^2}}{\sigma}\right)^2} \tag{6.2}$$

with symbols as in equation 6.1. An example of a two-dimensional version of this function is shown in Figure 6.2.[1] This function has a number of interesting properties for use as a basic color category model:

[1]By an N-dimensional normalized Gaussian function I mean a function of N variables.

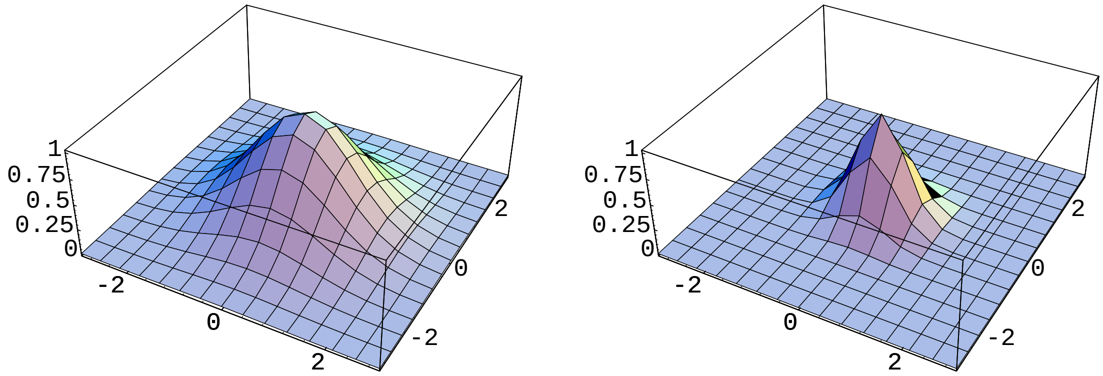


Figure 6.2: An example of a two-dimensional normalized Gaussian function with $\mu = (0, 0)$ and $\sigma = 1$ (left) or $\sigma = 0.5$ (right).

- The maximum value occurs at μ and is unity, regardless of the value of σ.
- The "width" of the curve is a function of the parameter σ only.
- The value decreases monotonically (but not linearly) as a function of the distance to μ.
- It is strictly positive (non-zero) everywhere except at infinite distance from μ.
- It has a simple mathematical definition, using only two parameters μ and σ.

These properties allow us to interpret the normalized Gaussian function as a category model, with μ interpreted as the location of the center or focus of the category, and the function value interpreted as the goodness value (or alternatively, the fuzzy membership value) of a stimulus represented by its color space coordinates. By modulating the value of σ we can affect the size of the volume of the color space that is included in the category, relative to some threshold function value. The maximum goodness (or membership) value is unity, as is common in fuzzy set models, but we can get a goodness value for any point in the color space, no matter how far removed from the focus. This is important for our purpose, as I will show below. The non-linear decrease of goodness with distance from the focus reflects the general shape of psychological categories as discussed by [Shepard 1987], and it can result in category extensions different from those obtained with a simple nearest-neighbor criterion, since neighboring categories may have different values of σ. The representation is also economical, since a category can be represented by only two parameters.

6.2 Locating the Berlin and Kay Color Stimuli in the Color Space

Before we can apply the category model described in the previous section to Berlin and Kay's color naming data, we need to quantify the stimulus set they used for their experiments. As described in Section 4.2, they used a set of 329 Munsell color chips consisting of 40 equally spaced hues at 8 equally spaced brightness (Value) levels each, all at maximum saturation (Chroma), and a gray scale consisting of nine equally spaced brightness (Value) steps. They asked subjects to point out both the extent and the foci of the basic color categories of their native language on the array of color chips, viewed under a light source approximating the CIE standard source A (Figure 6.3).[2]

In the following sections I will always use the CIE XYZ space, the CIE L*a*b* space, and the NPP color space for comparative purposes. The XYZ space is in a sense "the mother of all RGB spaces", since the various RGB spaces are simple linear transforms of it. It is generally accepted as an approximation to the spectral sensitivities of the human cone photoreceptors, and thus a "primary" representation, as close to the sensor as we can hope to get. The L*a*b* space is defined by the CIE to be perceptually equidistant across (most of) the color gamut, and is often used as a reference in color work. It is a non-linear transform of the XYZ space. It also performs very well for our purpose, as shown below. The NPP space is of course the one that we derived from neurophysiological measurements in Chapter 5, and is also a non-linear transform of the XYZ space. We use this space to attempt to link the category model to the underlying neurophysiology.

The conversion from Munsell coordinates, in which the stimulus set is defined, to CIE XYZ coordinates, which is the basis for the color spaces we are interested in, is non-trivial, and there is no simple mathematical conversion possible. Fortunately, the Munsell set of standard color reference chips, from which the Berlin and Kay set is chosen, has been measured spectrophotometrically and converted to CIE xyY coordinates in the past [Newhall

[2]The choice of CIE A as a light source is somewhat unfortunate, since its spectral power distribution is considerably skewed, which results in lower resolution measurements for some parts of the spectrum, as well as problems with color constancy, since most color work is done relative to either a C or a flat spectrum light source. We are thus forced to use perceptual color constancy in a somewhat perverse way, assuming that it takes care of any distortions introduced by the light source, but ignoring it for all other purposes.

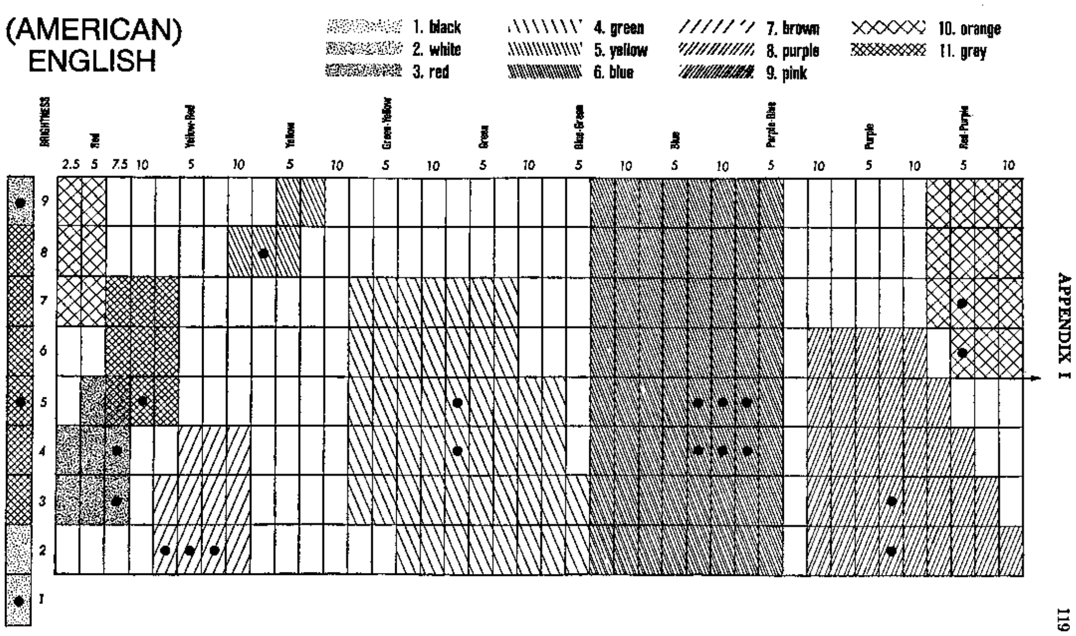


Figure 6.3: The extent and focus for each of the eleven basic color categories of (American) English, from [Berlin & Kay 1969, p.119]. The vertical bar on the left represents the nine values of the grey scale used, the rectangular array represents the set of 320 color chips: 40 equally spaced hues (horizontal) by 8 equally spaced brightness levels (values, vertical). The shaded areas represent the extent of the categories, the dots inside the shaded areas their foci (a category can have a focus extending over more than one chip). The legend above the array contains some errors (which are reproduced here from the original): "10. orange" should read "10. pink", "9. pink" should read "9. orange", and its pattern should be a narrow cross-hatch as in the third shaded area from the left in the color array. Compare to Figures 4.1 and 6.5 for the color version of the stimulus set.

et al. 1943].[3] After conversion from CIE xyY, we obtain unnormalized CIE XYZ coordinates for each of the stimuli contained in the Berlin and Kay set. To normalize the coordinates to the unit cube, with the gray axis going from $\langle 0,0,0\rangle$ to $\langle 1,1,1\rangle$, I used Von Kries adaptation:

$$VK(\overline{x}) = \frac{\overline{x}}{\overline{w}} \tag{6.3}$$

where $\overline{x}$ is the vector representing the unnormalized stimulus values, and $\overline{w}$ is the vector representing the unnormalized white reference stimulus values. Although Berlin and Kay's gray axis only runs from Munsell Value 1 to 9, I used the coordinates of Munsell Value 10 as white reference, since that is the maximum Munsell Value defined, i.e. the "whitest white" available. Although Von Kries adaptation cannot theoretically be shown to exactly undo all the effects of a non-flat spectrum light source, it works well enough in practice to be allowable, especially with a light source that is as close to a flat spectrum as the CIE C source used in these measurements [Wyszecki & Stiles 1982]. The obtained stimulus set is shown in Figure 6.4, represented in CIE XYZ, CIE L*a*b*, and NPP coordinates. Some

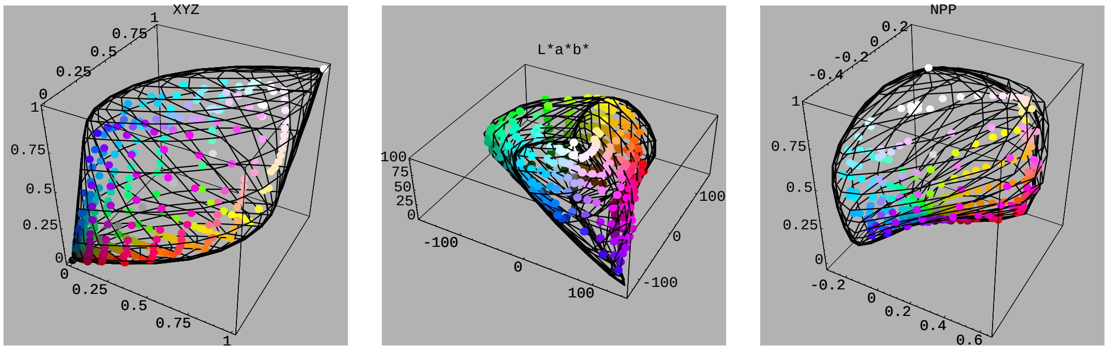


Figure 6.4: The Berlin and Kay stimulus set, represented in the CIE XYZ, CIE L*a*b*, and the NPP color spaces (left to right). Each stimulus is represented as a dot, with an approximate color rendition derived via conversion to RGB and gamma correction, as described in Section 5.4. The stimuli are shown together with a wire-frame rendition of the OCS surface, as described in the same section.

interesting things to note about this figure are:

[3] These measurements have been made relative to magnesium oxide as white reference, and with a CIE C light source – hence the assumption we have to make about color constancy.

- In all color spaces, the stimuli lie near or on the OCS surface, as is expected since they represent the maximum saturation levels available at each brightness and hue value. The OCS surface was computed entirely independently, however, as described in Section 5.4. This provides a nice cross-check for the consistency of both data sets.

- The spacing of the stimuli varies in the various color spaces. Although it is difficult to see from the figure, the spacing is most equal in the CIE L*a*b* space, which was created explicitly for this purpose. The NPP space is less than perfect in this respect. One might speculate that this is due in part to the relatively early stage in the visual neural pathway from which the corresponding measurements are collected, and that perceptually equal spacing is an effect of subsequent transformations of the neural signals.[4]

- There are some irregularities in the spacing of the stimuli, particularly in the blue region. These irregularities had already been noticed by [Newhall et al. 1943], who contribute them to problems with the underlying CIE standard observer functions themselves. They are not serious enough to cast doubt on the validity of Berlin and Kay's results, however.

Some of the same features may be seen in Figure 6.5, which is a recreation of the Berlin and Kay stimulus set based on the XYZ values obtained as described, and converted to RGB for display as before. In particular, note the irregularities in the lower blue region. I have added Munsell Values 0 and 10 to the gray axis, for a total of 11 stimuli.

Combining the information from Figure 6.3 with the derived color space coordinates of the stimuli, we can now describe the boundary of a Basic Color category as a polygon passing through the coordinates of each of the boundary stimuli, and the focus of a Basic Color category as the center of mass of the points indicated as focal points. Figure 6.6 shows the boundaries and foci obtained in this way. Note that the shape and size of the polygons is different in different color spaces, and that in general a straight line on the Berlin and Kay chart does not necessarily translate into a straight line in the color space, since the stimuli are lying on or near a curved surface.

[4] But of course I won't speculate.

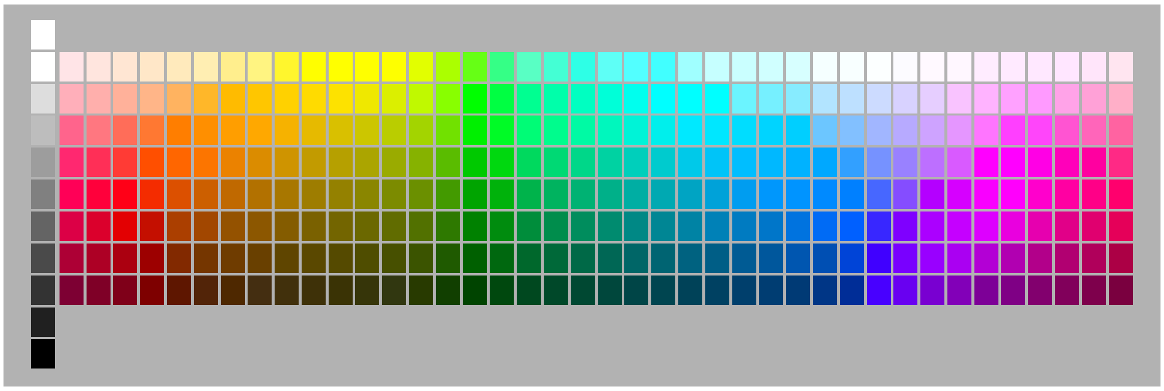


Figure 6.5: Recreation of the Berlin and Kay stimulus chart, based on XYZ values of the stimuli (see text), converted to RGB for display, as described above. Again, the displayed colors are only approximations because of the unknown color gamut of the device that produced the color rendition you are looking at, and the limitations of "typical monitor gamuts" which have been used in the conversion. All colors have been brought into the target gamut by truncating negative values and values exceeding 1 to 0 and 1, respectively (for normalized RGB coordinates). On a typical computer monitor, most of the colors are actually outside of the device gamut. For display purposes only, I have used Von Kries adaptation relative to Munsell Value 9, rather than 10, for no other reason than that seems to result in a better color rendition of the whole set.

6.3 Fitting the Model to the Data

After choosing a category model and quantifying the data set, we now need to fit the model to the data. I will describe a one-pass procedure to accomplish this, but in principle it is also possible to "learn" the fit incrementally, using one example at a time.[5]

The problem is complicated by having a continuous-valued model (the normalized Gaussian) but discrete data (member of the category or not, focal example or not). We need to choose some threshold value to be associated with the category boundaries, and another value to be associated with the foci. Given our model it is natural to choose a value of unity for the focus, and for the threshold value I have arbitrarily chosen $\frac{1}{2}$. After some experimentation, I have obtained the best results with a fitting procedure that minimizes

[5] See below for some remarks on learning versus optimization, and for a sketch of such an algorithm.

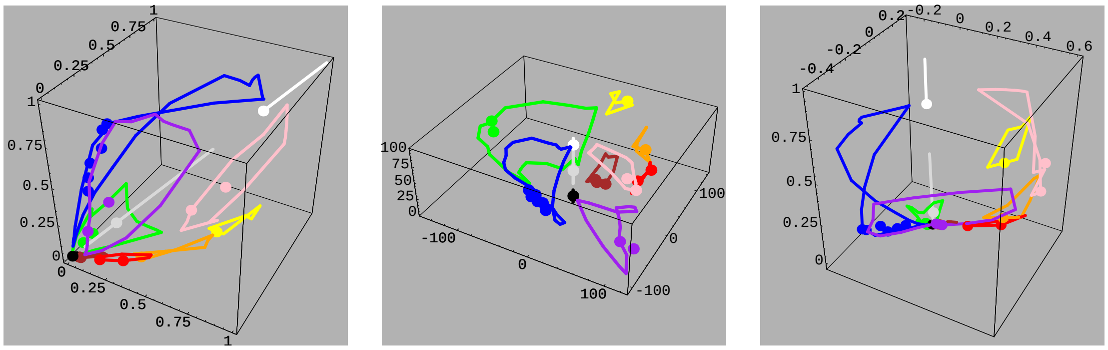


Figure 6.6: The boundaries and foci of the Basic Color categories for (American) English, represented in the CIE XYZ, CIE L*a*b*, and NPP spaces (left to right). Boundaries are represented as polygons and foci as dots, with the color corresponding to the category in question.

the following quantity (independently computed for each category):

$$E = \sum_{i=1}^{N_v} [G_n(\overline{v_i}, \overline{s}\,\overline{\mu}, \sigma) - \theta]^2 + w_f \sum_{i=1}^{N_f} \left[G_n(\overline{f_i}, \overline{s\mu}, \sigma) - 1\right]^2 + \sum_{i=1}^{N_o} [T\left(G_n(\overline{o_i}, \overline{s\mu}, \sigma)\right)]^2 \quad (6.4)$$

which is of course a sum squared error criterion. In equation 6.4, E is the total sum squared error for a particular category, composed of three separate terms. The first term quantifies the error of fit with respect to the vertices of the bounding polygon, with N_v representing the number of vertices, G_n the normalized Gaussian function from equation 6.2, $\overline{v_i}$ a boundary vertex vector[6], $\overline{s}$ a linear scaling vector for the initial focus $\overline{\mu}$ of the normalized Gaussian, σ the "width" parameter of the normalized Gaussian, and θ the threshold value for category membership, in our case equaling $\frac{1}{2}$. The second term quantifies the error of fit with respect to the focal stimuli, with w_f representing a fixed weight for this error term, set to $\frac{1}{3}$ by trial-and-error[7], N_f representing the number of focal stimuli (indicated by black dots in Figure 6.3), $\overline{f_i}$ a focal stimulus vector, and the remaining symbols as in the first term. The third term quantifies the error of fit with respect to other-category representatives, with

[6] The term "vertex" may be somewhat misleading here, as we are concerned with every stimulus along the boundary of the category, regardless of whether it happens to lie on a straight segment of the boundary or not.

[7] I am somewhat bothered by the frequent need for "magic numbers" like this, and the associated arbitrariness. But then to some extent the choice of any mathematical model is a somewhat arbitrary event, of course.

N_o representing the number of other-category representatives, T a threshold function (see below), $\overline{o_i}$ representing an other-category representative[8], and the other symbols as before. The other-category representatives are collected by selecting from the boundary or focus stimuli of each other category those that are closest (in terms of Euclidean distance) to the initial focus of the current category (see below), for each dimension separately. The union of the sets of such stimuli over all dimensions becomes the set of other-category representatives. It is necessary to select the nearest points in each dimension separately, because the dimensions may be scaled independently in a subsequent step. The threshold function T is defined as

$$T(x) = \begin{cases} 0, & x < \eta \\ x, & otherwise \end{cases} \tag{6.5}$$

The effect of using T in the third term is to create a horizon for other-category representatives: as long as their function value for the *current* category's model function is below some value *eta*, it is not counted at all. I have used $\eta = 0.9\,\theta$. The minimization of E is carried out with respect to σ and $\overline{s}$ (each component independently) by Mathematica's standard FindMinimum function, which finds local minima in a function by following the path of steepest descent from any point it reaches. Loosely speaking, this amounts to moving the focus of the model function (the normalized Gaussian) around in the color space so as to bring the boundary stimuli's membership values as close as possible to θ, the focus stimuli's membership values as close as possible to 1, and the membership values for any other category's representatives below the horizon value η.

The use of other-category representatives in the error function may cause the category boundaries of a category to shift to some extent as a function of the distance, or presence or absence, of other categories nearby. There is some evidence in Berlin and Kay's work that the extent of categories may indeed be influenced by the total number of basic categories, and hence by their mutual distance, but I have not examined this any further.

Since the minima found by the FindMinimum function are local, the initial values for the variables σ and $\overline{s}$ are quite important. For the initial value of σ I have used the identity

[8] No pun intended.

scaling vector $\langle 1, 1, 1 \rangle$, which combined with a value of μ given by

$$\mu = List_{i=1}^{N_d} \left[\frac{\sum_{j=1}^{N_f} f_{ji}}{N_f} \right] \tag{6.6}$$

(with N_d representing the number of dimensions of the space, and f_{ji} the i-th coordinate of focal stimulus j) places the initial model focus at the center of gravity of the focal stimuli, and hence on or near the OCS surface. In order to determine an initial value for σ, we find the same-category stimulus s_m (focal or boundary, but typically boundary) that is farthest away from μ as just defined (using Euclidean distance in the color space), and then determine σ such that $G_n(s_m, \overline{s}\,\overline{\mu}, \sigma) = \theta$. Since the value of the normalized Gaussian is a monotonic function of distance to the focus, and s_m is the point in the set farthest away from the focus, any value of σ larger than (i.e. a "wider" curve) that will result in a monotonically increasing square error over the complete set (for a constant focus), and values smaller than that will lead to the nearest local minimum. Even if this minimum is not the global one, this procedure will ensure it is the "same" minimum for all categories. In principle the error landscape as a function of σ can have up to N local minima for N points over which the error is computed. In practice the minimum found in this way seems to be usually the global one, or at least very close to it. Figure 6.7 illustrates this for the 1-dimensional case.

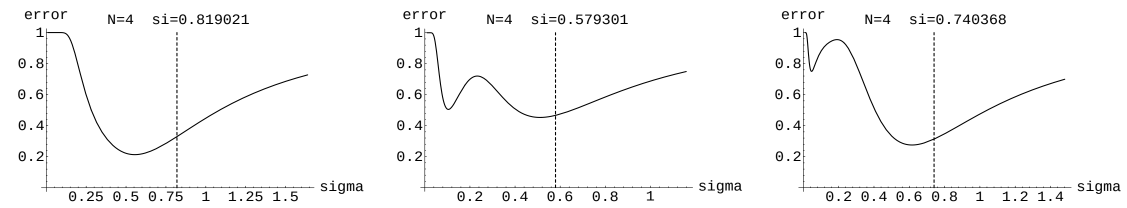


Figure 6.7: The value of the first term of the error E (equation 6.4) as a function of the parameter σ, for fixed values of $\mu = 0$ and $N_v = 4$, with randomly chosen 1-dimensional points in $[0, 1]$. The initial value of sigma as determined by the procedure described in the text is indicated by the dashed line. On the X axis: σ, on the Y axis: error. We can see that the error landscape can have more than one local minimum, but in each case the one closest to the initial value of σ is also the global one, although that is not true in general.

After determining the values of $\overline{s}$ and σ through error minimization, a second minimization step is applied, which involves distorting the Euclidean nature of the space. The first step used a Euclidean distance metric, resulting in spheroid contours of equal membership value. The second step allows the dimensions of the space to be scaled individually for each category model, which can effectively results in an non-spheroid contour of equal membership value (cf. the discussion of Shepard's model in section 6.1).[9] The procedure for this scaling involves an other error minimization comparable to the first, with the objective function defined by

$$E = \sum_{i=1}^{N_v} \left[G_n(\overline{s'}\,\overline{v_i}, \overline{s'}\,\overline{\mu'}, \sigma') - \theta\right]^2 + w_f \sum_{i=1}^{N_f} \left[G_n(\overline{s'}\,\overline{f_i}, \overline{s'}\,\overline{\mu'}, \sigma') - 1\right]^2 + \sum_{i=1}^{N_o} \left[T\left(G_n(\overline{s'}\,\overline{o_i}, \overline{s'}\,\overline{\mu'}, \sigma')\right)\right]^2 \tag{6.7}$$

which is basically the same as equation 6.4, but having both data points and category foci scaled by the vector $\overline{s'}$, and with $\overline{\mu'} = \overline{s}\,\overline{\mu}$ where $\overline{s}$ is the result of the first minimization step. The minimization is carried out as before by Mathematica's FindMinimum function, with initial values $\sigma' = \sigma$ after the first minimization step, and $\overline{s'} = \langle 1, 1, 1\rangle$. Informally speaking, this amounts to stretching (or compressing) the dimensions of each category to make it fit the data better. Alternatively, one can think of it as stretching the dimensions of the color space to make it fit the categories better, but locally only. The obtained scaling vectors for the XYZ, L*a*b*, and NPP spaces are shown in Figure 6.8. One way to interpret this figure is that the closer the vectors cluster to the main diagonal of the unit cube, the more Euclidean the space is (or the categories are), since that corresponds to spheroid surfaces of equal membership value. There are clear differences between the spaces in this respect, but there does not seem to be one "best" space in this respect.

The result of fitting the category models to Berlin and Kay's data for (American) English, using the two-step method described above, is shown in Figures 6.9 to 6.11. These figures show the locations of the category centers (the values of μ' for each category model) only; the boundaries will be discussed below. It is interesting to note that some of the foci lie outside of the OCS volume, which means that they do not correspond to physically re-

[9] In neurophysiological terms one might think of the "receptive field" of a neural (sub)network "implementing" a particular category, and it is not unreasonable to assume that those receptive fields might be somewhat differently shaped from one category to the next, and not necessarily spheroid in terms of the underlying color space coordinates – but all this is highly speculative.

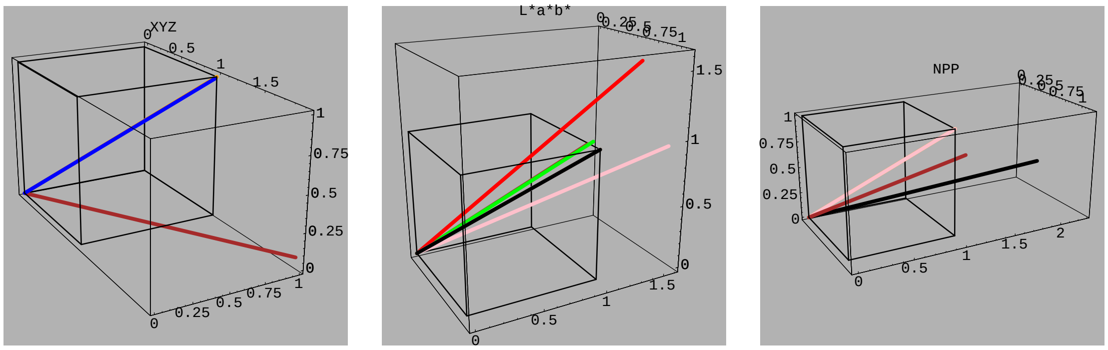


Figure 6.8: The scaling vectors obtained by minimizing equation 6.7 with respect to σ' and $\overline{s'}$, for the XYZ, L*a*b*, and NPP spaces (left to right). For reference, the unit cube is displayed as well. The unit scaling vector, which leaves the space unchanged, is the main diagonal of the unit cube, going from $\langle 0,0,0\rangle$ to $\langle 1,1,1\rangle$. The vectors are colored according to the Basic Color category they represent. Note that only one of them shows in case of multiple overlapping vectors.

alizable (and perceivable?) surface colors. Technically, this is the result of allowing the foci to "float" in the first part of the fitting procedure. While it is possible to constrain the foci to lie on or below the OCS surface, that results in a considerably poorer fit, and degraded performance for the algorithms using the category models, to be discussed in Chapter 7. I am not sure what might the the neurophysiological correlate of these "virtual foci", if any.

6.4 Theoretical Evaluation of the Category Model

To evaluate the category model just described, we will compare the model's categorial judgments to the Berlin and Kay data, i.e. the data to which it is fitted.[10] We will do this for each color sample in the Berlin and Kay stimulus set, which is actually a superset of the data the model was fitted to.[11] Note that our model (normalized Gaussians) implies convex Basic Color category regions, and while that is not always the case for the regions as depicted in Figure 6.3 (e.g. the green region), it is generally true for the regions as mapped onto the OCS surface in the various color spaces, perhaps with some minor local

[10] In Chapter 8 we will evaluate the model on other data sets as well.
[11] I am purposely avoiding terms like "training" and "learning" here.

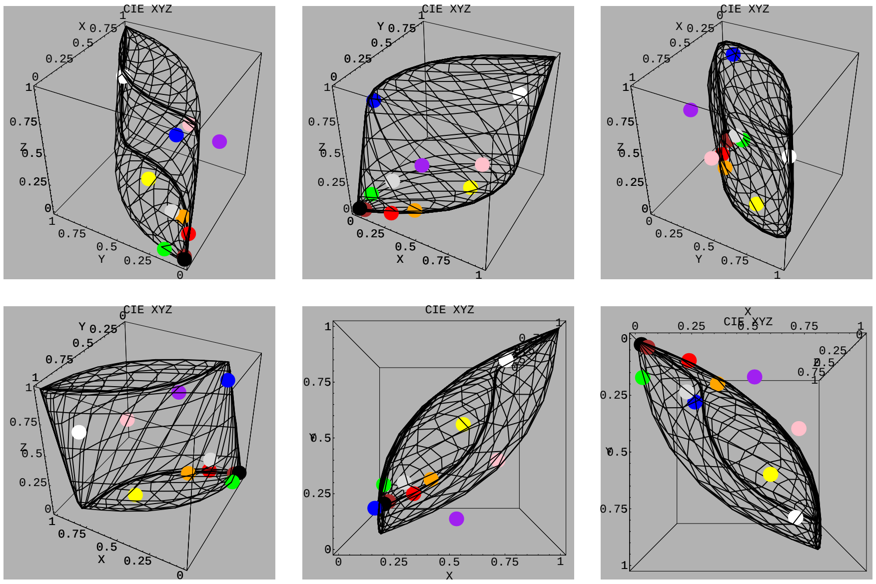


Figure 6.9: The locations of the category model foci (values of μ') in XYZ space is shown by the colored dots, relative to a wire-frame rendition of the OCS surface. In English reading order: four views rotating around the OCS surface, and one from above and below.

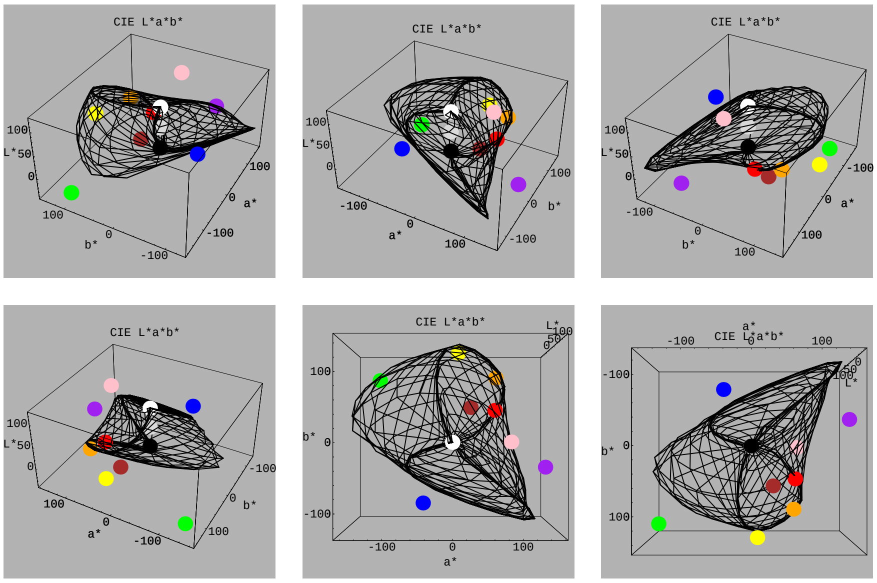


Figure 6.10: The locations of the category model foci (values of μ') in L*a*b* space is shown by the colored dots, relative to a wire-frame rendition of the OCS surface. In English reading order: four views rotating around the OCS surface, and one from above and below.

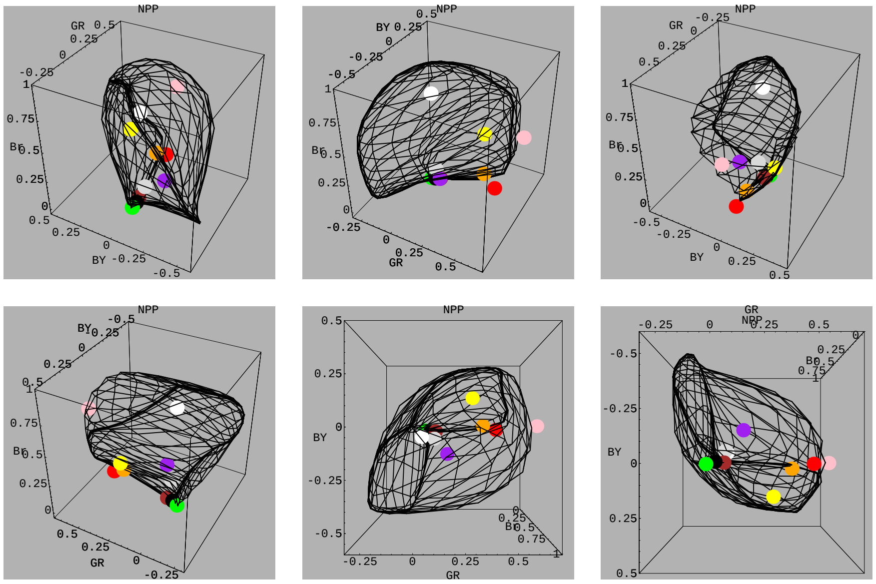


Figure 6.11: The locations of the category model foci (values of μ') in NPP space is shown by the colored dots, relative to a wire-frame rendition of the OCS surface. In English reading order: four views rotating around the OCS surface, and one from above and below. The location of the blue category focus is not shown because it lies too far outside of the figure's bounding box.

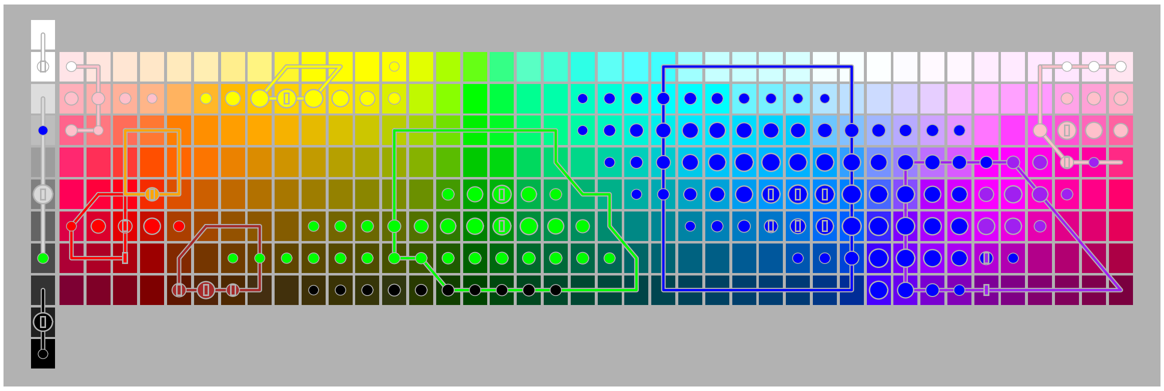


Figure 6.12: Categorization results for the CIE XYZ color space. The category model judgments are shown as disks of the color corresponding to the category function returning the largest membership value for each sample, with a diameter proportional to the membership value. If all category functions return a value below the threshold θ (Section 6.3), no disk is shown. This corresponds to a "none" or null judgment. The results are shown superimposed on a recreated stimulus chart as in Figure 6.5, with Berlin and Kay's category boundaries and focal stimuli indicated by lines and small rectangles, respectively.

exceptions.[12] Figures 6.12 to 6.14 show the results of this comparison.

From visual inspection of these figures it is apparent that the categorization works better in some spaces than in others. In some spaces a certain category may "bleed" outside of the Berlin and Kay category boundaries, e.g. in the XYZ space, green intrudes into the brown region (and these two region's boundaries do not touch in the Berlin and Kay data), and black into green; blue spills over into the purple region, purple somewhat into pink, white into pink, and green and blue slightly into gray. Other categories exceed their boundaries without intruding into another basic category's region, e.g. yellow in all three spaces (but particularly in L*a*b*). Another type of mismatch is a category that does not fill enough of its region (which may coincide with exceeding the boundaries in other areas), e.g. orange, brown, green, blue, and purple in XYZ. The same type of errors occur in all three spaces, but to varying extents. The L*a*b* seems to perform best in general, followed by the NPP space and the XYZ space. Some errors do not seem as serious as others, e.g. the row of white judgments for the very pale blue and pink stimuli in the top row in the NPP space

[12] We should keep in mind that the Berlin and Kay data represents averaged judgments over a number of subjects, which may introduce phenomena that are not necessarily present in any individual subject's data.

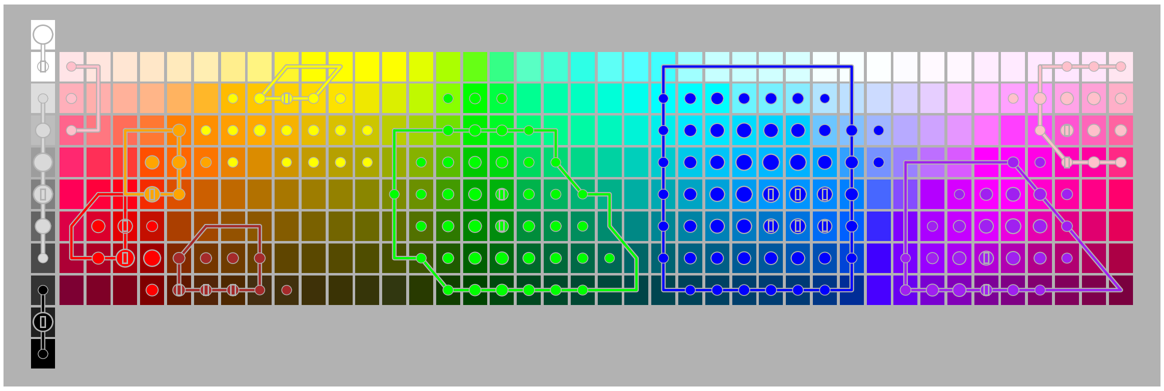


Figure 6.13: Categorization results for the CIE L*a*b* color space. See Figure 6.12 for details.

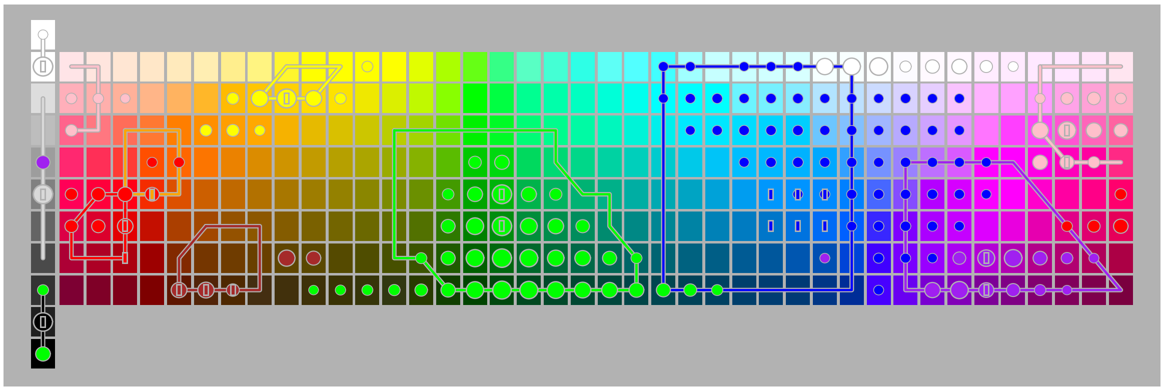


Figure 6.14: Categorization results for the NPP color space. See Figure 6.12 for details.

does not seem like a particularly serious mistake, but judging gray as green or blue in the XYZ space or as purple in the NPP space does. This judgment is qualitative in nature, of course, and somewhat subjective. In addition, performance on color-related tasks may be considered more important than this type of theoretical evaluation (see Chapter 8).

To try and get at a more objective measure of performance, we now turn to quantifying the "goodness of fit" of the category models. There are two concurrent criteria we can use: how well the extent (area or volume, depending on which representation we use) of a model category fits the extent of the corresponding category in the data, and how close the model focus of each category is to the corresponding category focus (or focal samples) in the data. The following error metric attempts to capture the first of these (the extent of the categories), and is computed over the complete set of Berlin and Kay stimuli

$$E_x = \sqrt{\frac{1}{N_s}\sum_{i=1}^{N_s}(\alpha_p^i - \alpha_e^i)^2} \tag{6.8}$$

$$\alpha_p^i = \begin{cases} 0, & \alpha_m^i < \theta \wedge C_m^i \notin P^i \\ \alpha_m^i, & \alpha_m^i < \theta \wedge C_m^i \in P^i \\ 0, & \alpha_m^i \geq \theta \wedge C_m^i \notin P^i \wedge P^i \neq \emptyset \\ \alpha_m^i - \theta, & \alpha_m^i \geq \theta \wedge C_m^i \notin P^i \wedge P^i = \emptyset \\ \theta, & \alpha_m^i \geq \theta \wedge C_m^i \in P^i \end{cases} \tag{6.9}$$

$$\alpha_e^i = \begin{cases} \theta, & P^i \neq \emptyset \\ 0, & otherwise \end{cases} \tag{6.10}$$

$$\alpha_m^i = \max\{\alpha_1^i \ldots \alpha_{N_c}^i\} \tag{6.11}$$

$$C_m^i = C_j \quad such\ that \quad \alpha_j^i = \alpha_m^i \tag{6.12}$$

$$P^i = \{C_i \mid s_i \in C_i\} \tag{6.13}$$

where E is the total error (again a Root Mean Square error metric), N_s is the number of stimuli in the set, α_p^i is the predicted (model) and α_e^i is the expected (data) membership value for stimulus i, α_m^i is the maximum membership value for stimulus i over all category model functions, θ is the membership threshold as before, C_m^i is the category yielding the

α^i_m, P^i is the set of all data categories that the stimulus belongs to (decided by being within the category boundaries as indicated on Figures 6.12 ff.), α^i_j is the membership value for stimulus i in model category C_j, and s_i is stimulus i.[13] The complicated way of determining α_p is necessary to deal with various cases such as the predicted category matching or mismatching the expected category, and with discrete data versus continuous model. In addition to the error over the complete data set, the square error for each stimulus is added to the running total for category C_j iff

$$(C_j \in P^i) \vee (C_j = C^i_m \wedge \alpha^i_m \geq \theta) \tag{6.14}$$

i.e. either when the data or the model says it should belong to category C_j, so we will get an error when the model category is either too small or too large, as compared to the same category in the data.

The error in the placement of the category model foci is determined as follows: for each category C_j, determine the stimulus s_i that is closest to its focus $\overline{\mu_j}$, using the regular Euclidean distance metric on the color space coordinates. This is also the stimulus with the maximum membership value α^i_j, since that is a monotonic function of distance to the focus. Then the center of gravity $\overline{\mu'_j}$ of the data focal stimuli is determined as in equation 6.6, and finally the focal error is computed as the Euclidean distance between the two foci:

$$E_f = \sqrt{\sum_{d=1}^{N_d} (\mu^d_j - \mu'^d_j)^2} \tag{6.15}$$

where E_f is the focal error, N_d is the number of dimensions of the color space (in our case 3), and x^d is the d-th coordinate of point $\overline{x}$. This is approximately equivalent to the distance between the data focus and the orthogonal projection of the model focus onto the OCS surface (there may be small deviations because the stimuli and data focus may lie slightly below the OCS surface). The focal error is computed per category and averaged over all categories.

[13]It would be easier to reproduce a piece of Mathematica code here than to try to write this down in standard mathematical notation, as I have done. Mathematical notation is just not meant to deal with complex data structures and other objects that occur frequently in programs.

The computed errors for the color spaces of interest are shown in Figure 6.15 to 6.17. Comparing these figures, we get confirmation that the L*a*b* space performs best in terms of the extent of the color categories (leftmost bar of the left part of the figures, marked "ALL"). For each of the categories individually except yellow, the error is less in L*a*b* space than in the other two spaces, and for black, gray, and white (i.e. the gray axis), the error is zero in L*a*b* space but not in the other two spaces. Between the RGB and NPP spaces the differences are smaller, with the overall error virtually the same. For some categories such as pink and white the error is smaller in NPP space, for others such as red and yellow the error is smaller in XYZ space. In terms of the error of focus location, the three spaces are comparable overall (leftmost bar marked "AVG" in the figures, with some per-category variation among the spaces.

6.5 Comparing Performance for Different Color Spaces

In the preceding section we have looked at differences in performance of the categorial model with respect to different underlying color spaces. We now take a closer look at why those differences might arise, i.e. if there are any intrinsic properties of the color spaces that might explain the differences in performance.

First of all it is important to note that both the XYZ and the L*a*b* are psychophysical color spaces, defined by the Commission Internationale de l'Éclairage (International Lighting Committee), based on color perception experimental data averaged over large populations. In contrast, the NPP color space is based on neurophysiological recordings made from a single individual (and a monkey at that). As such it should be representative of a particular individual, but not necessarily of a population mean. The color naming data of Berlin and Kay is also averaged over a number of experimental subjects, so it is likely that theirs would be a better fit with the CIE spaces than with the NPP space, when considering only the way both color spaces and naming data were constructed or collected.

The categorial model itself is of course also a psychological model, based on experimental observations over a wide range of tasks and sizable populations [Shepard 1987], so again we would expect similarly constructed color spaces to be at an advantage. The model works best on the L*a*b* space, which is meant to be a perceptually equidistant space,

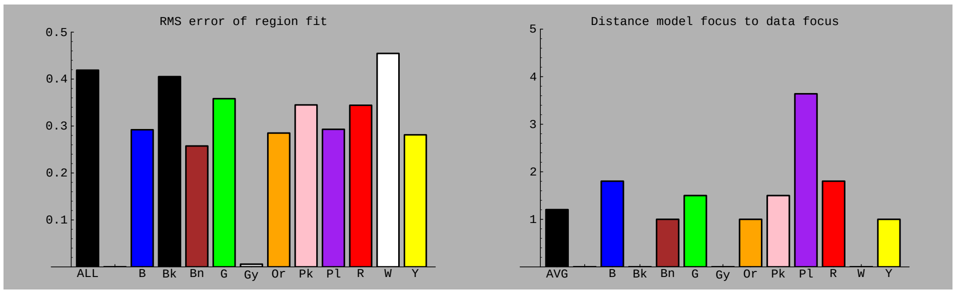


Figure 6.15: The error of fit of the model categories relative to the Berlin and Kay data, for CIE XYZ space. Left: RMS error of fit of the extent of the categories (equation 6.8), right: distance of projected model foci to data foci (equation 6.15). Averages over the set of stimuli/categories is shown in black, per-category error bars are color coded accordingly.

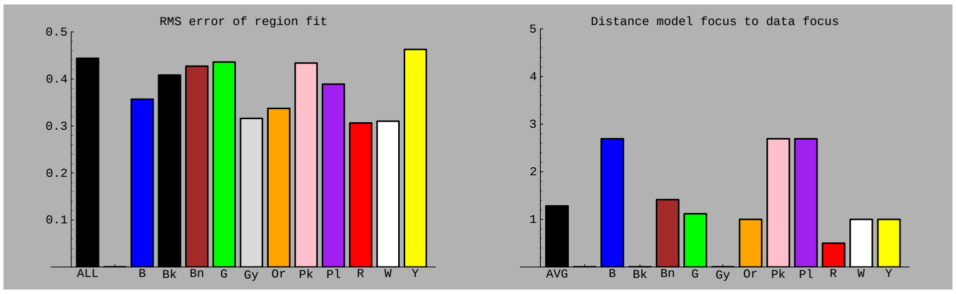


Figure 6.16: The error of fit of the model categories relative to the Berlin and Kay data, for CIE L*a*b* space. Left: RMS error of fit of the extent of the categories (equation 6.8), right: distance of projected model foci to data foci (equation 6.15). Averages over the set of stimuli/categories is shown in black, per-category error bars are color coded accordingly.

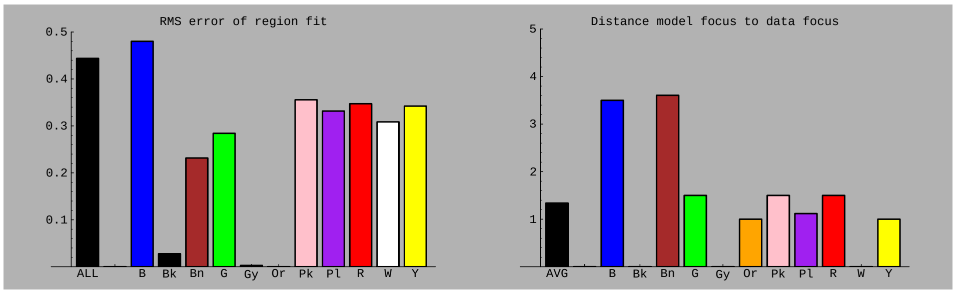


Figure 6.17: The error of fit of the model categories relative to the Berlin and Kay data, for the NPP space. Left: RMS error of fit of the extent of the categories (equation 6.8), right: distance of projected model foci to data foci (equation 6.15). Averages over the set of stimuli/categories is shown in black, per-category error bars are color coded accordingly.

meaning that a fixed distance between two points anywhere in the space should correspond to the same magnitude of perceptual difference (e.g., the perceptual distance between the top two gray scale samples in Figure 6.5 should be the same as the distance between the next two). This is not surprising giving the categorial model, which assumes a Euclidean distance measure on the underlying space, i.e. the perceptual differences should be equal in all directions and in all areas of the space.

The reason the NPP space does not perform better may be related to the scaling issues discussed in Section 5.3.5. For instance, the green region in the NPP space seems to be compressed and low on the brightness dimension, compared to the L*a*b* space (Figure 6.6). This means that the NPP space is not perceptually equidistant. The reason for this could be either that at the LGN stage of color perception, there is no perceptual equidistance yet, and equidistance is an effect of a "higher" stage in the neurophysiology, or it could be that using a different scaling method might result in a more equidistant space. But the latter could itself be seen as modeling a higher stage in neurophysiology, of course. I have not investigated this question any further yet, but it is certainly an interesting one.

Finally, the error metric as developed in the preceding sections does give us a quantitative basis for comparing the performance of different color spaces, but it is a less than perfect one. For instance, it does not take into account how "serious" a categorization error is: categorizing very pale blue as white is not as serious as categorizing green as red. It is difficult to include such qualitative judgments into the error metrics, but doing so might give somewhat different results.

6.6 A Sketch of a Developmental Model

Learning might very well be a domain where angels fear to tread, so my brief remarks on the subject in this section should not be taken as having any relevance for, or being encumbered by any knowledge of, the subject at large.

The fitting of the category model to the color naming data as discussed in Section 6.3 might be considered as an optimization problem, or perhaps a problem of parameter setting in a prior structure, to use the terminology of [Brown 1994], but probably not as a learning approach. Another distinction that [Brown 1994] makes is the one between *experience-*

expectant and *experience-dependent* processes. The former would depend on species-specific experience that might have (had) evolutionary importance, and might be described as "biased learning" or development, while the latter would depend on individual-specific experience, and would be closer to what is generally understood by learning. Biased learning is also related to Edelman's Theory of Neuronal Group Selection [G. Edelman 1992, G. Edelman 1989], in which the concept of *selection* plays an important role, as opposed to *recognition*. In a nutshell, the idea is that categorization (and, by extension, cognition) may be a matter of the selection of certain neuronal groups by certain stimuli, rather than the recognition of a stimulus by a general categorization mechanism.[14] The connection I see between biased learning and selection is that one might consider each of the neuronal groups to "implement" a different bias (or "attractor") for the feature/parameter space over which development and/or learning takes place.

Bringing the discussion back to color categorization, one might consider the foci of the Basic Color categories to represent biases or attractors of this kind. Indeed, Berlin and Kay suggest that the 11 basic color categories represent a universal inventory out of which particular languages choose to lexicalize some number up to 11, presumably dependent on their environment and needs (Section 4.2).[15] I have not been able to relate the location of the foci to any particular neurophysiological phenomena, but let's assume that they are indeed universal, for the purpose of the discussion.[16] Given that we know the locations of the foci in (a particular) color space, a simple developmental or experience-expectant algorithm for determining both the extent and the labels (names) of the categories might go like this:

1. Given: a stimulus, represented as a point $\overline{p}$ in color space
   optional: a label l, represented as a symbol or a string
2. collect the membership values α of $\overline{p}$ for every category in the set $\mathcal{C}$ of known categories: $\Omega = \{\alpha_i^p \mid C_i \in \mathcal{C}\}$
3. select the best candidate category: $C_m = C_i \;\; s.t. \;\; \alpha_i = \max \Omega$

[14] These concepts derive in part from Edelman's Nobel Prize winning work in immunology.

[15] There is some evidence that there may be more than 11 such universal categories, but I will ignore that for now.

[16] The location of the foci may have more to do with the environment in which the color perception mechanism evolved than with any intrinsic features of the mechanism itself, but at this point I can only speculate on this matter.

4. if necessary, adjust the width parameter σ_m of C_m such that $\alpha_m \geq \theta$, without changing the categorization of any other stimulus not belonging to the null category

5. if a label l was provided :

   (a) if no label L_m is associated with C_m, associate l with it: $L_m = l$

   (b) if a label L_m is associated with C_m, and $L_m = l$, do nothing

   (c) if a label L_m is associated with C_m, and $L_m \neq l$, shake.

Step 4 could be implemented by keeping a list of examples of each category on hand, and doing a minimization as described in equation 6.4, with the appropriately chosen other-category representatives, or such a list could be generated as needed each time around by selecting for each other category C_o, the point p that is closest to the focus of the best candidate category C_m, such that $\alpha_o^p = \theta$. This list has to be recomputed every time because the σ_i may change in the course of development. Step 5c is a somewhat complicated case, which could either be due to contradictory input, to a problem with the category model itself, or to a previously overgeneralized category. In any of these circumstances, the system has to be "shaken up" or relaxed into a new maximally consistent state, but I have not considered any algorithms for doing this. The sketched algorithm also needs the σ_i associated with each category to be initialized to some default value, which has an effect on the categorization behavior in its initial stages: choosing the initial σ_i to be small will result in categories that are the smallest possible to contain all examples "seen" to date, while choosing it to be large will result in the largest possible categories that do not conflict with any examples seen to date. These two possibilities may converge to the same state eventually after seeing sufficiently many examples, but I have not investigated that.

Chapter 7

Putting It All Together

From Visual Stimuli to Color Names, and Back

In this chapter I will put all the pieces together, and discuss the complete behaviors that can be modeled using the work described so far. Here we will refer back to the original goals of the research set forth in Section 1.2, viz. to enable an autonomous robotic agent to name colors of objects in its field of view, and to point out examples of objects with specified colors in its environment, both in close agreement with human performance on the same tasks.

As I have explained before, I will not be concerned with issues of color constancy, assuming constant and homogeneous flat-spectrum lighting, nor with issues related to color in context, i.e. the influence context may have on color perception. However, in Chapter 8, where I present an application based on the model, the constancy issue will be dealt with to some extent.

7.1 Naming Color Samples

The complete process required for naming the color of an object (or a blob) in one's field of view is schematized in Figure 7.1. The sensing device, e.g. a color camera, color scanner, or special-purpose color sensor, typically outputs RGB (Red, Green, Blue) values for each

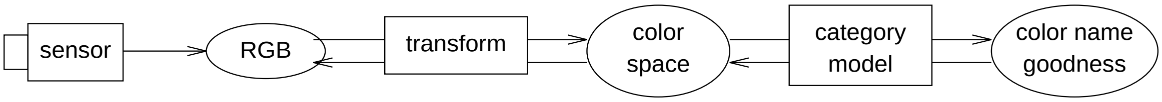


Figure 7.1: Schematic diagram of the color naming process. Boxes represent devices or operations, ellipses represent data representations. Directed line segments represent data transformations. In general, most operations are invertible, at least to some extent.

pixel in its sensor array, which are transformed into the color space coordinates of choice (e.g. XYZ, L*a*b*, or NPP). This transform is usually, but not always, reversible (this is especially easy when dealing with a simple linear transform such as the RGB to XYZ conversion). A blob of a uniform color in the field of view thus corresponds to a point in color space, $\overline{p}$. Once we have determined $\overline{p}$, we determine its membership (or goodness) value with respect to each of the defined color categories $C_i \in \mathcal{C}$:

$$\mathcal{A} = \{\langle C_i, \alpha_i \rangle \mid C_i \in \mathcal{C}\} \tag{7.1}$$

$$\alpha_i = G_n(\overline{p}, \overline{\mu_i}, \sigma_i) \tag{7.2}$$

$$C_i = \langle \overline{\mu_i}, \sigma_i, l_i \rangle \tag{7.3}$$

where $\mathcal{A}$ is the set of all membership values, α_i is the membership value for category C_i, and $\mathcal{C}$ is the set of all defined (non-null) categories.[1] G_n is the normalized Gaussian function from Section 6.1 with the corresponding parameters μ (the focus) and σ, and l_i is the label (name) associated with category C_i. Note that it is always possible to get a non-zero membership value for any category, because the normalized Gaussian is strictly positive everywhere except at infinite distance from the focus. This property is important in forced-choice identification tasks, as we will discuss below.

Next we select all categories for which the membership value exceeds the threshold for

[1]If we think of a category model as a fuzzy characteristic function for the corresponding fuzzy set C, we could write $\alpha = \mu_C(\overline{p})$, but I am avoiding this notation because of the potential confusion with the category parameter μ.

category membership θ, and the corresponding membership values:[2]

$$\mathcal{B} = \{\langle C_i, \alpha_i \rangle \mid \langle C_i, \alpha_i \rangle \in \mathcal{A} \wedge \alpha_i \geq \theta\} \tag{7.4}$$

These are the candidates to provide a name for the color of the stimulus.[3] More than one membership value can potentially exceed the threshold, so there may be more than one element in this set (the set may be empty too, as the combined extent of the categories does not cover the entire color space, at least when thresholded). This is particularly the case for overlapping categories such as red and orange (Figure 6.5 p. 128).

Next we sort the candidates by decreasing membership value:

$$\mathcal{S} = \langle \langle C_1, \alpha_1 \rangle, \ldots, \langle C_n, \alpha_n \rangle \rangle, \quad \alpha_i \geq \alpha_j \Leftrightarrow i \geq j \tag{7.5}$$

and the first element of this n-tuple is our categorization judgment with the corresponding membership or goodness value. The name corresponding to the perceived color is then simply $3^{rd}(C_1)$.

Using a particular threshold value θ amounts to doing a free choice naming experiment with 11 named basic color categories and one null category for everything that does not exceed the threshold for any of the named categories. Using a zero (or no) threshold amounts to a forced-choice experiment where we will always assign one of the 11 named categories to the stimulus, regardless of how low the best membership value might be. This technique has been used by [Hurvich & Jameson 1957] for instance, using only the four primary colors red, green, blue, and yellow, to obtain psychophysical measurements of the color-opponent processes assumed to underlie color perception.

[2]The description of the various steps is mainly for explanatory purposes, and need not reflect an actual implementation strategy. The first two steps can easily be combined into one, for instance.

[3]In practice I apply a "fudge factor" ϕ to θ, so that the condition becomes $\alpha_i \geq \phi\theta$, with $\phi = 0.99$. The reason for this is that the minimization procedure as described in the previous chapter often results in values slightly below θ for the boundary stimuli, because of the RMS error metric used.

7.1.1 Complex color names

Once we have the sorted list of candidates $\mathcal{S}$, we can not only return the name of the best candidate category, but also a complex color name, constructed from the names of the top two candidates. The following algorithm presents a tentative way of doing this. I claim no significance for this algorithm other than being an illustration of how one might go about constructing compound names. I have not compared it with experimental or other existing data on compound color names (if any).

1. Given: $\mathcal{S}$ (equation 7.5), but using $\theta' = \frac{3}{4}\theta$ (equation 7.4)
2. $p = 1^{st}(\mathcal{S})$
   $s = 2^{nd}(\mathcal{S})$
3. the compound name is given by
   (a) $null$, if $\mathcal{S} = \emptyset \vee 2^{nd}(p) < \theta$
   (b) $3^{rd}(p)$, if $s = null$
   (c) $\langle 3^{rd}(s) + \text{"}ish\text{"}, 3^{rd}(p)\rangle$, if $2^{nd}(s) \geq \theta$
   (d) $\langle \text{"}somewhat\text{"}, 3^{rd}(s) + \text{"}ish\text{"}, 3^{rd}(p)\rangle$, otherwise

where 2^{nd} selects the second element of a tuple, and 3^{rd} the third. The lower threshold θ' is used to select both primary and secondary candidates, because the secondary candidate's goodness value is allowed to be lower than the regular threshold for category membership. Step 3a returns a null name (i.e. a no-category judgment) if no candidates exceed the regular threshold θ. Step 3b returns only the primary candidate's name if no secondary candidate is found. Step 3c returns names of the form "greenish blue", where "blue" is the primary name (highest membership value) and "green" the secondary, if both candidates exceed the regular threshold θ. Step 3d, finally, returns names of the form "somewhat greenish blue" if the secondary candidate does not exceed θ (but the primary candidate does). Chapter 8 shows some examples of compound names derived with this algorithm. Varying the parameters (magic numbers) θ and θ' affects the behavior of the algorithm, but I have not explored this in any detail.

7.2 Pointing Out Examples of Colors

The complete process required for pointing out an object (or a blob) of a named color in one's field of view is schematized in Figure 7.2. Compared to the color naming process, we

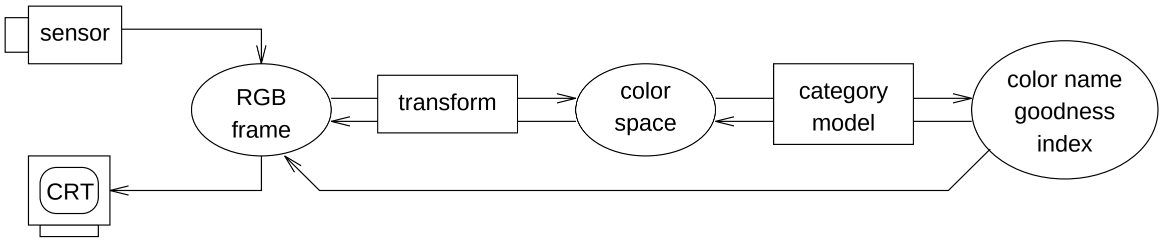


Figure 7.2: Schematic diagram of the color pointing-out process. Boxes represent devices or operations, ellipses represent data representations. Directed line segments represent data transformations. In general, most operations are invertible, at least to some extent. The pointing device suggested here is a monitor (CRT), but could in principle be a robot arm or some other device.

also need a pointing device (a color monitor is suggested in the diagram, but other pointing devices like a robot arm could be used), a complete color image (frame) rather than just a blob, and an index into the image for each blob to be categorized. In Chapter 8, I present an application along these lines, and further implementation detail is provided there.

When given a color name N to point out an example referent of, we sample the image using a certain sample (blob) size, collecting the averaged device RGB values and the image coordinates of (the center of) each blob, and transform the RGB values to the color space of choice, which gives us a set of points in color space with corresponding image coordinates:[4]

$$\mathcal{I} = \{\langle \overline{p_1}, \overline{c_1} \rangle, \ldots, \langle \overline{p_s}, \overline{c_s} \rangle\} \tag{7.6}$$

where $\overline{p_i}$ is a point in color space, $\overline{c_i}$ is the position vector of the corresponding (center of the) blob in the image (field of view), and s is the total number of samples (blobs).

We then determine the membership (goodness) value of the samples in the appropriate

[4] Again, the description is for explanatory purposes, and not meant to suggest an implementation strategy.

category:

$$\begin{aligned}\mathcal{A} &= \{\langle \alpha_j, \overline{c_i}\rangle \mid \langle \overline{p_i}, \overline{c_i}\rangle \in \mathcal{I}\} &(7.7)\\ \alpha_j &= G_n(\overline{p_i}, \mu_j, \sigma_j) &(7.8)\\ C_j &= \langle \mu_j, \sigma_j, N\rangle &(7.9)\end{aligned}$$

where G_n is the normalized Gaussian as before, and C_j is the category with which the name N is associated, with its corresponding parameters μ_j and σ_j.

Next we select all samples with a membership value exceeding the threshold for category membership θ:

$$\mathcal{B} = \{\langle \alpha_j, \overline{c_i}\rangle \mid \langle \alpha_j, \overline{c_i}\rangle \in \mathcal{A} \wedge \alpha_j \geq \theta\} \qquad (7.10)$$

These are all candidate referents for the name N.

As before, we sort the candidates by decreasing membership value:

$$\mathcal{S} = \langle\langle \alpha_1, \overline{c_1}\rangle, \ldots, \langle \alpha_n, \overline{c_n}\rangle\rangle, \quad \alpha_i \geq \alpha_j \Leftrightarrow i \geq j \qquad (7.11)$$

If we just want any referent for the category named N, we pick an arbitrary element $r \in \mathcal{S}$ from this tuple (or from set B), and if we want the best example we take the first element $r = 1^{st}(\mathcal{S})$.

After selecting a referent r, all that is left to do is point it out in the image (or in the world giving rise to the image, which is harder to do), using the index vector $2^{nd}(r)$.

As is the case with color naming, we can do forced choice experiments if we use a zero threshold, in which case a referent will always be selected regardless of how good an example it is, or we can do free choice experiments using the standard threshold θ, in which case a referent will only be selected if it is a "good enough" example of the category in question.

7.3 Choosing Objects by Color

When combined with an image segmentation algorithm, we can use the same techniques to choose objects by color. For instance, if a robot is instructed to "get the red foozle", and it does not have the capability to distinguish foozles from non-foozles,[5] it can still make an educated guess provided it can segment foozles from the background, and there aren't too many red objects around. Color provides another constraint for determining the referents of expressions, and while it may not be sufficient to determine the referent uniquely, it may provide enough constraint to enable a unique determination in combination with other constraints such as shape and size, even if none of these alone would be sufficient. Interestingly, some non-basic color names are so specific that they practically provide all the information necessary to pick out the intended (class of) referent(s). Consider for instance a term like "blond", which is applicable to very few object classes only (mainly hair, possibly beer too).[6] If one has a perceptual category for such a term, one can pick out the intended referent by color only.

Another interesting observation in this respect is that the robot's color perception mechanism need not be as good to perform this task (discrimination) as to perform the naming or pointing tasks. The categories may be considerably "wider" and more overlapping, or the robot's idea of what constitutes a particular color may vary to some extent compared to the agent issuing the request, as long as the colors of potential referents are distributed widely enough throughout the color space. This task is probably best performed with $\theta = 0$, to avoid quibbles over whether or not a particular object color is a good enough example of the requested kind (or in other words, to be maximally cooperative).

7.4 Semantics, Grounding, and Truth

> Meaning derives from embodiment and function, understanding arises when concepts are meaningful in this sense and truth is considered to arise when the understanding of a statement fits one's understanding of a situation closely enough for one's own purposes. Thus, there is no absolute truth or God's-eye view. Our view of what exists (metaphysics) is not independent of how we know it (epistemology). [G. Edelman 1992, p. 250]

We now return to the subject of Chapter 3, discussing some of these issues in the light

[5] This is of course hard to imagine, but let's assume it is the case for the sake of the argument.
[6] Flemish has a term "ros" which means "red", but is only applicable to hair, to my knowledge.

of the model of color perception and color naming we have presented.

I consider the model of color perception and color naming as presented in this dissertation to be a true referential semantic model in the domain of (basic) color names. It is true in the sense that it is well-defined and computable, and enables certain behaviors that require a semantic model to bridge the gap between the external world and internal symbols, and referential in the sense that it expresses the meaning of color terms in terms of a mapping to/from another domain. The referents of the color terms are not directly objects or properties in the world, however, but certain areas in an internally represented color space. But the color space itself is causally connected to the outside world, which I consider a necessary property for any referential semantic model, or at least for the models that constitute the "direct grounding" for some set of terms. Other terms may derive their meaning from being systematically related to directly grounded ones, but without such a core set of directly grounded terms, I believe no meaning or understanding is possible. To put it succinctly: no amount of semantics will ever allow a robot to relate its internal symbols to its environment without some set of directly grounded terms. The algorithm for determining compound color names as presented in Section 7.1.1 may be regarded as a primitive compositional semantic characterization of such terms, which are indirectly grounded in the meanings of (characteristic functions of) the set of basic color terms. I therefore do not agree with an extreme solipsistic view of semantics that would hold that the world is permanently "out of reach" of cognition, and that it therefore does not matter how we choose to characterize the meanings of concepts, as long as this characterization is internally consistent. Meaning is a function of at least two variables: what is "out there", and what is "in here". The world and the cognitive mechanisms both have a necessary role in meaning and understanding. An agent equipped with an arm can actually reach out and touch the things out there that in its internal representation correspond to the referents of some of its internal symbols, and while one may contend that the referent of its symbols is really part of its internal representation and not part of the external world, the difference is not relevant for most practical purposes. One might consider the internal thing to be the "direct referent" and the external thing the "indirect referent", and in most cases they will be causally related.[7] It is possible, however, to uncouple this "alignment" (see Appendix A)

[7]In some cases, one can completely internalize the indirect referent of one's symbols, e.g. by eating it.

under certain circumstances, and in this case we might speak of "being deceived by one's senses" (as in optical illusions) or of "perceptual defects" (as in blindness).

A ubiquitous red herring in writings about semantics is truth. Truth, of the metaphysical God's-eye variety, has very little to do with meaning, in my opinion. The Edelman quote above expresses this sentiment quite well. The often-found Tarskian statement that "snow is white if and only if ⌈snow⌉ ⌈is⌉ ⌈white⌉", where ⌈x⌉ denotes the referent of x (an object or relation over objects) is of no use if those referents are assumed to be in the world, and one is supposed to verify the metaphysical, God's-eye truthfulness of the description of the state of affairs, without the intervention of any perceptual or cognitive mechanism. The most one can say is "I believe snow is white if and only if $G_n(\overline{p_{snow}}, \mu_{white}, \sigma_{white}) \geq \theta$)", where $\overline{p_{snow}}$ is a point in color space corresponding to a sample of an image of something belonging to the perceptual class "snow". Usually, though not always, this will be the case when one has the appropriate white and cold stuff in one's field of view. It is futile to torture oneself with existential doubt as to whether what one is beholding is "really" snow, or merely a very good imitation of it. The duck test applies.

With our concrete model of basic color term semantics before us, it is also easy to see what makes a symbolic representation symbolic. The labels l_i in equations 7.1 are the (names of the) symbols we use for the perceptual categories that they are paired (associated) with. They carry meaning only by virtue of being associated with those categories, but there is nothing intrinsic about them that makes them mean what they mean. The association is arbitrary, and can easily be changed without changing anything fundamental about our understanding of color. A rose would look just as red by any other color name, so to speak. What is *not* arbitrary is the perceptual model, however, since changing any parameters in there will literally make us see the world differently. The perceptual categories might thus be considered analog or iconic in nature, as opposed to symbolic. This also sheds some light on how translation is possible. Without any perceptual underpinnings, the knowledge that "red" = "rood" does not mean much at all. While it may allow us to syntactically substitute one for the other, it would not help us much if we were to move to Flanders. But in the presence of the perceptual category, knowing that "red" = "rood" allows us to move to Flanders and understand what people mean when they talk about "rood" things, and pick out the referents in the world without any further learning. I contend that without a more or less common set of perceptual categories (and hence directly grounded terms), learning a foreign language would not be possible — indeed it would not be possible to understand what anyone else is talking about, and probably it would not be possible to learn any language at all.

Some related issues are brought up in [Davidoff & Concar 1993], where the authors discuss the "memory palette" for colors, or the "internal color space" which functions as a link between the mental worlds of color vision and color language. They report that children have difficulty learning color names: red, green, yellow, and blue are learned first, in no particular order, and only when all four of these are learned can they use any of them correctly. These colors have been called "landmark colors". They are easy to name, form associations with, and children prefer to point to examples of them. They play a central role in learning to name other regions of the color space. Categorization is reported to be not necessarily hardwired, and to some extent alterable by experience. It can be impaired while color vision itself is intact. These findings support the two-stage model of a color space with a separate set of categorization functions defined on that space, and the "looseness" of the coupling between perceptual color categories and color names. Some similar views can be found in [G. Edelman 1989], who mentions research in pre-verbal concepts and complains that concepts and their names are usually tied together in computational models, which is not the case in our model of course. As will have become obvious by now, I share the view of cognition as being based in perception.

With respect to Knowledge Representation and Reasoning issues, the color model can be seen as providing the direct grounding for a set of atomic concepts (base nodes in SNePS terminology, cf. [Shapiro & Rapaport 1987]). These may in turn take part in reasoning about color or about colored objects, but I believe that to be relevant, such reasoning must take the semantics of these terms (the perceptual model) into account, for instance in determining the meaning of compound color names based on the meanings of their constituent terms, as was hinted at in Section 7.1.1. In other words, meaningful reasoning requires meaningful terms. Such meaning must derive directly or indirectly from grounding through perception and action (see Appendix A), and cannot be based on imaginary worlds or descriptive semantics in manuals.

Chapter 8

An Application and Some Empirical Results

This chapter presents a simple application for naming colors, pointing out examples of colors, and selecting objects by color, based on the model presented in the preceding chapters. Some empirical results are presented and discussed, and some ideas are presented for future work along these lines.

8.1 A Simple Application for Color Naming, Pointing Out, and Selecting

8.1.1 Outline

For ease of experimentation, the application consists of two separate parts, one concerned with selection and display of samples from images, and the other concerned with the actual color perception and categorization model (Figure 8.1).

The display program runs under X Windows, and allows one to display a 24-bit RGB image (a TrueColor Visual, in X lingo) acquired from, for instance, a camera and frame grabber, or a color scanner. It also allows one to select samples (blobs) of a certain pre-

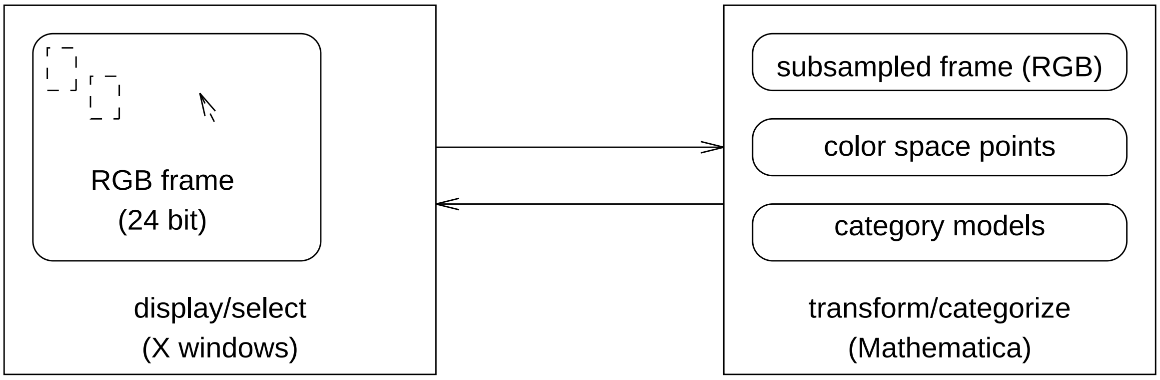


Figure 8.1: Outline of the color naming/pointing/out selecting application, consisting of a display and select part (left), implemented as an X Windows program, and a transformation and categorization part (right), implemented in Mathematica code. The two parts communicate asynchronously via a simple file protocol.

defined size (currently 12×16 pixels) from the image using the mouse, which will be passed to the categorization program. It can subsample the entire image using the same blob size, and pass the result to the categorization program. Finally, it can draw boxes around blobs, whose center coordinates it gets from the categorization program.

The categorization program is a collection of Mathematica functions that can

- *Name* the (average) color of a blob pointed out on the image being displayed by the display program, and provide a goodness value.

- *Point out* examples of a named color category on the image, and provide a goodness value.

- *Select* one from a number of samples whose color best fits a given category, specified by name.

The names returned can be simple or complex, and the best n candidates can be returned. The category membership threshold θ may be specified, as well as the underlying color space to use. The names specified for the pointing-out function can be simple or complex as well, and the threshold θ can be specified. The function can either return (point to) any n examples (the first exceeding the threshold) or the best n examples. The underlying color space to be used can be specified as well. The select function always points to the best example of the specified category (name) within the set of samples provided, using a specifiable underlying color space. It will ask the user to provide it with a set of samples from the image first, to get around the absence of any image segmentation or object recognition algorithms.

8.1.2 Constancy

To make the color naming and pointing algorithms work in practice, with real images, we can no longer avoid the issue of color constancy. The approach I have adopted to deal with this problem is of a rather pragmatic nature, and I make no claims about its generality or applicability to other problems. Nevertheless, the results are interesting, and the approach seems to be fairly robust with respect to different lighting conditions, and to some extent even with respect to different sensing devices. It is presumably related to the "white balance" algorithms implemented in home video cameras and the like, but that is difficult to verify because of the secrecy surrounding commercial applications.[1]

The first thing that needs to be done, if it hasn't been done already, is to gamma correct the image, to make intensities perceptually linear:

$$I' = I^{\frac{1}{\gamma}} \tag{8.1}$$

where I represents the normalized intensity in $[0, 1]$, γ represents the gamma correction factor, typically somewhere in the range $[1, 2.5]$, and I' represents the gamma corrected intensity. Video cameras (especially home camcorders) often incorporate gamma correction, but scanners may not, or may allow it to be specified under software control. This step is required since the CIE XYZ standard assumes perceptually linear intensity responses, and also for display on computer (or TV) monitors. I used a gamma correction factor of 2.2, if necessary.

Next we need to compensate for differences in lighting, both with respect to intensity

[1] For instance, despite repeated attempts and written assurances that the information would not be spread any further, I could not convince the manufacturer of the color scanner I used to acquire some of the test images to provide me with the spectral sensitivity functions of the scanner's color sensors. These sensitivities can be measured given the right equipment, which I did not have access to, however. Sad to say, I have had no more success getting similar information from some academic sources that shall remain equally anonymous.

and with respect to spectral characteristics. Although the spectrum of the light source can never be completely recovered from the image (see Section 2.1), we can use Von Kries adaptation (Section 6.3) with good results, as long as the spectrum of the light source is not too wildly skewed or irregular.[2] In real camera (and scanner) images it turns out that not only is white not white (i.e. the RGB values are not equal and unity, viz. $\langle 1, 1, 1\rangle$), but black is often not black either (i.e. the RGB values are not equal and zero, viz. $\langle 0, 0, 0\rangle$). To compound the problem, lighting may vary considerably across a single image (Figure 8.2).

Figure 8.2: An illustration of the lighting problem. This is a narrow strip taken from an actual camera image, representing a homogeneous white paper background. The image is not very homogeneous, however, and not even all that white, when seen out of context. Paradoxically, taking things out of context is the only way we can fool our visual system into seeing something's "true colors", but this is all our algorithm has at its disposal.

The "constancy" algorithm I have developed searches (a subsampled version of) the image for likely representatives of white and black, and then uses those do to a modified Von Kries adaptation:

$$\overline{p'} = \frac{\overline{p} - \overline{b}}{\overline{w} - \overline{b}} \tag{8.2}$$

where $\overline{p'}$ is the "adapted" pixel value (an RGB vector), $\overline{p}$ is the original pixel value, $\overline{b}$ is the black representative, and $\overline{w}$ is the white representative (both RGB vectors). Finding the black and white representatives works as follows (regarding an image as a set of pixels for convenience):

$$\begin{aligned}
\mathcal{I} &= \{\langle r, g, b\rangle\} && (8.3)\\
\mathcal{S} &= \{\langle\langle r, g, b\rangle, r + g + b\rangle \mid \langle r, g, b\rangle \in \mathcal{I} \wedge \sqrt{(r-g)^2 + (g-b)^2 + (b-r)^2} < \epsilon\} && (8.4)\\
\overline{b} &= 1st(\min_{2nd} \mathcal{S}) && (8.5)\\
\overline{w} &= 1st(\max_{2nd} \mathcal{S}) && (8.6)
\end{aligned}$$

[2] An extreme case would be a single-frequency light source, in which case the effects can never be undone, and no color perception is possible at all.

where $\mathcal{I}$ is the set of image pixels, represented as normalized RGB vectors $\langle r, g, b\rangle$ in $[0, 1]$, $\mathcal{S}$ is the subset of $\mathcal{I}$ containing potential gray axis pixels, and $\overline{b}$ and $\overline{w}$ are the chosen black and white representatives, respectively. The criterion used to select potential gray axis pixels is another RMS measure, enforcing a summed square error between pixel components of less than ϵ. I have obtained good results with $\epsilon = 0.01$. The black and white representatives, then, are the potential gray axis pixels with the lowest and highest brightness, respectively, as determined by a simple sum-of-components measure. A visual way to think of this process is that it stretches the image's actual gray axis and realigns it with the theoretical gray axis for perfectly homogeneous and flat-spectrum lighting, i.e. the line segment $[\langle 0, 0, 0\rangle, \langle 1, 1, 1\rangle]$. The procedure is illustrated in Figure 8.3. One can image that this procedure might fail on

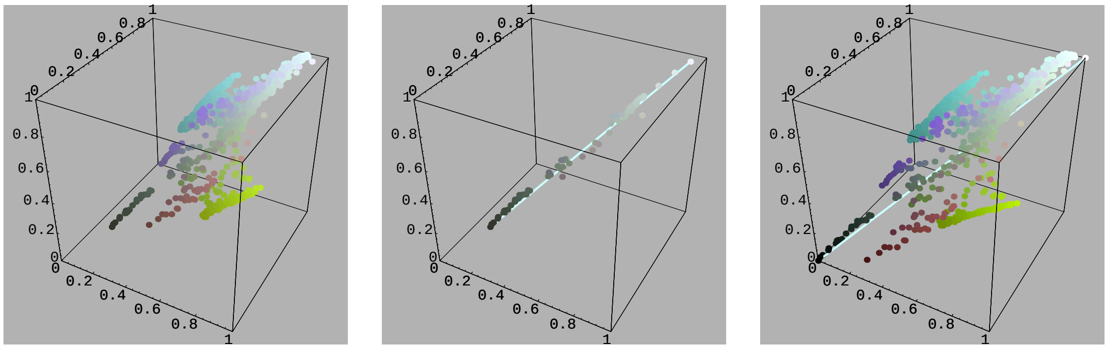


Figure 8.3: An illustration of the adaptation procedure, run on the (subsampled) image from Figure 8.4. Left: the image samples in RGB space. Middle: the set of potential gray axis pixels $\mathcal{S}$, with a line connecting the black and white representatives $\overline{b}$ and $\overline{w}$. Right: the set of adapted image samples, and a line connecting the adapted black and white representatives.

certain images. It is not meant to be foolproof under any circumstances, but to work well enough for our purpose on a large class of "ordinary" images.[3]

The performance of the algorithm is of course influenced by the choice of the parameter ϵ, and perhaps by the method chosen to compute the brightness of pixels (or blobs). I have not investigated this in any detail, but settled on values that seem to work well for the range of images I have used to test the naming functions on.

[3] And of course, people's adaptive mechanism can fail too, under certain specific circumstances. The important thing is that it works under most ordinary circumstances.

8.2 Some Empirical Results

The best way to get a good feel for how well (or how poorly) the algorithm performs is to run it for oneself (See Appendix C for downloading instructions). Barring that, the best we can do is look at an image and the judgments the algorithm makes on it. Again, I must caution against taking the color reproduction on paper too seriously; too many unknowns are involved, but it will do for a rough impression. Figure 8.4 shows a particular test image (which has not been involved in "training" or optimizing the category models at all) taken with a hand-held regular home camcorder whose composite NTSC[4] signal was hooked up to a cheap Sun VideoPix frame grabber. This equipment is certainly not of lab grade quality, not calibrated, and the lighting for the scene was just the available office TL lighting, not controlled or adjusted in any way. The image shows some toy utensils and other objects of various colors, none particularly good examples of Basic (or primary) colors.[5]

A trace of the run is shown below, with only minor editing to remove superfluous white space, and with comments added, preceded by percent signs ("%").

```
% the demo1 function starts the external image display and blob
% selection/display program, and calls the various functions in turn with
% appropriate parameters.

In[2]:= demo1[]
520 x 381 pixels, max value 255
Color Naming

Mouse commands:
left  - sample blob around point
middle  - stop sampling
right  - toggle examples display
Keyboard commands (in the Color Naming window):
S - Subsample and save
Q - Quit program

% The first thing it does is to call the adaptation function, which needs
% to read (a subsampled version of) the image. Since the display program
% and the Mathematica process communicate asynchronously, we see their
% respective outputs interleaved.

% this is Mma
```

[4] Never The Same Color.

[5] These objects are relics of a robotics project that was supposed to include manipulation of the objects with a robot arm as well as color perception (Appendix A). Unfortunately the project met with a premature demise, mainly due to lack of working equipment and funding.

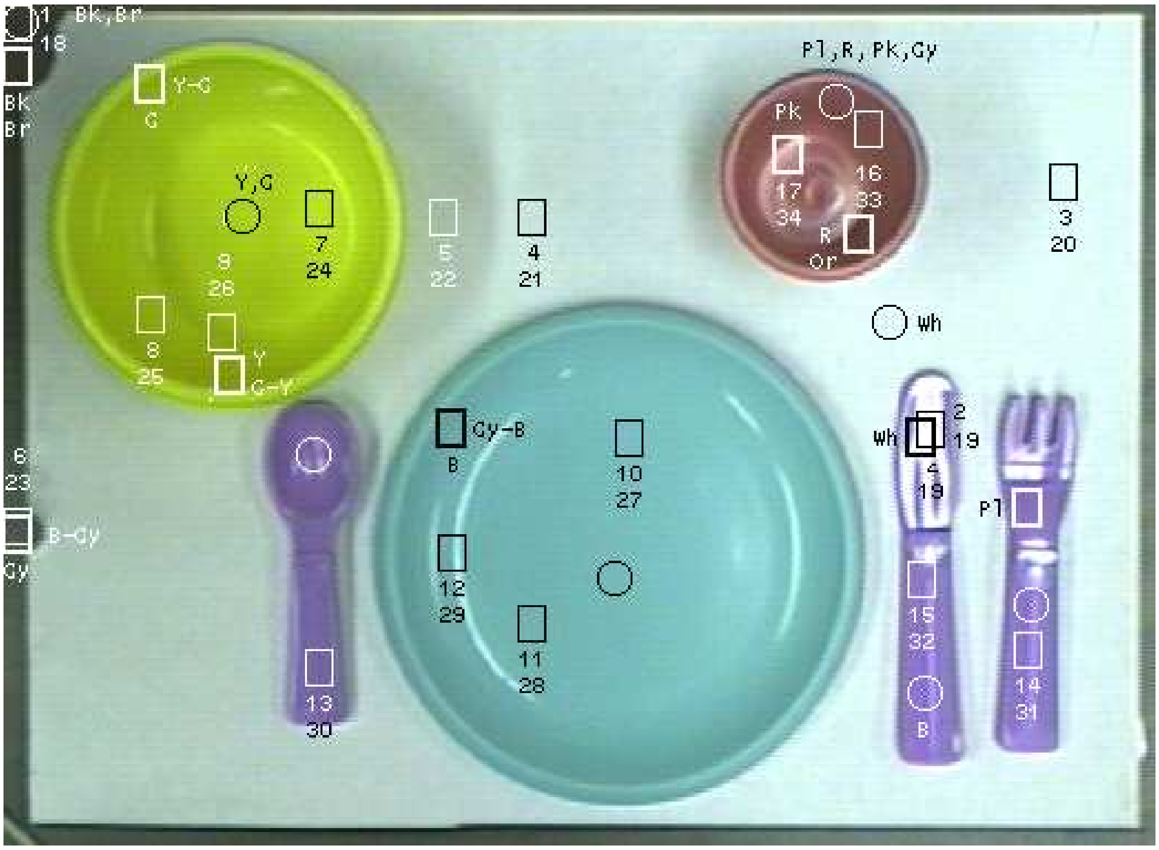


Figure 8.4: A frame grabbed from a regular home camcorder, in ordinary office lighting conditions. Note the shadows and the varying brightness across the even white background, and fuzziness of the objects' outlines. The objects are placed on a regular piece of white paper, on a real simulated woodgrain desk top.

```
Please sample the image
working...

% this is the display prg

Wrote 43 x 23 subsampled file /projects/lammens/pix/subsampled.ppm

% This is the adaptation function reporting the normalized RGB values found
% for black and white, as described above.

{BlackRGB, WhiteRGB} = {{0.196078, 0.203922, 0.164706},

>    {0.937255, 0.945098, 0.988235}}

% Next the color naming function is called, in simple name mode. It uses a
% category membership threshold of 0.35, and the L*a*b* color space. It asks
% for a sample which it gets from the display/choose program, which in turn
```

```
% reports the coordinates of the center of the blob and the average
% unnormalized RGB values. This corresponds to the thin rectangle labeled
% ``1'' in the figure (upper left corner). Note that the RGB values are
% around 50 each, or 0.25 normalized, i.e. not black in absolute terms.

*** Color Naming, simple names ***
sample?
Blob at (8,9): 51 54 46

% After turning the crank a bit, the naming function returns its color
% judgment as a list of {name, goodness} pairs (only one has been
% requested). The reported color is black, because of the adaptation
% performed. It's not all that good a black, however.

{{black, 0.573432}}

% Sample 2 selects the highlight on the blade of the knife, which is a
% specular reflection of the overhead office lights. Not surprisingly, the
% answer is ``white'' -- remember that these algorithms work in a
% context-free manner. It's a pretty good white too.

sample?
Blob at (417,192): 252 253 254
{{white, 0.876566}}

% Sample 3 is from the white paper background, in the upper right quadrant.
% This too is recognized as white, but a less good one than the previous
% sample.

sample?
Blob at (480,79): 223 253 249
{{white, 0.657369}}

% Sample 4 is again from the white paper background, but a slightly more
% shaded area near the middle. Again the answer is white, but a less good
% one still.

sample?
Blob at (240,96): 197 227 220
{{white, 0.55741}}

% Sample 5 is in the shadow of the bowl, and in these case a ``null''
% judgment is returned, meaning the sample does not belong to any basic
% color category. Here our visual system's context-sensitive operation is
% clearly superior (but try looking at the sample through a hole in a piece
% of white paper, and see if it still looks white).

sample?
Blob at (200,95): 164 189 163
{}

% Sample 6 is from the left middle piece of desktop showing underneath the
% white paper, and it is reported as a pretty good gray. The human visual
```

```
% system would connect this sample with the one from the upper left corner
% and not report them as different in color.

sample?
Blob at (7,236): 74 86 77
{{gray, 0.878559}}

% Sample 7 is from the upper right bottom of the bowl, and is
% reported as a not very good green (in fact barely above the threshold for
% category memberhip of 0.35)

sample?
Blob at (140,87): 183 225 33
{{green, 0.38532}}

% Sample 8 is from the shaded lower left inner wall of the bowl, but is
% nevertheless reported as a virtually equally good (or bad) green. The
% algorithm thus show some measure of robustness with respect to varying
% lighting.

sample?
Blob at (62,143): 134 164 11
{{green, 0.385104}}

% Sample 9 is from the lower region of the bowl where the wall and the
% bottom join, and is reported as a moderate yellow. The
% color seems to be in between yellow and green, according to the
% algorithm. And again there are no constraints on neighboring samples to
% be judged the same, so the category may suddenly switch due to small
% local variations.

sample?
Blob at (93,153): 135 162 16
{{yellow, 0.451289}}

% Sample 10 is from the upper middle part of the plate, and reported as a
% moderate blue

sample?
Blob at (280,202): 119 189 194
{{blue, 0.418403}}

% Sample 11 is from the lower left part of the plate, and reported as a
% virtually identical blue.

sample?
Blob at (236,280): 112 179 178
{{blue, 0.413634}}

% Sample 12 is from the left part of the plate, and contains a rather
% bright highlight, but the reported color and goodness are almost the same
% anyway.
```

```
sample?
Blob at (202,247): 127 190 193
{{blue, 0.403172}}

% Sample 13 is from the spoon handle, and reported as a moderate blue too.
% This is not entirely satisfactory, but see below.

sample?
Blob at (142,299): 115 99 162
{{blue, 0.420509}}

% Samples 14 and 15 are from the fork and knife handles, and reported the
% practically the same color as the spoon, notwithstanding the difference
% in lighting.

sample?
Blob at (463,301): 147 125 219
{{blue, 0.423843}}

sample?
Blob at (413,260): 146 127 201
{{blue, 0.412522}}

% Sample 16 is from the upper right of the inner wall of the cup, and
% reported as a moderately good gray. This is again not satisfactory.

sample?
Blob at (393,57): 118 77 73
{{gray, 0.446365}}

% Sample 17 is from a different part of the cup, containing a highlight,
% but it also reported as gray.

sample?
Blob at (347,73): 191 139 140
{{gray, 0.415984}}

% A null sample signals the end of the current naming mode, and the next
% function called is again the naming function, but this time in complex
% name mode. The samples are the same as the first 17 (plus or minus a few
% pixels, because they were entered anew with the mouse).

sample?
*** Color Naming, complex names ***

% Sample 18. The earlier ``black'' judgment is now refined to a somewhat
% greenish-black. The modifier (secondary category) is named first, and the
% goodness values are given in the same order. The algorithm currently does
% not provide a single goodness value for the compound name.

sample?
Blob at (7,9): 51 54 46
{somewhat green-ish black, {0.343353, 0.573432}}
```

```
% Sample 19. The white highlight is now reported as somewhat pinkish white,
% probably due to the colored substrate of the knife (pink is relatively
% close to purple in the color space).

sample?
Blob at (417,193): 252 253 255
{somewhat pink-ish white, {0.324792, 0.884955}}

% Samples 20 and 21. The background becomes blue-ish white, whish is rather
% accurate (compare to the white of the paper surrounding the figure). Note
% that the white goodness goes down, and the blue goodness up a bit as we
% move into the more shaded regions of the white paper.

sample?
Blob at (488,79): 223 253 249
{somewhat blue-ish white, {0.303342, 0.657369}}

sample?
Blob at (241,95): 196 227 219
{somewhat blue-ish white, {0.315056, 0.530577}}

% Sample 22. The heavily shaded region is still not in any basic category.

sample?
Blob at (200,94): 164 189 163
none

% Sample 23. The gray of the desk top is now reported as bluish, which is
% odd.

sample?
Blob at (7,237): 74 86 77
{blue-ish gray, {0.355959, 0.878559}}

% Sample 24. The green of the bowl bottom is still unqualified green,
% albeit not a very good one.

sample?
Blob at (138,85): 181 222 33
{green, 0.386859}

% Sample 25. Elswhere in the bowl, yellow creeps in.

sample?
Blob at (56,140): 135 161 11
{somewhat yellow-ish green, {0.340837, 0.379807}}

% Sample 26. When the top candidate changes, the previous top candidate now
% becomes the modifier, indicating some waffling on the part of the naming
% algorithm.

sample?
```

```
Blob at (89,150): 135 164 16
{green-ish yellow, {0.426267, 0.442337}}

% Samples 27-29. The plate is now reported as grayish blue, which is quite
% acceptable.

sample?
Blob at (275,197): 120 188 192
{somewhat gray-ish blue, {0.263667, 0.414681}}

sample?
Blob at (232,279): 112 178 178
{somewhat gray-ish blue, {0.301262, 0.415194}}

sample?
Blob at (203,247): 127 190 193
{somewhat gray-ish blue, {0.269991, 0.403172}}

% Samples 30-32. The cutlery is now reported as either blue or pinkish
% blue, which is not bad, but purple or purplish blue would be more
% appropriate.

sample?
Blob at (141,301): 116 100 163
{blue, 0.420315}

sample?
Blob at (464,294): 148 125 218
{somewhat pink-ish blue, {0.273953, 0.420837}}

sample?
Blob at (414,259): 151 134 206
{somewhat pink-ish blue, {0.293498, 0.410624}}

% Samples 33-34. The cup color is now modified to pinkish gray, which is
% reasonable given the highlights, but not great.

sample?
Blob at (392,58): 119 77 73
{somewhat pink-ish gray, {0.289888, 0.434842}}

sample?
Blob at (346,73): 186 135 135
{pink-ish gray, {0.372527, 0.440735}}

% After a null sample, the next function called is the pointing-out
% function. It is asked to report the best 3 examples of every color given
% to it by name, but only the top choice is shown in the figure, to reduce
% clutter. The threshold used is now much lower at 0.01, to encourage the
% algorithm to be maximally cooperative and broad-minded (broad-categoried,
% actually). Before it can point anything out, it needs to (sub)sample the
% image, which is effectively its domain of discourse, or its environment,
% for this task.
```

```
sample?
*** Pointing Out Colors, simple or complex names ***
Please sample the image
Wrote 43 x 23 subsampled file /projects/lammens/pix/subsampled.ppm
working...
Enter color names as quoted [list of] string[s], or "" to stop

% First we enter only simple color names. The returned result is a list of
% the top three candidate referents in the image, specified as triplets of
% internal color space coordinates, goodness value, and linearized index.
% The index is converted back to (x,y) image coordinates and passed to the
% pointing/display program, which draws boxes of varying width around the
% corresponding image blobs. The first one of these is shown as a thick
% rectangle in the figure, with the corresponding abbreviated color name.
% The ‘‘>’’ signs in the margin are Mathematica’s continuation sign for
% multi-line outputs.

% The best example of black in the image is the blob in the upper left
% corner, but it’s not a very good one.

Color: "black"
{{{-0.949934, -8.93606, 8.94645}, 0.573432, 0}, {{0., 0., 0.}, 0.499995, 43},

>   {{-8.80682, -0.988521, 16.7996}, 0.434245, 1}}

% The best example of white is the highlight on the knife blade.

Color: "white"
{{{0., 0., 100.}, 0.994671, 550},
>   {{1.19341, -2.5512, 97.4658}, 0.925709, 507},
>   {{-3.30085, 1.8013, 98.0898}, 0.896158, 334}}

% The best example of gray is the blob halfway down the left edge. This is
% probably the only one that does not contain part of the paper edge, other
% than the ones in the upper left corner.

Color: "gray"
{{{-7.9972, 3.37291, 52.6723}, 0.912785, 645},
>   {{-9.15905, 2.19436, 54.5701}, 0.906193, 688},
>   {{3.36029, -6.16337, 62.3559}, 0.898141, 484}}

% The best example of red is on the inner wall of the cup, but its goodness
% value is very low. The redness does not escape the program’s attention,
% so to speak, but the goodness is not high enough to label the object as
% red in the naming mode.

Color: "red"
{{{26.9112, 14.3771, 40.8237}, 0.0651646, 290},
>   {{24.6857, 10.9119, 46.7973}, 0.0347762, 248},
>   {{24.8426, 10.6163, 53.0361}, 0.0235578, 160}}

% As expected, the best exmamples of both green and yellow are from the bowl.
```

```
Color: "green"
{{{-33.8928, 119.015, 78.373}, 0.443789, 91},
>   {{-33.0592, 128.975, 77.1126}, 0.442608, 175},
>   {{-29.7397, 128.662, 75.3204}, 0.441054, 437}}

Color: "yellow"
{{{-30.7869, 148.075, 74.873}, 0.471412, 438},
>   {{-29.7397, 128.662, 75.3204}, 0.4484, 437},
>   {{-32.0185, 149.603, 76.4553}, 0.446844, 394}}

% The best blue is from the plate

Color: "blue"
{{{-20.6264, -5.98904, 77.6899}, 0.428033, 532},
>   {{-21.6781, -5.23694, 76.3669}, 0.426202, 926},
>   {{12.1377, -21.1522, 62.9835}, 0.425152, 829}}

% The best brown is in the upper left corner, from the desk top, but the
% second best canidate (not shown in the figure) is actually on the cup,
% which is good (brown is a kind of dark red, in an artist's conception of
% color).

Color: "brown"
{{{0., 0., 0.}, 0.371792, 43}, {{26.9112, 14.3771, 40.8237}, 0.318304, 290},
>   {{-0.949934, -8.93606, 8.94645}, 0.269128, 0}}

% The best purple is appropriately on the fork. So again, even if a
% different category is chosen in the naming mode, the purpleness is noted
% nevertheless.

Color: "purple"
{{{22.1124, -29.2723, 68.7999}, 0.0935326, 640},
>   {{26.9112, 14.3771, 40.8237}, 0.09335, 290},
>   {{22.4642, -30.4113, 71.309}, 0.0923015, 894}}

% The best pink is appropriately from the cup highlight

Color: "pink"
{{{16.3689, 3.61084, 78.632}, 0.385165, 201},
>   {{17.8138, 5.53527, 70.6242}, 0.354974, 243},
>   {{20.2324, 4.45821, 66.0619}, 0.348803, 202}}

% The best orange is also on the cup, but the goodness is extremely low,
% almost below the already very low threshold. Only one above-threshold
% example is found in this case.

Color: "orange"
{{{26.9112, 14.3771, 40.8237}, 0.0126883, 290}}

% Interestingly, the best example of grayish blue is on the plate, but the
% best example of bluish gray is on the desk top, demonstrating that
% compound category names do not simply represent a conjunction of the two
```

```
% categories.

Color: {"gray", "blue"}
{{{-20.6264, -5.98904, 77.6899}, {0.428033, 0.348169}, 532},
>   {{-21.6781, -5.23694, 76.3669}, {0.426202, 0.356378}, 926},
>   {{18.9681, -27.0463, 63.0806}, {0.424739, 0.281654}, 742}}

Color: {"blue", "gray"}
{{{-7.9972, 3.37291, 52.6723}, {0.912785, 0.337975}, 645},
>   {{-9.15905, 2.19436, 54.5701}, {0.906193, 0.349694}, 688},
>   {{3.36029, -6.16337, 62.3559}, {0.898141, 0.36831}, 484}}

% The best examples of yellowish green and greenish yellow are both on the
% bowl however, again indicating the ambiguity of the color in the
% algorithm's ``mind''.

Color: {"yellow", "green"}
{{{-33.8928, 119.015, 78.373}, {0.443789, 0.358894}, 91},
>   {{-33.0592, 128.975, 77.1126}, {0.442608, 0.409575}, 175},
>   {{-32.3242, 110.273, 78.338}, {0.441002, 0.329341}, 306}}

Color: {"green", "yellow"}
{{{-30.7869, 148.075, 74.873}, {0.471412, 0.429748}, 438},
>   {{-29.7397, 128.662, 75.3204}, {0.4484, 0.441054}, 437},
>   {{-32.0185, 149.603, 76.4553}, {0.446844, 0.425336}, 394}}

% After a null name, the next function activated is the one to select
% objects by (simple) color names. A number of object samples are collected
% first, representing each of the objects in the image as well as the
% paper background and the desk top.

Color: ""
*** Choosing Objects by Color, simple names ***
object sample?
Blob at (109,104): 165 205 25
object sample?
Blob at (137,205): 130 114 176
object sample?
Blob at (276,251): 119 186 189
object sample?
Blob at (417,294): 147 125 214
object sample?
Blob at (465,274): 148 127 213
object sample?
Blob at (376,46): 120 77 75
object sample?
Blob at (389,153): 228 254 251
object sample?
Blob at (9,11): 50 54 45
object sample?
Enter color names as quoted string[s], or "" to stop

% The circles in the figure represent the object samples, and the
```

```
% abbreviated color names next to them represent the choice the algorithm
% makes if given the corresponding color name. The same cooperative
% threshold of 0.01 is used.

% Black and white lead to the expected choices.

Color: "black"
{{{-7.88055, -6.83839, 7.5829}, 0.458093, 7}}

Color: "white"
{{{-6.21646, 2.01139, 101.314}, 0.710922, 6}}


% When given a limited number of ojbects to choose from, gray, red, purple,
% and pink all refer to the same object, viz. the cup.

Color: "gray"
{{{26.7022, 6.30509, 50.5676}, 0.425217, 5}}

Color: "red"
{{{26.7022, 6.30509, 50.5676}, 0.0226092, 5}}

Color: "purple"
{{{26.7022, 6.30509, 50.5676}, 0.0974246, 5}}

Color: "pink"
{{{26.7022, 6.30509, 50.5676}, 0.297329, 5}}

% As expected, yellow and green both refer to the bowl

Color: "green"
{{{-33.6247, 99.8405, 87.0685}, 0.408762, 0}}

Color: "yellow"
{{{-33.6247, 99.8405, 87.0685}, 0.222119, 0}}

% Blue refers to the knife, with the fork and spoon probably close
% runners-up

Color: "blue"
{{{21.2171, -29.1064, 72.5961}, 0.420029, 3}}

% brown refers to the desk top

Color: "brown"
{{{-7.88055, -6.83839, 7.5829}, 0.25868, 7}}

% orange does not refer to anything

Color: "orange"
huh?

% None of the simple color names refers to the plate. Probably that would
```

```
% need a complex color name, but this has not been implemented for the
% selection task.

Color: ""

% To get an idea of the effectiveness of the adaptation algorithm, we now
% run the some of the same tests without it, using the default absolute
% black and white values. For brevity, only the results that differ from
% the previous ones are commented. These results are not shown in the figure.

In[4]:= demo2[]
520 x 381 pixels, max value 255
Color Naming

Mouse commands:
left  - sample blob around point
middle  - stop sampling
right  - toggle examples display
Keyboard commands (in the Color Naming window):
S - Subsample and save
Q - Quit program

*** Without Adaptation ***
{BlackRGB, WhiteRGB} = {{0, 0, 0}, {1, 1, 1}}
*** Color Naming, simple names ***

% Sample 1 (desk) was black before, but is now gray.

sample?
Blob at (9,9): 51 55 46
{{gray, 0.95517}}
sample?
Blob at (417,192): 252 253 254
{{white, 0.997448}}
sample?
Blob at (477,84): 222 252 248
{{white, 0.827816}}
sample?
Blob at (242,93): 195 226 217
{{white, 0.643781}}
sample?
Blob at (200,93): 164 188 163
{}
sample?
Blob at (6,237): 75 86 76
{{gray, 0.890602}}

% Sample 7 (bowl) was green before, and is now nothing.

sample?
Blob at (138,84): 180 222 33
{}
sample?
```

```
Blob at (59,142): 134 163 11
{{green, 0.382396}}

% Sample 9 (bowl) was yellow before, and is now green

sample?
Blob at (92,151): 135 163 16
{{green, 0.375072}}
sample?
Blob at (277,198): 120 189 193
{{blue, 0.396411}}
sample?
Blob at (236,284): 112 179 177
{{blue, 0.391728}}
sample?
Blob at (202,249): 127 189 192
{{blue, 0.387349}}

% Sample 13 (spoon handle) was blue before, and is now gray.

sample?
Blob at (141,304): 116 100 163
{{gray, 0.443478}}
sample?
Blob at (465,296): 148 126 219
{{blue, 0.413124}}
sample?
Blob at (413,258): 150 134 203
{{blue, 0.397054}}
sample?
Blob at (392,59): 119 77 72
{{gray, 0.734896}}
sample?
Blob at (347,71): 186 134 135
{{gray, 0.432342}}
sample?
*** Color Naming, complex names ***

% sample 18 (desk) was somewhat greenish black before, and is now somewhat
% blueish gray

sample?
Blob at (9,9): 51 55 46
{somewhat blue-ish gray, {0.318699, 0.95517}}
sample?
Blob at (417,191): 252 252 254
{somewhat pink-ish white, {0.32611, 0.997364}}
sample?
Blob at (483,77): 223 254 250
{somewhat blue-ish white, {0.319529, 0.821147}}
sample?
Blob at (237,94): 198 228 219
{somewhat blue-ish white, {0.323538, 0.672593}}
```

```
sample?
Blob at (199,94): 164 189 162
none

% Sample 23 (desk) was blueish gray before, and somewhat blueish gray now

sample?
Blob at (8,237): 74 86 77
{somewhat blue-ish gray, {0.34003, 0.892704}}

% Sample 24 (bowl) was green before, none now

sample?
Blob at (136,86): 179 221 32
none

% Sample 25 was somewhat yellow-ish green before, green now

sample?
Blob at (61,140): 136 165 11
{green, 0.381509}

% Sample 26 was green-ish yellow before, green now.

sample?
Blob at (89,148): 135 164 16
{green, 0.375411}
sample?
Blob at (282,199): 120 190 194
{somewhat gray-ish blue, {0.277526, 0.396891}}
sample?
Blob at (233,277): 112 178 178
{somewhat gray-ish blue, {0.323495, 0.394109}}
sample?
Blob at (202,246): 127 190 193
{somewhat gray-ish blue, {0.285553, 0.387858}}

% Sample 30 was blue before, blue-ish gray now

sample?
Blob at (141,302): 116 100 163
{blue-ish gray, {0.406485, 0.443478}}
sample?
Blob at (463,299): 147 125 219
{somewhat pink-ish blue, {0.297212, 0.414387}}

% Sample 32 was somewhat pink-ish blue before, gray-ish blue now

sample?
Blob at (412,260): 142 125 195
{gray-ish blue, {0.350279, 0.400874}}
sample?
Blob at (388,52): 119 77 75
```

```
{somewhat pink-ish gray, {0.291203, 0.739966}}
sample?
Blob at (347,72): 188 137 138
{pink-ish gray, {0.352761, 0.421948}}
sample?
*** Pointing Out Colors, simple or complex names ***
Please sample the image
Wrote 43 x 23 subsampled file /projects/lammens/pix/subsampled.ppm
working...
Enter color names as quoted [list of] string[s], or "" to stop

% The previous best example of black was on the desk, now there is none.

Color: "black"
{}
Color: "white"
{{{0.532836, -2.37186, 97.8842}, 0.947593, 550},
>   {{-2.09951, -0.735345, 96.4016}, 0.914406, 334},
>   {{-2.68196, -2.32448, 97.9623}, 0.904322, 552}}

% The previous best example of gray was half way down the left edge, now it
% is in the upper left.

Color: "gray"
{{{-3.02248, 4.28535, 52.6059}, 0.961785, 0},
>   {{-4.55361, 3.84577, 55.9149}, 0.961349, 44},
>   {{-4.38929, 4.59856, 55.5826}, 0.956423, 86}}

% now no best example of red is found

Color: "red"
{}

% the best examples of yellow and green are still from the bowl, but in
% different places than before

Color: "green"
{{{-23.5407, 62.7201, 80.1716}, 0.381321, 348},
>   {{-23.3872, 61.536, 79.7746}, 0.380848, 392},
>   {{-24.859, 61.5938, 81.5351}, 0.378426, 305}}
Color: "yellow"
{{{-23.5407, 62.7201, 80.1716}, 0.110165, 348},
>   {{-23.3872, 61.536, 79.7746}, 0.106706, 392},
>   {{-25.7179, 63.496, 83.541}, 0.102021, 398}}

% The best blue was on the knife handle before, now on the plate

Color: "blue"
{{{14.5833, -22.6702, 77.6263}, 0.413639, 851},
>   {{14.9457, -23.0869, 78.1905}, 0.413597, 812},
>   {{15.2259, -22.919, 77.2831}, 0.41233, 894}}
Color: "brown"
{{{-3.09103, 5.85521, 51.7689}, 0.147524, 43},
```

```
>   {{-3.02248, 4.28535, 52.6059}, 0.138768, 0},
>   {{9.80109, 9.4391, 60.507}, 0.138071, 290}}
Color: "purple"
{{{15.2259, -22.919, 77.2831}, 0.0661223, 894},
>   {{14.5268, -21.4402, 75.6475}, 0.0654806, 640},
>   {{14.9457, -23.0869, 78.1905}, 0.0647991, 812}}
Color: "pink"
{{{11.0969, 2.14573, 82.0917}, 0.360105, 201},
>   {{11.043, 4.11308, 76.6872}, 0.336285, 243},
>   {{11.9717, 3.68921, 73.7798}, 0.330684, 202}}

% the best orange was on the cup before, now none is found

Color: "orange"
{}
Color: ""
```

Some tests on rather different images, for instance color cartoons acquired with a scanner, show very comparable results. Although the application presented in this chapter is an interesting demonstration of what can be accomplished with the color perception and naming model, it is certainly open for improvement. For instance, it would be interesting to explore an active version of the adaptation algorithm that could move the camera around searching for appropriate lighting conditions.

Chapter 9

Discussion and Conclusion

The goal for the research presented in this dissertation, as set forth in Chapter 1, was twofold: firstly to contribute to theories of autonomous agency, in particular to the study of *symbol grounding* or *embodiment*, and secondly to do this by modeling a particular aspect of perception and natural language semantics, viz. the domain of color perception and color naming. From a methodological point of view, the approach chosen was to study these phenomena from a "vertically integrated" perspective, resulting in an experimental implementation of a complete (albeit narrow-minded) color-naming and color-pointing agent, ranging all the way from real visual stimuli captured by a camera to symbolic descriptions using natural language terms, and back.

With respect to color perception, the goal was to model (an aspect of) human color perception by modeling known neurophysiological data on the responses of a certain class of color-sensitive neurons found in the human (and primate) visual system, relating these responses to existing color perception models of a psychological or psychophysical nature. The attempted explanation is of course only partial, as it does not deal with phenomena such as color constancy and the influence of context on color perception. In this respect the research presented was partly successful, in that it showed how one can derive a psychophysical color space (the NPP space) from neurophysiological data, and the resulting color space shows some remarkable similarities to existing psychological and psychophysical color spaces that have been derived independently, and using entirely different means. Various features of the derived color space, and the method used to derive it, are worthy of further investigation in and of themselves. The success in this area was only partial because the derived color space does not perform better for the purpose of color naming and color pointing than other psychophysical color spaces, in particular the CIE L*a*b* space. I believe this is mainly due to the fact that the NPP space is not fully perceptually equidistant, something which the L*a*b* space is explicitly constructed to be. In addition, the NPP space is based on measured responses from a single Macaque monkey only, and the CIE spaces are based on average experimental data from large populations of human subjects.

The category model that was developed, representing a perceptual color category as a normalized Gaussian function in three-dimensional color space (described by a focus location and a parameter determining the "width" of the function) performs well for our purpose. The model itself is not an arbitrary construct, but is based on independent psychological research in categorization, and presumably has a wider applicability to categorization in general. Since it assumes an underlying perceptually equidistant (Euclidean) space, it is not surprising that it works best in conjunction with the CIE L*a*b* space. The category model may also offer a tool to study bias in learning or development of visual (and perceptual) categories, but this connection is very tentative. The learning dimension of the color naming problem has only been touched on very briefly, and there certainly remains more work to be done in this area.

The color perception and naming model as implemented is of course limited. For instance, it does not deal with dynamically changing visual input, and for the most part assumes a constant adaptation state of the visual system. The computational approach adopted to deal to some extent with color constancy seems adequate for our purpose, but it is not clear whether it has any wider applicability. The more general question of how a color perception system comes about, with precisely this set of basic color categories, is not really addressed by the work presented. There are probably evolutionary reasons for why our color perception system works the way it does, but that is outside of the scope of the current research.

With respect to the larger goal of making a contribution to theories of cognitive agent architecture, I believe the research is successful. It shows how a particular set of terms (either terms from a natural language or terms from a knowledge representation and reasoning system) can be *grounded* or *embodied* in a perceptual categorization model, and presents a working implementation of such a model. Of course the scope of the model is limited, dealing only with Basic Color Terms, but it nevertheless represents a step in the direction of a general theory of (artificial) agency. The autonomous agent architecture my colleagues and I have developed as a tangent to the work presented in this dissertation hopefully contributes a little to this larger goal. One of the most important features of the color model in this respect is that it has a large bottom-up component (everything leading up to the color space representation of visual stimuli), causally connected to the outside world (the agent's environment), in addition to a top-down component (the categorization and naming model). These two components interact in a well-defined way, and while we can describe and study them separately to some extent, neither is eventually of any use without the other. I believe that the knowledge representation and reasoning community would do better to work on the problem of grounding some fundamental set of terms in this way than to worry about unicorns, round squares, possible worlds, and modeling logical inference (at least for the foreseeable future). The latter subjects may be interesting as an aid in modeling and understanding some of *our own* thought processes, but it is of no use to an artificial cognitive agent as long as it cannot relate to its environment in even the most basic of ways. In this respect I subscribe to the methodology of what has come to be known as "nouvelle AI" in some circles, emphasizing complete (albeit simple) artificial organisms that can function in a real world environment, before moving on to more "high level" problems. I believe AI can no longer afford to live in an ivory symbolic tower, and needs to deal with the "low level" grunge of the real world, and how an agent can relate to and interact with it.

Appendix A

An Architecture for Autonomous Agents

My work in color perception and color naming can be seen as an instance of *symbol grounding* [Harnad 1990], which is a methodology for providing artificial agents with an "understanding" of the world around them, and an inherent meaningfulness for the symbols they use. This appendix describes joint work with Henry H. Hexmoor and Stuart C. Shapiro in architectures for intelligent autonomous agents, which is to appear as [Lammens et al. 1994]. The text below is reproduced verbatim from the text submitted for the proceedings of the "NATO Advanced Study Institute on the Biology and Technology of Intelligent Autonomous Agents", held in Trento, Italy, March 1-12, 1993, which is referred to as "the current workshop" below.

A.1 Introduction and Overview

In the elephant paper [Brooks 1990] appearing in the proceedings of the predecessor of the current workshop, Brooks criticizes the ungroundedness of traditional symbolic AI systems, and proposes physically grounded systems as an alternative, particularly the subsumption architecture. Subsumption has been highly successful in generating a variety of interesting and seemingly intelligent behaviors in a variety of mobile robots. As such it has established itself as an influential approach to generating complex physical behavior in autonomous agents. In the current paper we explore the possibilities for integrating the old with the new, in an autonomous agent architecture that ranges from physical behavior generation inspired by subsumption to classical knowledge representation and reasoning, and a new proposed level in between the two. Although we are still struggling with many of the issues involved, we believe we can contribute to a solution for some of the problems for both classical systems and physically grounded systems mentioned in [Brooks 1990], in particular:

- The ungroundedness of symbolic systems (referred to as "the symbol grounding problem" by [Harnad 1990]): our architecture attempts to ground high level symbols in perception and action, through a process of embodiment.
- The potential mismatch between symbolic representations and the agent's sensors and actuators: the embodied semantics of our symbols makes sure that this match exists.
- Our symbolic representations do not have to be named entities. The knowledge representation and reasoning system we use in our implementations allows the use of unnamed intensional concepts.
- We have some ideas about how to automate the construction of behavior generating modules through learning, but much remains to be done.

We agree with the requirement of physically implemented systems as the true test for any autonomous agent architecture, and to this end we are working on several different implementations. We will present both our general multi-level architecture for intelligent autonomous agents with integrated sensory and motor capabilities, GLAIR[1], and a physical implementation and two simulation studies of GLAIR-agents.

By an *architecture* we mean an organization of components of a system, what is integral to the system, and how the various components interact.[2] Which components go into an architecture for an autonomous agent has traditionally depended to a large extent on whether we are *building a physical system*, *understanding/modeling behaviors of an anthropomorphic agent*, or *integrating a select number of behaviors*. The organization of an

[1] Grounded Layered Architecture with Integrated Reasoning

[2] Our discussion of architecture in this paper extends beyond any particular physical or software implementation.

architecture may also be influenced by whether or not one adopts the *modularity* assumption of Fodor [Fodor 1983], or a *connectionist* point of view, e.g. [McClelland et al. 1986], or an *anti-modularity* assumption as in Brooks's subsumption architecture [Brooks 1985]. The *modularity* assumption supports (among other things) a division of the mind into a *central system*, i.e., cognitive processes such as learning, planning, and reasoning, and a *peripheral system*, i.e., sensory and motor processing [Chapman 1990]. Our architecture is characterized by a three-level organization into a Knowledge level (KL), a Perceptuo-Motor level (PML), and a Sensory-Actuator level (SAL). This organization is neither modular, anti-modular, hierarchical, anti-hierarchical, nor connectionist in the conventional sense. It integrates a traditional symbol system with a physically grounded system, i.e., a *behavior-based* architecture. The most important difference with a behavior-based architecture like Brooks's subsumption is the presence of three distinct levels with different representations and implementation mechanisms for each, particularly the presence of an explicit Knowledge level. Representation, reasoning (including planning), perception, and generation of behavior are distributed through all three levels. Our architecture is best described using a resolution pyramid metaphor as used in computer vision work [Ballard & Brown 1982], rather than a central vs. peripheral metaphor.

Architectures for building physical systems, e.g., robotic architectures [Albus et al. 1981], tend to address the relationship between a physical entity, (e.g., a robot), sensors, effectors, and tasks to be accomplished. Since these physical systems are performance centered, they often lack general knowledge representation and reasoning techniques. These architectures tend to be primarily concerned with the *body*, that is, how to get the physical system to exhibit intelligent behavior through its physical activity. We say these systems are not concerned with *consciousness*. These architectures address what John Pollock calls *Quick and Inflexible* (Q&I) processes [Pollock 1989]. We define consciousness for a robotic agent operationally as being aware of one's environment, as evidenced by (1) having some internal states or representations that are causally connected to the environment through perception, (2) being able to reason explicitly about the environment, and (3) being able to communicate with an external agent about the environment.[3]

[3] A machine like a vending machine or an industrial robot has responses, but it is *unconscious*. See [Culbertson 1963] for a discussion of independence of consciousness from having a response. Also, intelligent behavior is independent of consciousness in our opinion.

Architectures for understanding/modeling behaviors of an anthropomorphic agent, e.g., cognitive architectures [Anderson 1983, Pollock 1989, Langley et al. 1991], tend to address the relationships that exist among the structure of memory, reasoning abilities, intelligent behavior, and mental states and experiences. These architectures often do not take the *body* into account. Instead they primarily focus on the *mind* and *consciousness*. Our architecture ranges from general knowledge representation and reasoning to body-dependent physical behavior, and the other way around.

We are interested in autonomous agents that are embedded[4] in a dynamic environment. Such an agent needs to continually interact with and react to its environment and exhibit intelligent behavior through its physical activity. To be successful, the agent needs to reason about events and actions in the abstract as well as in concrete terms. This means combining situated activity with acts based on reasoning about goal-accomplishment, i.e., deliberative acting or planning. In the latter part of this paper, we will present a family of agents based on our architecture. These agents are designed with a robot in mind, but their structure is also akin to anthropomorphic agents. Figure A.1 schematically presents our architecture.

There are several features that contribute to the robustness of our architecture. We highlight them below (an in-depth discussion follows later):

- We differentiate conscious reasoning from unconscious Perceptuo-Motor and Sensori-Actuator processing.[5]
- The levels of our architecture are semi-autonomous and processed in parallel.[6]
- Conscious reasoning takes place through explicit knowledge representation and reasoning. Unconscious behavior makes use of several different mechanisms.

[4] "Embedded agents are computer systems that sense and act on their environment, monitoring complex dynamic conditions and affecting the environment in goal-oriented ways." ([Kaelbling & Rosenschein 1990] page 1).

[5] We consider body-related processes to be *unconscious*, but that is not meant to imply anything about their complexity or importance to the architecture as a whole. Indeed, we believe that the unconscious levels of our architecture (the Perceptuo-Motor level and the Sensori-Actuator level) are at least as important to the architecture as the conscious one (the Knowledge level). We reserve the term *sub-conscious* for implicit cognitive processes such as category subsumption in KRR systems. See [Shapiro 1990] for a discussion of sub-conscious reasoning.

[6] This autonomy is similar to Brooks's subsumption architecture [Brooks 1985], but at a more macroscopic level. Brooks does not distinguish between the three levels we describe, as his work is solely concerned with behaviors whose controlling mechanism we would situate at the Perceptuo-Motor level.

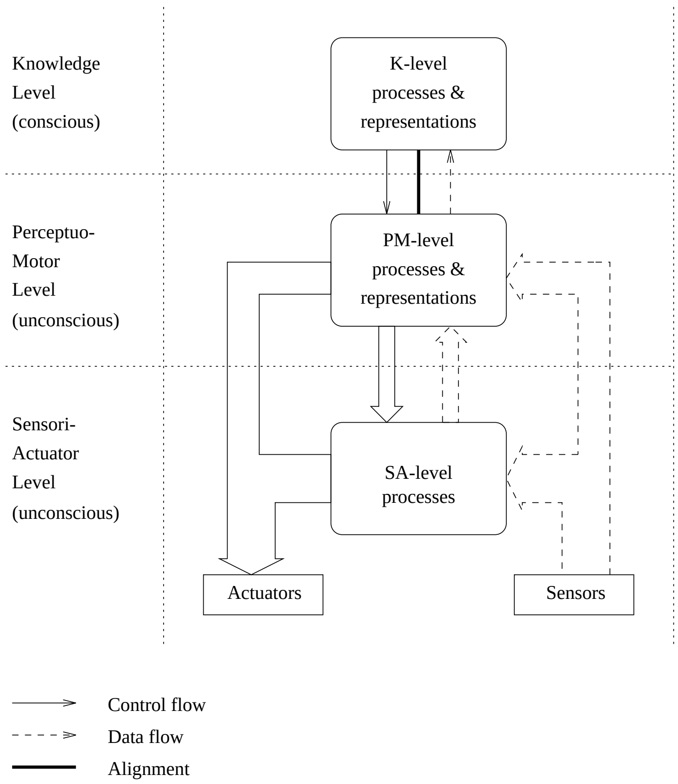


Figure A.1: Schematic representation of the agent architecture. Width of control and data paths suggests the amount of information passing through (bandwidth). Sensors include both world-sensors and proprio-sensors.

- Conscious reasoning guides the unconscious behavior, and the unconscious levels, which are constantly engaged in perceptual and motor processing, can *alarm* the conscious level of important events, taking control if necessary. Control and generation of behavior are layered and not exclusively top-down.

- Lower level mechanisms can pre-empt higher level ones. This is kind of subsumption on its head, but everything depends on the placement of behaviors in the hierarchy of course. We haven't quite decided yet whether inhibition should work the other way around as well.

- There is a correspondence between terms in the Knowledge Representation and Reasoning (KRR) system on one hand, and sensory perceived objects, properties, events, and states of affairs in the world and motor capabilities on the other hand. We call this correspondence *alignment*.

- Our architecture may be appropriate both for modeling elephants and for modeling chess-playing agents.

A.2 The GLAIR Architecture

In this section we discuss in detail our autonomous agent architecture for integrating perception and acting with grounded, embodied, symbolic reasoning.

A.2.1 Related work

Architectures proposed in the literature do not fall into neatly separable classes, mainly because the scope of the models and the motivations vary widely. However, we can divide a review of related work into theoretical issues of agent architectures, on the one hand, and implemented architectures, on the other.

A.2.1.1 Theoretical Issues

We believe that behavior-based AI has adopted the right treatment of every day behavior for agents that function in the world. However, this has been done at the expense of ignoring cognitive processing such as planning and reasoning. Clearly, what is needed is an approach that allows for both. We believe that our architecture meets this need. As in behavior-based AI, GLAIR gains validity from its being grounded in its interaction with the environment, while it benefits from a knowledge level that, independent of reacting to a changing environment, performs reasoning and planning.

The Model Human Processor (MHP) is a cognitive model [Card et al. 1983] that suggests the three components of *perception, cognition*, and *motor*. Cognition consists of working memory, long-term memory, and the cognitive processor. Perception is a hierarchy of sensory processing. Motor executes the actions in the working memory. This is a traditional symbol-system decomposition of human information processing. This type of decomposition has shown only limited success in building physical systems. Despite this, systems like SOAR adhere to this model. In our architecture, we purposively avoid this kind of top-down problem decomposition by allowing independent control mechanisms at different levels to take control of the agent's behavior, and pre-empt higher level control while doing so. It may be necessary to allow higher level mechanisms to selectively inhibit lower-level ones as well, but we have found no good reason to do so yet.

A situated agent, at any moment, attends to only a handful of entities and relationships in its immediate surroundings. In this type of setting, the agent often does not care to uniquely identify objects. It is sufficient to know the current relationship of the relevant objects to the agent, and what roles the objects play in the agent's activities. Agre and Chapman in [Agre & Chapman 1987] proposed indexical-functional representations (which [Agre 1988] refers to as deictic representations) to be the more natural way agents refer to objects in common everyday environments. They called entities and relationships of interest *entities* and *aspects*, respectively. With respect to its current activities, the agent needs only to focus on representing those entities and relationships. Although the objects in the environment come and go, the representations of entities and relationships remains the same. For example, the-cup-that-I-am-holding[7] is an indexical-functional notation that abstracts the essentials of what the agent needs to know in its interaction. These representations serve to limit the scope of focus on entities. For example, if the agent wants to pick up a cup, it does not need to know *who owns the cup* or *how much coffee the cup can hold*; only the relevant attributes of the cup apply. We believe that systems endowed with general KRR abilities can and should generate deictic representations to create and maintain a focus on entities in the world, but we have not yet designed an implementation strategy.

A.2.1.2 Implemented Architectures

Brooks's *subsumption* architecture, [Brooks 1985, Brooks 1987, Brooks 1990], clusters behaviors into layers. Low-level behaviors, like deciding the direction of motion and speed, can be inhibited (subsumed) by behaviors determined at higher levels, such as avoiding obstacles. Subsumption behaviors are written as finite state machines augmented with timing elements. A compiler is used to simulate the operation of finite state machines and parallelism. This architecture is implemented on a variety of mobile robots. Frequently used behaviors are placed at a lower level than less frequently-used behaviors. This organization of behaviors gives the system fast response time and high reactivity. Our architecture is sim-

[7]This kind of designation is merely a mnemonic representation intended to suggest the entity and aspect under consideration, for the purpose of our exposition. It is not the actual representation that would be used by an agent.

ilar to Brooks's in our intra-level implementations of behaviors. However, the subsumption architecture lacks the separation we have made into conscious and non-conscious spheres. In anthropomorphic terms, Brooks's agents are all non-conscious. We believe that the off-line specification and compilation of behavior modules is too inflexible for autonomous agents that can adapt to a wide range of circumstances, especially if they have to learn from their interactions with the environment. Pattie Maes has experimented with a version of a behavior-based architecture, which she calls ANA [Maes 1991]. This architecture consists of competence modules for action and a belief set in a network relating modules through links denoting successors, predecessors, and conflicts. Competence modules have activation levels. Activations are propagated and the competence module with the highest activation level is given control. Maes has explored learning and has applied her architecture to robotic systems.

In the subsumption architecture, sensations and actions are abstracted by giving them names like "straightening behavior" in order to make things easier to understand for human observers. Much in the spirit of [Agre 1988], we believe that behavior modules should more naturally emerge from the interaction of the agent with its environment. In contrast to hand coding behaviors and in order to facilitate embodiment, in GLAIR we are experimenting with (unnamed) emergent behavior modules that are learned by a robot from scratch. An (unnamed) behavior module can be thought of as a set of tuples (P,A) where P is a set of grounded sensations and A is an instance of an act. For instance, reaching for an object might be a set of tuples (vision/sonar data, wheel motor actuation). After learning, this new behavior module will become active only if the grounded sensations match any of the grounded sensations experienced before. As a measure of abstraction and generalization, we may allow near matches for sensations. To bootstrap the learning process, we need a set of primary or first-order ("innate", for the philosophically inclined) sensations and actions. We will return to this point briefly in section A.3.3.1.

The Servo, Subsumption, Symbolic (SSS) architecture [Connell 1992] is a hybrid architecture for mobile robots that integrates the three independent layers of servo control, Brooksian behavior based modules, and a symbolic layer. Our architecture is similar to this in its general spirit of identification and integration of three distinct levels corresponding to levels of affinity-of-interaction (i.e., the rate at which it is in real-time contact with the world) with the outside world. This similarity also constitutes a point of departure, however, in that SSS is defined with respect to specific (and different) implementation techniques. For example, the symbolic layer in SSS seems to be a decision table versus a general KRR as intended in GLAIR. Unlike GLAIR, SSS assigns particular tasks for each layer and uses a hard-wired interconnection channel among layers.

Albus et al's hierarchical control architecture [Albus et al. 1981] is an example of a robotic architecture; we would say it is *body centered.* This architecture proposes abstraction levels for behavior generation, sensory processing, and world modeling. By descending down the hierarchy, tasks are decomposed into robot-motion primitives. This differs from our architecture, which is not strictly top-down controlled. Concurrently, at each level of the hierarchy, feedback processing modules extract the information needed for control decisions at that level from the sensory data stream and from the lower level control modules. Extracted environmental information is compared with the expected internal states to find differences. The differences are used for planning at higher levels.

Payton in [Payton 1986] introduced an architecture for controlling an autonomous land vehicle. This architecture has four levels: mission planning, map-based planning, local planning, and reflexive planning. All levels operate in parallel. Higher levels are charged with tasks requiring high assimilation and low immediacy. The lower levels operate on tasks requiring high immediacy and low assimilation. Our architecture is similar in this respect. The reflexive planning is designed to consist of pairs of the form $\langle virtualsensor, reflexivebehavior\rangle$. Each reflexive behavior has an associated priority, and a central blackboard style manager arbitrates among the reflex behaviors. Some of the problems with the earlier implementation due to using the blackboard model were solved in [Rosenblatt & Payton 1989].

Rosenschein and Kaelbling's work [Kaelbling 1988, Kaelbling & Rosenschein 1990] describes tools (REX, GAPP, RULER) that, given task descriptions of the world, construct reactive control mechanisms termed *situated automata.* Their architecture consists of perception and action components. The robot's sensory input and its feedback are inputs to the perception component. The action component computes actions that suit the perceptual situation. We should note that unlike Brooks's behavior modules, situated automata use internal states, so their decisions are not Markovian (i.e., they are not ahistoric). They are mainly intended to produce circuits that operate in real-time, and some properties of their operation are provable. The mechanism for generating situated automata, although impressive, seems to inflexible for autonomous agents that have to operate in a wide variety of (possibly unknown) circumstances . Perhaps the operation of our Perceptuo-motor level could be modeled by a situated automaton, but we are not convinced that this is the right formalism to use, due to its inflexibility.

Gat in [Gat 1991] describes ATLANTIS, an architecture for the control of mobile robots. This architecture has three components: control, sequencing, and deliberation. The control layer is designed as a set of circuit-like functions using Gat's language for circuits, ALPHA. The sequencing is a variation of Jim Firby's RAP system [Firby 1987]. The deliberation layer is the least described layer. As with situated automata, we are not convinced that this is the right kind of formalism to use, for the same reasons.

An architecture for low-level and high-level reactivity is suggested in [Hexmoor 1989]. High-level reactivity is reactivity at the conceptual level. This architecture suggests that an autonomous agent maintains several different types of goals. High-level reactivity is charged with noticing impacts of events and actions in the environment on the agent's goals. Subsequently, high-level reactivity needs to guide the agent's low-level reactivity. Low-level reactivity is at the sensory, perceptual, and motor level. The mechanism for low-level reactivity is similar to other reactive systems that have components for perception and action arbitration. The novelty of this architecture is the incorporation of high-level reactivity and a supervisory level of planning and reasoning, which guides the choice of low-level reactive behaviors. In our present conception of agent architecture, we avoid a sharp separation between the two types of reactivity. We also relax the top-down nature of interaction between levels. Reactivity may be initiated at any level of our architecture either due to interaction with other levels or in direct response to external stimuli.

SOAR [Laird et al. 1987] was designed to be a general problem solving architecture. SOAR integrates a type of learning known as *chunking* in its production system. Recently, SOAR has been applied to robotic tasks [Laird et al. 1991]. In this framework, planning and acting is uniformly represented and controlled in SOAR. This approach lacks the ability of our architecture for generating behavior at non-conscious levels as well as the conscious level (or at different levels in general), and for having different-level behaviors interact in an asynchronous fashion. It also lacks our multi-level representations.

Simmons's Task Control Architecture (TCA) [Simmons 1990] interleaves planning and acting by adding *delay-planning constraints* to postpone refinement of planning until execution. For example, a plan for a robot to collect used cups for trash is decomposed into: navigate to the cup; pick it up; navigate to trash bin; deposit the cup. Since the robot does not have sensory information about the cup yet, the plan to pick it up is delayed until the robot gets close enough. Selectively delaying refinement of plans allows for reactivity. This type of "stepwise refinement" follows effortlessly from our architecture, without the need to explicitly implement it. Since conscious planning which goes on at the Knowledge level uses a more coarse-grained world model, there is simply no possibility to express fine details of planning and execution. These can only be represented and/or computed at the lower Perceptuo-Motor level and Sensori-Actuator level. Planning and execution in our architecture may proceed in a lock-step fashion, but they need not be. TCA uses a message-passing scheme among modules that allows concurrent execution of tasks. It has been used to control the six-legged walking robot Ambler and a cup-collecting robot.

A.2.2 Architecture levels

We now proceed to discuss one of the distinguishing characteristics of GLAIR: its three levels.

A.2.2.1 Motivation

The three levels of our architecture are of organizational as well as theoretical importance. Organizationally, the layered architecture allows us to work on individual levels in a relatively independent manner, although all levels are constrained by the nature of their interactions with the adjoining level(s). The architecture is hierarchical, in that level i can only communicate with levels $i-1$ and $i+1$, if any.

The levels of our architecture are semi-independent. While control flows mainly top-down and data mainly bottom-up, local control mechanisms at any level can preempt higher-level control, and these local mechanisms filter the data stream for their own purpose, in parallel with higher-level ones. Representations become coarser-grained from bottom to top, while control data becomes more fine-grained from top to bottom. The terms in the Knowledge Level's KRR system model conscious awareness of the world (and the body), and the perception and motor capabilities in the other levels provide the grounding for an *embodied semantics* of the former. Routine, reflex-like activities are controlled by close coupling of perception with motor actions at the (unconscious) Perceptuo-Motor and Sensori-Actuator levels. This close coupling avoids having to exert control over these activities from the conscious level, as in purely top-down structured architectures with a symbol level at the top of the hierarchy. In the latter kind of system, signals must first be transformed to symbols and vice versa. The low-level coupling provides for better real-time performance capabilities, and relieves the Knowledge level of unnecessary work.

In general, we have multi-level layered representations of objects, properties, events, states of affairs, and motor capabilities, and the various levels are *aligned*. By alignment we mean a correspondence between representations of an entity at different levels. This organization contributes to the robustness and computational efficiency of implementations. The semi-autonomous nature of the levels allows for graceful degradation of system performance in case of component failure or situation-dependent incapacitatedness. Lower levels can function to some extent without higher-level control, and higher levels can function to some extent without lower-level input.[8]

Our architecture allows us to elegantly model a wide range of behaviors: from mindless, spontaneous, reflex-like, and automatic behavior, e.g., "stop if you hit an obstacle", to plan-following, rational, incremental, and monitored behavior, e.g., "Get in the car now, if you want to go to LA on Friday".[9]

In anthropomorphic terms, we identify the Knowledge level with consciously accessible data and processing; the Perceptuo-Motor level with "hardwired", not consciously accessible processing and data involved with motor control and perceptual processing; and the Sensori-Actuator level with the lowest-level muscular and sensor control, also not consciously accessible. The substrate of grounding and embodiment [Harnad 1990, Lakoff 1987, Suchman 1988] of actions, concepts, and reasoning is mainly the Perceptuo-Motor level and to some extent the Sensori-Actuator level.

We will now explore representation and computation at the individual levels in more detail.

[8] For instance, in the context of autonomous vehicles, if obstacle avoidance or returning to the base is a lower-level behavior than planning exploration strategies, then a failure of the hardware implementing the latter does not necessarily prevent the former.

[9] The plan is to get in the car to go to the travel agency to get a ticket to fly to LA on Friday. Today is Thursday and it is near the end of the business day. Also, the agency won't accept telephone reservations. This example is suggested in [Pollock 1992].

A.2.2.2 The Knowledge Level

The *Knowledge level* contains a traditional KRR and/or planning system, using a relatively course-grained representation of objects, events (including actions), and states of affairs. For instance, objects are represented at this level as unique identifiers, typically without further detail about their physical characteristics or precise locations. It is possible to represent such detail explicitly at this level, but not required. Only if the detail becomes important to will it be represented, though not necessarily in the same way as at a lower level. For example, knowledge about the physical size and weight of an object might become available at the Knowledge Level through the agent's actively using measuring devices like a ruler or a scale, but this knowledge is not the same as the embodied knowledge about dimensions and weight represented at the Perceptuo-Motor level for the particular object or its object class. As a rule of thumb, representations at this level are limited to objects, events, and states of affairs that the agent needs to be consciously aware of in order to reason and plan, and in order to communicate with other agents at the grain size of natural language. The Knowledge level can be implemented using different KRR and/or planning systems.

Traditional use of the concept of world modeling refers to building models of interactions between the agent and its environment at the conscious level. These models maintain internal states for the agent. The difference in our use of the term "world model" is that we do not intend to have a precise model of all objects in the environment. Instead, we want to model only the entities relevant to the agent's interaction with its world. This requires filtering out some details accessible at the Perceptuo-motor level as the entities are aligned with their counterparts on the Knowledge level. This is known as "perceptual reduction". Physical details of interaction with entities are handled at the Perceptuo-motor level. Representations at the Knowledge level are needed only for explicit reasoning about entities, and contain only the information necessary for doing so. That might include details about physical characteristics in some cases, but it need not. In other cases, it may be limited to a nondescript intensional representation of an object.[10] Conversely, some entities may be represented at the Knowledge level but not at the Perceptuo-Motor level (abstract concepts, for instance). Knowledge level representations are needed for reasoning about entities; Perceptuo-Motor level representations are needed for physically interacting

[10]See [Shapiro & Rapaport 1987] for our use of "intensional representation".

with entities.

A.2.2.3 The Perceptuo-Motor Level

The *Perceptuo-Motor level* uses a more fine-grained representation of events, objects, and states of affairs. For instance, they specify such things as size, weight, and location of objects on the kinematic side, and shape, texture, color, distance, pitch, loudness, smell, taste, weight, and tactile features on the perceptual side. At this level, enough detail must be provided to enable the precise control of actuators, and sensors or motor memory must be able to provide some or all of this detail for particular objects and situations. The Perceptuo-Motor level is partly *aligned* with the Knowledge level, in that there is a correspondence between object identifiers at the Knowledge level and objects at the Perceptuo-motor level.

Kinematic and perceptual representations of particular objects or typical object class instances may be unified or separate, and both kinds of representations may be incomplete. Also at this level are elementary categorial representations; the kinds of representations that function as the grounding for elementary grounded symbols at the Knowledge level, i.e., sensory-invariant representations constructed from sensory data by the perceptual processor [Harnad 1990].

The representations at this level are *embodied* (cf. [Lakoff 1987]), meaning that they depend on the body of the agent, its particular dimensions and characteristics. Robots will therefore have different representations at this level than people would, and different robots will have different representations as well. These representations are agent-centered and agent-specific. For instance, they would not be in terms of grams and meters, but in terms of how much torque to apply to an object to lift it,[11] or what percentage of the maximum to open the hand to grasp an object. Weights of things in this kind of representation are relative to the agent's lifting capacity, which is effectively the maximum weight representable. An agent may have a conscious (Knowledge level) understanding and representation of weights far exceeding its own lifting capacity, but that is irrelevant to the

[11]Of course this also depends on how far the object is removed from the body, or how far the arm is stretched out, but that can be taken into account (also in body-specific terms). People's Perceptuo-Motor level idea of how heavy something is is most likely not in terms of grams, either (in fact, a conscious estimate in grams can be far off), but in terms of how much effort to apply to something to lift it. That estimate can be off, too, which results in either throwing the object up in the air or not being able to lift it at the first attempt, something we have all experienced. On the other hand, having a wrong conscious estimate of the weight of an object in grams does not necessarily influence one's manipulation of the object.

Perceptuo-Motor level. When it comes to lifting it, a thousand-pound object is as heavy as a ten-thousand-pound one, if the capacity is only a hundred or so. Similarly, sizes are relative to the agent's own size. Manipulating small things is not the same as manipulating large things, even if they are just scaled versions of each other. A consequence of using embodied representations is that using different "body parts" (actuators or sensors) requires different representations to be programmed or (preferably) learned. While that may be a drawback at first, once the representations are learned they make for faster processing and reactive potential. Representations are direct; there is no need to convert from an object-centered model to agent-centered specifications. This makes the computations at this level more like table lookup than like traditional kinematics computations, which can be quite involved. Learning new representations for new objects is also much simpler; it is almost as easy as trying to grasp or manipulate an object, and merely recording one's efforts in one's own terms. The same holds, mutatis mutandis, for perceptual representations.

There are a number of behaviors that originate at this level: some are performed in service of other levels (particularly deliberative behaviors), some are performed in service of other behaviors at this level, a few are ongoing, and some others yet are in direct response to external stimuli. An agent may consciously decide to perform Perceptuo-motor actions such as looking, as in *look for all red objects*, or to perform a motor action, such as *grasp a cup*. These actions originate at the Knowledge level and are propagated to this level for realization. An agent has to perform special perceptual tasks to serve other behaviors, such as to *find the grasp point of a cup* in order to *grasp a cup*. These perceptual tasks may originate at this or another level.

At the Perceptuo-Motor level, an agent has a close coupling between its behaviors, i.e., responses, and stimuli, i.e., significant world states. We observe that, for a typical agent, there are a finite (manageably small) number of primitive ("innate") behaviors available. As the agent interacts with its environment, it may learn sophisticated ways of combining its behaviors and add these to its repertoire of primitive behaviors. We will consider only an agent's primitive abilities for now. We further assume that the agent starts out with a finite number of ways of connecting world states to behaviors, i.e., reflex/reactive rules. Following these observations, we suggest that at this level, the agent's behavior-generating mechanism is much like a finite state automaton. As we noted earlier, learning will change this automaton. The agent starts with an automaton with limited acuity, and uses its conscious level to deal with world states not recognizable at the Perceptuo-Motor level. For instance, the Perceptuo-Motor level of a person beginning to learn how to drive, is not sophisticated enough to respond to driving conditions automatically. As the agent becomes a better driver, the conscious level is freed to attend to other things while driving. This is called *automaticity* in psychology. We discuss an implementation mechanism for these automated behaviors later in this paper.

A.2.2.4 The Sensori-Actuator Level

The *Sensori-Actuator level* is the level of primitive motor and sensory actions, for instance "move from $\langle x, y, z\rangle$ to $\langle x', y', z'\rangle$ or "look at $\langle x, y, z\rangle$". At this level, there are no object representations as there are at the Knowledge level and the Perceptuo-Motor level. There are no explicit declarative representations of any kind, only procedural representations (on the actuator side) and sensor data (on the sensory side). Primitive motor actions may typically be implemented in a robot control language like VAL, and some elementary data processing routines may be implemented in a sensory sub-system, like dedicated vision hardware. At this level, we also situate *reflexes*, which we consider to be low-level loops from sensors to actuators, controlled by simple thresholding devices, operating independently of higher-level mechanisms, and able to pre-empt the latter. We see reflexes as primitive mechanisms whose main purpose is prevention of damage to the hardware, or to put it in anthropomorphic terms, survival of the organism. As such they take precedence over any other behavior. When reflexes are triggered, the higher levels are made "aware" of this by the propagation of a signal, but they have no control over the reflex's execution, which is brief and simple (like a withdrawal reflex seen in people when they unintentionally stick their hand into a fire).[12][13] After the completion of a reflex, the higher levels regain control and must decide on how to continue or discontinue the activity that was interrupted by the reflex. Reflex-like processes may also be used to shift the focus of attention of the Knowledge level.

[12] An appropriate reflex for a robot (arm) might be to withdraw or stop when it meets too much mechanical resistance to its movement, as evidenced for instance by a sharp rise in motor current draw. Such a reflex could supplant the more primitive fuse protection of motors, and make an appropriate response by the system possible. Needless to say, a robot that can detect and correct problems is much more useful than one that merely blows a fuse and stops working altogether.

[13] The fact that the withdrawal reflex may not be as strong, or not present at all, when doing this intentionally may point to the need for top-down inhibition as well.

A.2.3 Symbol grounding: A non-Tarskian semantics

> Tarskian Semantics has nothing to say about how descriptions of objects in plans relate to the objects in the world [McDermott 1991, p. 13].

Let's digress for a moment to some esoteric matters of semantics and reference. One problem an agent has to solve is how to find and maintain a correspondence between a referent in the world and a symbol in an agent's world model. As noted above, the referent in the world is (by necessity) only indirectly considered via its embodied Perceptuo-Motor level representation, hence the problem becomes one of aligning the Knowledge level representations with the Perceptuo-Motor level representations. From the perspective of cognitive science, the problem has been labeled the *symbol grounding problem* [Harnad 1990]. The question is how to make the semantics of a robot's systematically interpretable Knowledge level symbols cohere equally systematically with the robot's interactions with the world, such that the symbols refer to the world on their own, rather than merely because of an external interpretation we place on them. This requires that the robot be able to discriminate, identify, and manipulate the objects, events, and states of affairs that its symbols refer to [Harnad 1992]. Grounding is accomplished in our architecture in part through the alignment of the Knowledge and Perceptuo-Motor levels. Elementary symbols at the Knowledge level are grounded in the sense that they only attach to "the right kind" of representations at the Perceptuo-Motor level. If we think of the Perceptuo-Motor level as implementing categorial perception (and perhaps "categorial action"), then the elementary symbols of the Knowledge level are the names attached to the categories. In other words, the alignment of the Knowledge and Perceptuo-Motor level constitutes an *internal referential semantic model* of elementary symbols. Note that, like McDermott, we do not take the Tarskian stance which requires the referents of symbols to be in the world; rather, they are system-internal, similar to what Hausser proposes [Hausser 1989], or what Harnad calls iconic representations: "proximal sensory projections of distal objects, events, and states of affairs in the world" [Harnad 1990]. The Knowledge level is the only level that is accessible for conscious reasoning, and also the only level that is accessible for inter-agent communication. Access to the Perceptuo-Motor level and the Sensori-Actuator level would not be useful for communication, as the representations and processing at these levels are too agent-centered and too agent-specific to be informative to other agents.

Since the Perceptuo-Motor level representations serving as the grounding for symbols of the Knowledge level are embodied (section A.2.4), equivalent symbols may have somewhat different semantics for different agents having different bodies. We don't see that as a problem, as long as the differences are not too large.[14] Indeed, we believe that this is quite realistic in human terms as well; no two persons are likely to have *exactly* the same semantics for their concepts, which nevertheless does not prevent them from understanding each other, grosso modo at least (cf. [Rapaport 1988]). The problems of translation and communication in general consist at least in part of establishing a correspondence between concepts (and symbols) used by the participants. It is helpful to be able to use referents in the external world as landmarks in the semantic landscape, but one consequence of embodied semantics is that *even* if it is possible to establish these common external referents for symbols, there is still no guarantee that the symbols will actually *mean* exactly the same thing, because in effect the same referent in the world is *not* the same thing to different agents. If we accept this view, it is clear that approaches to semantics based on traditional logical model theory are doomed to fail, because they *presuppose* "identity of referents" and an unambiguous mapping from symbols to referents, the same one for all agents. Another problem is of course the presupposition that all objects are uniquely identifiable. The use of deictic representations does not impose such a condition; as far as our agents are concerned, if it looks and feels the same, it is the same.[15] Nothing hinges on whether or not the objects in the agent's surroundings are *really* extensionally the same as the identical-looking ones that were there a moment ago or will be there a moment later.

A.2.4 Embodied representation

In section A.2.2.3 we already mentioned the use of embodied representations at the Perceptuo-Motor level. We now look at the principle of embodiment from a more abstract point of view.

One of the most general motivations behind our work is the desire to be able to "program" a robotic autonomous agent by requesting it to do something and have it "understand", rather than telling it how to do something in terms of primitive motions with little

[14] It is never a problem as long as agents need not communicate with the outside world (other agents), of course, cf. [Winston 1975].

[15] This is of course the "duck test", made famous by a former US president.

or no "understanding". For instance, we want to tell it to go find a red pen, pick it up, and bring it to us, and not have to program it at a low level to do these things.[16] One might say that we want to communicate with the robot at the *speech act* level. To do this, the agent needs a set of general-purpose perceptual and motor capabilities along with an "understanding" of these capabilities. The agent also needs a set of concepts which are similar enough to ours to enable easy communication. The best way to accomplish this is to endow the agent with embodied concepts, grounded in perception and action.

We define *embodiment* as the notion that the representation and extension of high level concepts is in part determined by the physiology (the bodily functions) of an agent, and in part by the interaction of the agent with the world. For instance, the extension of color concepts is in part determined by the physiology of our color perception mechanism, and in part by the visual stimuli we look at. The result is the establishment of a mapping between color concepts and certain properties of both the color perception mechanism and objects in the world. Another example is the extension of concepts of action: it is partly determined by the physiology of the agent's motor mechanisms, and partly by the interaction with objects in the world. The result is the establishment of a mapping between concepts of action and certain properties of both the motor mechanisms and objects in the world (what we might call "the shapes of acts").

At an abstract level, the way to provide an autonomous agent with human-like embodied concepts is to intersect the set of human physiological capabilities with the set of the agent's potential physiological capabilities, and endow the agent with what is in this intersection. To determine an agent's potential physiological capabilities, we consider it to be made up of a set of primitive actuators and sensors, combined with a general purpose computational mechanism. The physical limitations of the sensors, actuators, and computational mechanism bound the set of potential capabilities. For instance with respect to color perception, if the agent uses a CCD color camera (whose spectral sensitivity is usually wider than that of the human eye), combined with a powerful computational mechanism, we consider its potential capabilities wider than the human ones, and thus restrict the implemented capabilities to the human ones. We endow the agent with a color perception mechanism whose functional properties reflect the physiology of human color perception. That results in color

[16] Retrieving "canned" parameterized routines is still a low-level programming style that we want to avoid.

concepts that are similar to human color concepts. With respect to the manipulation of objects, most robot manipulators are inferior to human arms and hands, hence we restrict the implemented capabilities to the ones that are allowed by the robot's physiology. The robot's motor mechanism then reflects the properties of its own physiology, rather than those of the human physiology. This results in a set of motor concepts that is a subset of the human one. Embodiment also calls for body-centered and body-measured representations, relative to the agent's own physiology. We provide more details on embodiment in GLAIR in [Hexmoor et al. 1993c].

A.2.5 Alignment

When a GLAIR-agent notices something in its environment, it registers that it has come to know of an object. Regardless of whether the agent recognizes the type of the object, we want it to explicitly represent the existence of the object in the Knowledge level while processing sensory information about the object in the Perceptuo-Motor level. Similar to sensing objects, when properties of objects or relationships among objects are sensed by the GLAIR-agent, we want it to explicitly represent these properties and relationships, even if no more is known about them than the fact that they exist. We use unnamed intensional concepts for this purpose [Shapiro & Rapaport 1987].

Having sensed an object, an assertion is made about the object being sensed in the GLAIR Knowledge level. Once the object is no longer in the "field of perception", the assertion about its being sensed is removed. This is tantamount to disconnecting the relationship between the symbolic representation and the world. If at the Perceptuo-Motor level a previously sensed object is again being sensed, we reassert the fact that the object, the same one represented before in the Knowledge level, is being sensed. An example of this type of (unconscious) perception is when we look at an object, look away, and then look back at the same object. The unconscious level can provide a short term sensory memory in which memories of objects are stored, and when we see them from time to time, the conscious layer is alerted to that fact. We can think of this phenomenon as a type of *continuity in perception* at the unconscious level. We believe that if we assume this continuity, we should re-use previously constructed representations to represent again-sensed objects. In order for a GLAIR-agent to re-use its previously established representations about objects for again-sensed objects, we either have to assume that the agent has a continuity of perception at the unconscious layer or that a conscious matching of existing representations to sensed objects is performed.

A.2.6 Consciousness

As we pointed out above, we identify the Knowledge level with consciously accessible data and processing; the Perceptuo-Motor level with "hard-wired", not consciously accessible processing and data involved with motor control and perceptual processing; and the Sensori-Actuator level with the lowest-level muscular and sensor control, also not consciously accessible. The distinction of conscious (Knowledge) levels vs. unconscious (Perceptuo-Motor and Sensori-Actuator) levels is convenient as an anthropomorphic metaphor, as it allows us to separate explicitly represented and reasoned about knowledge from implicitly represented and processed knowledge. This corresponds *grosso modo* to consciously accessible and not consciously accessible knowledge for people.[17] Although we are aware of the pitfalls of introspection, this provides us with a rule of thumb for assigning knowledge (and skills, behaviors, etc.) to the various levels of the architecture. We believe that our organization is to some extent psychologically relevant, although we have not yet undertaken any experimental investigations in this respect. The real test for our architecture is its usefulness in applications to physical (robotic) autonomous agents (section A.3).

Knowledge in GLAIR can migrate from conscious to unconscious levels. In [Hexmoor et al. 1993a] we show how a video-game playing agent learns how to dynamically "compile" a game playing strategy that is initially formulated as explicit reasoning rules at the Knowledge level into an implicit form of knowledge at the Perceptuo-Motor level, a Perceptuo-Motor Automaton (PMA).

There are also clear computational advantages to our architectural organization. A Knowledge Representation and Reasoning system as used for the conscious Knowledge level is by its very nature slow and requires lots of computational resources.[18] The implementation mechanisms we use for the unconscious levels, such as PMAs, are much faster and

[17] The term "knowledge" should be taken in a very broad sense here.

[18] Many reasoning problems are NP-complete, meaning there are no polynomial-time deterministic algorithms known for solving them, or in plain English: they are very hard to solve in a reasonable amount of time (see e.g. [Levesque 1988]). Elephants don't stand a chance.

require much less resources. Since the three levels of our architecture are semi-independent, they can be implemented in a (coarse-grained) parallel distributed fashion; at least each level may be implemented on distinct hardware, and even separate mechanisms within the levels (such as individual reflex behaviors) may be. Our Robot Waiter agent, for instance, uses distinct hardware for the three levels (section A.3.1).

A.3 Applications

Our architecture as described in section A.2 can be populated with components that make up the machinery for mapping sensory inputs to *response* actions, as does Russell in [Russell 1991]. We now discuss some applications of GLAIR that we are currently developing.

Some important general features of GLAIR-agent are the following:

- Varieties of behaviors are integrated: We distinguish between deliberative, reactive, and reflexive behaviors. At the unconscious level, behavior is generated by mechanisms with the computational power of a finite state machine (or less), whereas, at the conscious level, behavior is generated via reasoning (of Turing Machine capabilities). As we move down the architectural levels, computational and representational power (and generality) is traded off for better response time and simplicity of control. Embodied representations aid in this respect (section A.2.4).

- We assume agents to possess a set of primitive motor capabilities. The motor capabilities are primitive in the sense that (a) they cannot be further decomposed, (b) they are described in terms of the agent's physiology, and (c) no reference is made to external objects. The second property of motor capabilities is so that the success of performing an action should depend only on the agent's bodily functions and proprioceptive sensing. For example, for a robot arm, we might have the following as its motor abilities: calibrate, close-hand, raise-hand, lower-hand, move.

- Our architecture provides a natural framework for modeling four distinct types of behavior, which we call reflexive, reactive, situated, and deliberative. Reflexive and reactive behaviors are predominantly unconscious behaviors, whereas situated and deliberative actions are conscious behaviors.

Reflexive behavior[19] occurs when sensed data produces a response, with little or no processing of the data. A reflex is immediate. The agent has no expectations about the outcome of its reflex. The reflexive response is not generated based on a history of prior events or projections of changing events, e.g., a gradual temperature rise. Instead, reflexive responses are generated based on spontaneous changes in the environment of the agent, e.g. a sudden sharp rise in temperature. In anthropomorphic terms, this is innate behavior that serves directly to protect the organism from damage in situations where there is no time for conscious thought and decision making, e.g., the withdrawal reflex when inadvertently touching something hot. Reflexive behavior does not require conscious reasoning or detailed sensory processing, so our lowest level, the Sensori-Actuator level, is charged with producing these behaviors. Our initial mechanism for modeling reflexive behavior is to design processes of the form T $\mapsto$ A, where T is a trigger and A is an action. A trigger can be a simple temporal-thresholding gate. The action A is limited to what can be expressed at the Sensori-Actuator level, and is simple and fast.

Reactive behavior requires some processing of data and results in *situated action* [Suchman 1988]. However, its generation is subconscious. *Situated action* refers to an action that is appropriate in the environment of the agent. In anthropomorphic terms, this is learned behavior. An example would be gripping harder when one feels an object is slipping from one's fingers, or driving a car and tracking the road. We use the term *tracking* to refer to an action that requires continual adjustments, like steering while driving. Examples of this type of reactive behavior are given in [Payton 1986, Anderson et al. 1991]. Situated behavior requires assessment of the state the system finds itself in (in some state space) and acting on the basis of that. It might be modeled by the workings of a finite state automaton, for example, the Micronesian behavior described in [Suchman 1988]. Situated action is used in *reactive planning* [Agre & Chapman 1987, Firby 1987, Schoppers 1987].

Deliberative behavior requires considerable processing of data and reasoning which results in action. In anthropomorphic terms, this is learned behavior that requires reasoning that can be modeled by a Turing machine (or first order logic), for example explicit planning and action.

We have developed an implementation mechanism for the Perceptuo-Motor-level which

[19] E.g., visual reflexes in [Regan & Beverly 1978]: Here responses are generated to certain visual stimuli that do not require detailed spatial analysis.

we call Perceptuo-Motor-automata (PMA), [Hexmoor & Nute 1992]. A PMA is a finite state machine in which each state is associated with an act and arcs are associated with perceptions. In each PMA, a distinguished state is used to correspond to the no-op act. Each state also contains an auxiliary part we call Internal State (IS). An IS is used in arbitrating among competing arcs. Arcs in a PMA are situations that the agent perceives in the environment. When a PMA arc emanating from a state becomes active, it behaves like an asynchronous interrupt to the act in execution in the state. This causes the PMA to stop executing the act in the state and to start executing the act at the next state at the end of the arc connecting the two states. This means that in our model the agent is never idle, and it is always executing an act. The primary mode of acquiring PMAs in GLAIR is by converting plans in the Knowledge level into PMAs through a process described in [Hexmoor & Nute 1992]. A PMA may become active as the result of an intention to execute an action at the Knowledge level. Once a PMA becomes active, sensory perception will be used by the PMA to move along the arcs. The sensory perceptions that form the situations on the arcs as well as subsequent actions on the PMA may be noticed at the Knowledge level. In general, the sensory information is filtered into separate streams for PMAs and for the Knowledge level.

A.3.1 A physical implementation: the Robot Waiter

We are developing an experimental setup in which a robot arm will set a dinner table in various configurations, guided by input from a color camera and controlled by a host computer.

The physical setup includes a dinner table, a supplies table containing kitchenware, a Puma 260 robot arm, a CCD camera, a PC-based color frame grabber, and a Sun 4/260 workstation host computer. In a later phase of this project, we hope to replace the Puma 260 robot arm with a larger Puma 560 model. Figure A.2 represents the setup.

The human operator instructs the agent to set the table for dinner, breakfast, etc., specifying by named color which objects to choose from the supplies table (color is not a sufficient feature to recognize objects, as each type of object is available in several colors). The camera is mounted in a fixed position overlooking both tables and the robot. This testbed provides a dynamic, yet controllable, environment in which an agent is placed so

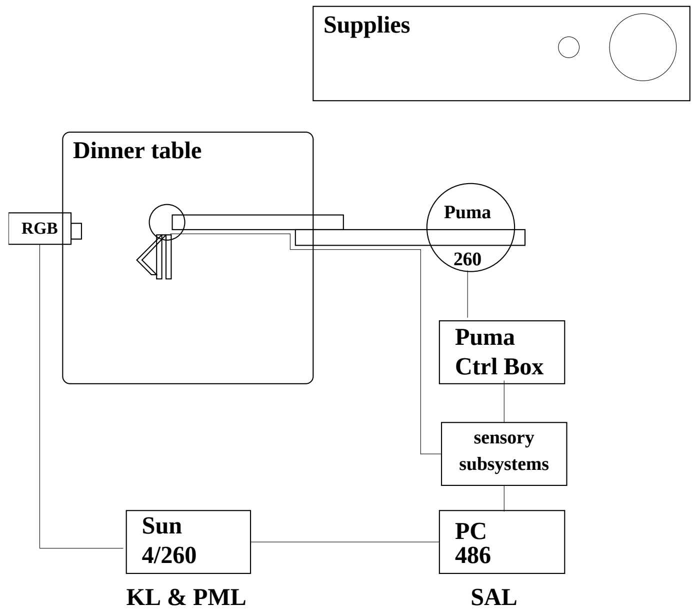


Figure A.2: The Robot Waiter physical setup.

as to facilitate empirical studies of its behavior. The places, number, kind, and color of objects is not fixed, and unannounced human intervention in the domain is allowed while the robot is carrying out its task.

We call the agent for this project the Robot Waiter (RW). RW is being developed in accordance with the principles of the GLAIR architecture. Figure A.3 schematically presents its structure. The categorizer uses domain constraints to determine what objects are in the visual field. It can also be told to look for a certain object, e.g., *a red cup*. The sensory memory acts as an attentional register, keeping track of the object that is being manipulated or is to be inspected.

Actions are transmitted from Knowledge level to Perceptuo-Motor level, e.g., *look for a red apple* and *pick it up*. Once the sensory memory contains an object matching the object desired, actions involving that object can be understood at the Perceptuo-Motor level. For instance, once a red apple is discovered and recorded in the sensory memory, an action like *pick it up* is expanded by a PMA into robot arm tasks to reach for the apple, grasp it, and lift it. A PMA is a implementation mechanism for routine activities at an "unconscious" level, [Hexmoor & Nute 1992, Hexmoor et al. 1992, Hexmoor et al. 1993a]. Each task involving a robot motion is subsequently submitted to the path constructor. Some motions may have to be decomposed to visit intermediate points in order to avoid fixtures in the

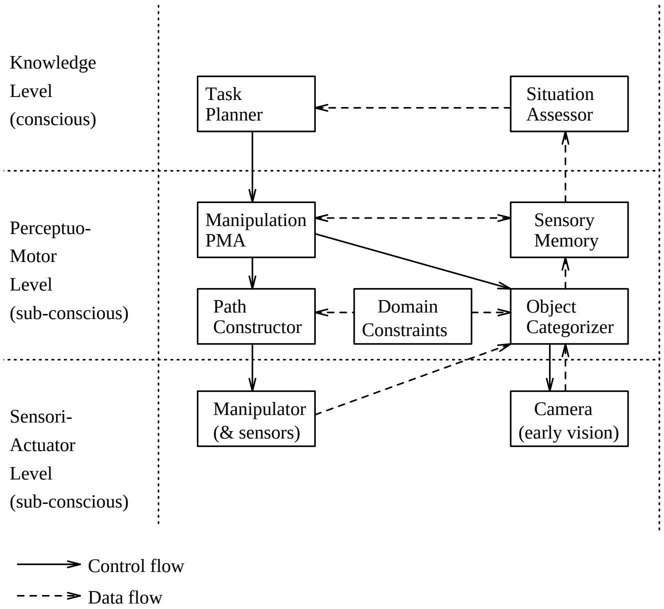


Figure A.3: Schematic representation of the Robot Waiter GLAIR-agent.

environment. The path constructor generates path segments for each robot motion. Each path segment generated by the path constructor is transmitted to the Sensori-Actuator level for execution.

RW incorporates an embodied model of color perception and color naming, modeled after the physiology of human color perception. This model allows the agent to (1) name colors shown to it, and express a typicality judgment, (2) point out examples of named colors in its environment, and (3) learn new names for colors. This model provides the perceptual grounding for a set of basic color terms [Berlin & Kay 1969] represented at the Knowledge level. The color domain was chosen as a case study for embodied perception because of the relative abundance of psychological, psychophysical and neurophysiological data in the literature [Lammens 1992]. It is a complex enough domain to allow the usefulness of embodiment for computational models of perception to be demonstrated, yet feasible enough to be implemented in actual autonomous agents.

The main components of the Robot Waiter system are listed below:

- A World model: In the general KRR system, we maintain a conscious world model.[20] This model explicitly represents the agent's knowledge (or beliefs) about the entities in

[20] Albus defines a world model as the agent's best estimate of objective reality [Albus 1991].

its world. Representations at this level may or may not correspond to representations at the Perceptuo-Motor level, as explained in section A.2.2. As much as it might be desirable to avoid building internal models of the world,[21] having some modeling capacity is necessary, we believe.

- Specialized knowledge bases: for instance knowledge about action selection, planning, learning, experimentation, and perception.

- A Kinematic/Perceptual model: At the Perceptuo-Motor level, we maintain a model of the simple agent-level physics of the objects of interest to the agent.[22] The kinematic/perceptual model models motor capacities and motor memory that might be implemented in different ways (e.g., purely procedurally or in a network of nodes and weighted links), but we prefer the declarative approach for its ease of interpretation and debugging. It also contains perceptual representations of perceived objects. Representations at this level are embodied and agent-centered (section A.2.4).

- Reactive processes: Also at the Perceptuo-Motor level, a number of independent processes monitor perceptual inputs, and control reactive behaviors of the agent. We need a mechanism for arbitrating among various reactive and other processes at the same level, which we have not worked out yet.

- Primitive motor and sensory actions: At the Sensori-Actuator level, a number of these primitive actions are implemented. A primitive motor action might be "move ahead", and a primitive sensory action might be "look at position $\langle x, y, z \rangle$" (which in turn may involve primitive motor actions). Sensing as such is not considered a primitive sensory *action*, and it goes on continuously.

- Reflexes: Also at the Sensori-Actuator level, a number of reflexes are implemented as low-level independent processes that monitor raw sensory data and can control actuators directly, temporarily pre-empting higher level control.

[21] Situated cognition and reactive planning are proponents of avoiding world modeling, e.g., [Brooks 1990, Suchman 1988].

[22] Even the prominent advocators of doing away with world models actually use a variety of models, some of which qualify as kinematic/perceptual models. For instance, see Chapman's work on Sonja [Chapman 1990] where Sonja has to build a convex hull of obstacles and compute angles in order to decide the best way to avoid them.

As of this writing, the Robot Waiter project is partially implemented, but not operational yet.

A.3.2 A simulation study: Air Battle

We are interested in modeling behavior generation by agents that function in dynamic environments. We make the following assumptions for the agent:

- The environment demands continual and rapid acting, e.g., playing a video-game.
- The impact of the agent's actions depends on the situations under which actions are applied and on other agents' actions.
- Other agents' actions are nondeterministic.
- The agent does not know about long term consequences (i.e., beyond the current situation) of its actions.
- The agent is computationally resource bounded. We assume that the agent needs time to think about the best action and in general there is not enough time.

To cope in dynamic environments, an agent which is resource bound needs to rely on different types of behaviors, for instance, reflexive, reactive, situated, and deliberative behaviors. Reflexive and reactive behaviors are predominantly "unconscious" behaviors, situated action may be either "unconscious" or "conscious", and deliberative actions are predominantly "conscious" behaviors. We assume that in general "conscious" behavior generation takes more time than "unconscious" behavior generation.

We have written a program, Air Battle Simulation (ABS), that simulates World War I style airplane dog-fights. ABS is an interactive video-game where a human player plays against a computer driven agent. The game runs on SparcStations and starts up by displaying a game window and a control panel window (figure A.4). The human player's plane is always displayed in the center of the screen. The aerial two-dimensional position of the enemy plane is displayed on the screen with the direction of flight relative to the human player's plane. The human player looks at the game screen to determine his airplane's position and orientation with respect to the enemy's plane. (S)he then uses the control panel to choose a move. A move is a combination of changing altitude, speed, and direction. When the human player presses the go button, the computer agent also selects a move. The game simulator then considers the human player's move and the computer agent's move to determine the outcome of moves, and updates the screen and the accumulated damage to planes. ABS simulates simultaneous moves this way. If a player's plane is close in altitude and position to the enemy plane, and the enemy is in frontal sight, the latter is fired on automatically (i.e., firing is not a separate action). The levels of damage are recorded in a side panel, and the game ends when one or both of the two player's planes are destroyed.

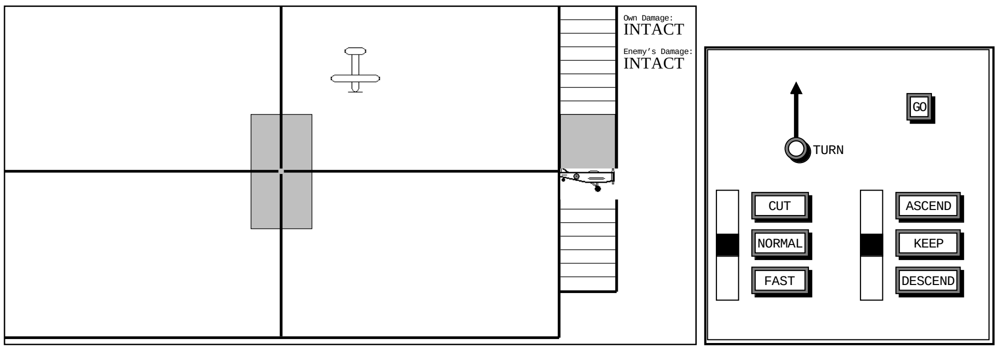


Figure A.4: Air Battle Simulation game window and control panel (see text).

The agent is developed in accordance with the principles of the GLAIR architecture. Figure A.5 schematically represents its structure. Initially, the agent has not acquired a PMA, and uses conscious level reasoning to decide what move to make. Once transitions are learned and cached in a PMA, the agent uses the PMA for deciding its next move whenever possible. By adding learning strategies, a PMA can be developed that caches moves decided at the Knowledge level for future use. Learning can be used to mark PMA moves that prove unwise and to reinforce moves that turn out to be successful. We are exploring these learning issues. We started ABS with an empty PMA and as the game was played, transitions of the PMA were learned. Also as the transitions were learned, when similar situations occurred and there was an appropriate PMA response, the PMA executed that action. As the game was played, we observed that the agent became more reactive since the PMA was increasingly used to generate behaviors instead of the Knowledge level.

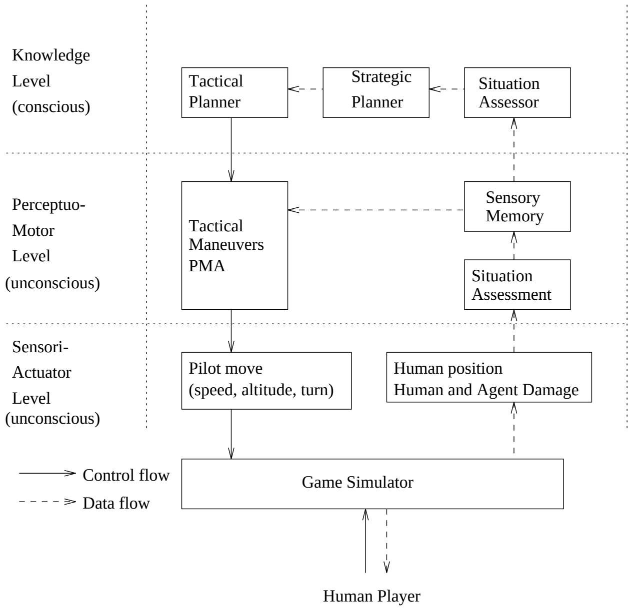


Figure A.5: Schematic representation of the Air Battle Simulation GLAIR-agent.

A.3.2.1 Improving "Unconscious" Behaviors

The rules of a PMA are pairs of situation/action. As it turns out, a situation can be paired up with multiple actions. The object of learning here is to learn which actions when associated with a situation yield a better result, i.e., the pilot ends up in a more desirable situation.

Some situations in ABS are more desirable for the pilot than others, e.g., being right behind the enemy and in shooting range. Let's assume that we can assign a goodness value G(s) to each situation s between -1 and 1. As the pilot makes a move, it finds itself in a new situation. This new situation is not known to the pilot since it also depends on the other pilot's move. Since the new situation is not uniquely determined by the pilot's move, the pilot's view of the game is not Markovian.

Q(s,a) is the evaluation of how appropriate action a is in situation s. **R**(s,a) is the goodness value of the state that the pilot finds itself after performing a in situation s. **R**(s,a) is determined as the game is played and cannot be determined beforehand. This is called the immediate reward. γ is a parameter between 0 and 1 that we plan to vary that to determine how important it is to be in the state that the pilot ends up in after his move. In reinforcement based learning this is known as the discount factor. We let $\mathbf{Q}$(s,a) = $\mathbf{R}$(s,a) + γ $\max_k \mathbf{Q}$(s',k) where situation s' results after the pilot performs a in s. At the start of game, all $\mathbf{Q}$(s,a) in the PMA are set to 1. As the game is played, $\mathbf{Q}$ is updated. As of this writing we are experimenting with setting appropriate parameters for $\mathbf{Q}$.

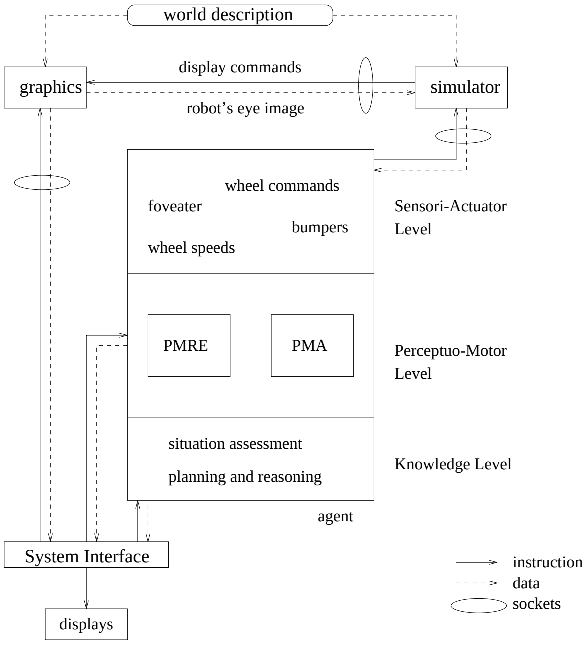
A.3.2.2 Observing Successful Patterns of Interaction in the World

We assumed that the agent does not know about long term consequences of its actions. Furthermore, the reinforcement based learning we described in the previous section assumes a Markovian environment. That is, the agent believes the world changes only due to its own actions. This makes it necessary to observe interactions with the world in order to learn sequences of actions. Over a finite number of actions, when the agent observes a substantially improved situation, chances are he has found a successful *Routine*. We record such detected *Routines* and as they re-occur, we increase our confidence in them. When our confidence in a *Routine* reaches a certain level, a concept is created at the Knowledge level of GLAIR for the routine and from then on, this routine can be treated as a single action at that level.

We plan to explore other learning techniques such as experimentation as a form of learning [Shen 1989]. We are also interested in developing experiments that will help in psychological validation of GLAIR and the learning strategies used in ABS. As of the time of writing ABS is fully operational, but several issues are still being investigated, as noted above.

A.3.3 A simulation study: the Mobile Robot Lab

We now describe the Mobile Robot Lab (MRL), a simulation environment we are developing for mobile robots that function as GLAIR-conformant autonomous agents. The simulation is relatively simple, but nevertheless provides a rich and realistic enough environment to function as a testbed for the development of physical GLAIR-agents. To make the simulation more realistic and less predictable, some stochastic properties are built in. More than one agent can be accommodated. A complete setup using MRL consists of a GLAIR-agent, a simulator with an incorporated description of a physical environment, and a graphical interface (figure A.6).



MRL system diagram

Figure A.6: Overview of a complete setup using MRL. It consists of a GLAIR-agent, a simulator with an incorporated model of a physical environment, and a graphical interface. Arrows represent direction of data flow among the components.

A.3.3.1 Emergent Behaviors

A major objective for this project is learning emergent behaviors. Like Agre with his improvised actions [Agre & Chapman 1987] and Brooks with his subsumption architecture [Brooks 1985] we believe complex behaviors emerge from interaction of the agent with its environment without planning. However, previous work in this area hard-coded a lot of primitive actions. Furthermore, it did not attempt to learn the improvised behavior. In this simulation, we plan to start with a minimal number of primitive actions and sensations. Our basis for this minimality and the choice of primitive actions is physiological. In other words, in our modeling an agent, we will choose actions that are physically basic for the agent's body as primitive. We then instruct the agent to perform tasks and in the midst of accomplishing this, we expect it to notice some types of behaviors emerge. An example of an emergent behavior we will explore is *moving toward an object*. We expect the agent to be learning to coordinate its wheel motions, starting from nothing more than the primitive sensation of contact with an external object, and the primitive actions of turning its motors independently on or off.

A.3.3.2 The physical environment description

The simulator uses a description of the physical environment that the simulated robot operates in. This description is easily modifiable (without reprogramming). It includes the physical characteristics of the mobile robot and the space in which it moves. A 2D bird's eye view of a typical room setup with a robot inside is show in figure A.7.

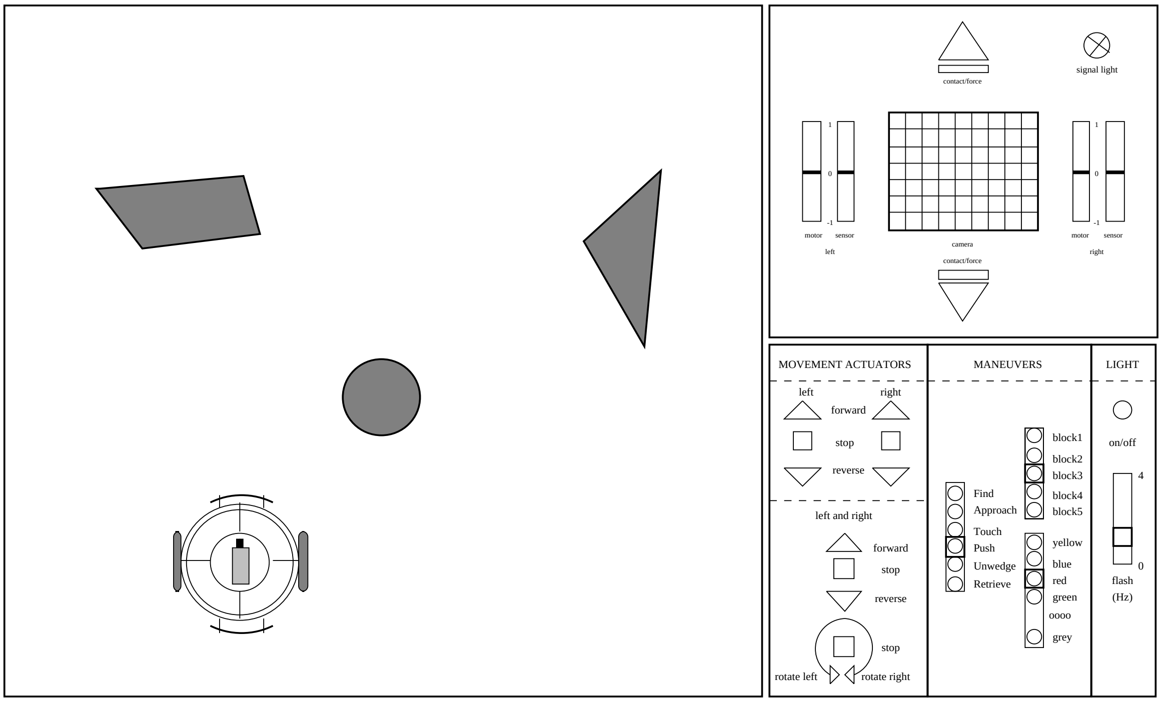


Figure A.7: A 2D bird's eye view of a typical room setup with a mobile robot inside (left), the Sensors and Actuators Display (upper right) and the Movement Control Panel (lower right). See text.

The room the robot moves is has a polygonal floor plan and vertical walls, and contains a number of solid objects with convex polygonal bases and vertical faces, each with an associated user-defined spectral power distribution (SPD).

Any number of robots may inhabit the room. They have two independently driven wheels on either side, and two small support wheels underneath in the front and the back. Furthermore a bumper bar front and back, with contact and force sensors built in, and a color camera on top, parallel to the direction of the driven wheels. The camera is fixed and mounted horizontally. The robot also has a non-directional light with a user-defined SPD on top, which it can switch on and off or flash.

A.3.3.3 The simulator

The simulator interfaces with the agent and with the graphical interface. It takes care of any I/O with the agent that would otherwise come from the sensors and go to the actuators of a real mobile robot. It also takes care of any I/O with the graphical interface, needed to keep the graphical display of the robot and its physical environment updated.

The simulator incorporates a simplified model of the physics of motion and sensing for the mobile robot. It continually updates the position of the robot depending on the rotation speed and direction of its wheels, and provides the agent with appropriate sensory data about wheel rotation and contact with objects. It also prevents the robot from going "through" walls or objects. It provides simulated camera input to the agent. Camera input is simplified in that it consists of a 9x7 pixel array (square pixels), with each pixel represented as an RGB triplet. This simplified camera view is computed and passed to the simulator by the graphical interface, on the basis of the 3D perspective views (see below).

The simulator incorporates a simplified lighting model to determine the appearance (color) of objects in the room. Light sources can either be point sources or homogeneous diffuse sources. Each light source has its own SPD. Each object has its own spectral reflectance function. All objects are assumed to be Lambertian reflectors.

To enhance the realism of the simulation and to introduce some uncertainty into the environment, all robot controls and sensors work stochastically. Sensor readings vary stochastically over time around the "true" value, and the same is true for actuator output. We assume normal distributions for all such variations, with variable standard deviations.

A.3.3.4 The graphical interface

The graphical interface provides a real-time view of the robot and the robot's environment, using data obtained from the simulator (figure A.7). It consists of a 2D display showing a bird's eye view of the room, the objects, and the robot in it, a sensors and actuators monitor display, and a 3D perspective display that shows the environment from the robot's point of view (not shown in figure A.7). The graphical interface also contains a control panel display which provides manual control over the robot's movements and maneuvers. A movement command for a robot is composed by the user by selecting power levels and directions of rotation for each of the two wheels. A maneuver is a higher-level instruction for the robot. Initially, it can be one of 1) find a block, 2) approach a block, 3) touch a block, 4) push a block, 5) unwedge a block, and 6) retrieve a block. Blocks can be referenced by color (not necessarily unique) or by a numeric identifier. The control panel also allows the robot's signal light to be switched on or off.

A.3.3.5 The agent

The autonomous agent for this project, with its simulated mobile robot body, conforms to the GLAIR architecture.

A.3.3.5.1 Primitive abilities. The agent has the following primitive sensations:

- Speed and direction of rotation for each of the left and right wheel motors are independently sensed. Sensed values for each motor range from 2 foot/sec of wheel circumference to -2 foot/sec of wheel circumference (in increments of 0.1).
- The bumpers in the front and the back can sense initial contact between the bumper and external object as well as reaction forces due to pushing. Sensor values range from 0 to 2, in increments of 1.
- The camera inputs an 9x7 array of RGB triplets, each triplet representing an average value over the corresponding portion of the camera's field of view.

The agent has the following primitive actions:

- Speed and direction of rotation for each of the left and right wheel motors are independently controlled. Actuation values for each motor increase/decrease by 1/10 foot/sec of wheel circumference in forward and reverse direction. Brakes can be applied to each wheel which can bring it to stop. We assume negligible acceleration/deceleration times.
- The light on top of each robot is either on, off, or flashing with a frequency of 1-4 Hz (in increments of 1 Hz).

A.3.3.5.2 The Sensori-Actuator level. Sensing and acting are paired up at this level to form reflex behaviors. The following reflex behaviors are implemented. Reflexes have direct access to sensors and actuators, and each may use its own representations and implementation strategies; we list the input and output domains.

- Stop reflex 1: **bumpers** $\mapsto$ **motor controls**; if the robot bumps into anything, it stops moving immediately.
- Force limit reflex: **motor controls** $\times$ **motor sensors** $\mapsto$ **motor controls**; if the sensed speed is significantly different from the speed control settings, the robot stops moving. This may occur when something is holding the robot back, for instance.
- Attention reflex: **camera input** $\mapsto$ **motor controls**; if anything moves in the robot's peripheral visual field, it stops and turns towards the location of movement.
- Stop reflex 2: **camera input** $\mapsto$ **motor controls**; if anything comes close in the robot's field of view, it stops moving.
- Escape reflex: **camera input** $\mapsto$ **motor controls**; if anything approaches the robot fast, the robot runs away.[23]

A.3.3.5.3 The Perceptuo-Motor Level. At the Perceptuo-Motor level, primitive sensations and actions are processed and combined to form more complex perceptions and behaviors. We list some of these below. We are currently investigating the use of foveal vision for the agent, which would alter some of the descriptions below.

A signal, object-seen, is generated to denote having seen an object. Signals will be generated to denote that the object is in the right, center, or left field of view. If an object in the field of view becomes larger (due to getting closer), a signal is generated that the object is object-bigger. Similarly, the object in the field of view will get smaller when departing from the object, signaled by object-smaller. If the object becomes an obstacle for the robot, the vision system generates an object-too-close signal. If anything moves in the robot's peripheral visual field, it will generate the signal moving-object (together with a position signal). Another complex perception is the trajectory of robot movement. As the

[23]Inhibiting this reflex might turn the agent from a chicken with a long life expectancy into an eagle with short life expectancy.

robot senses its wheel speed and direction of rotation combined with visual input, it builds a trajectory of movement. This can be used in building a map of the domain.

When the robot is in motion, it will use cues from its environment to guide its behavior selections and subsequent learning. In order for the robot to guide its behavior, we need to associate rewards with actions that result in desirable sensations. For instance, if the robot wants (at the Knowledge level) to touch an object and is taking actions to move towards the object, and the object is in the left field of view and the robot moves left (increases its right wheel motor speed) it will bring the object to the center of the field of view. The action of turning left in this situation will be positively rewarded. The result of learning (sequences of) actions will be recorded as a PMA. For instance, touching an object will evolve into a PMA. Even after the initial learning of a PMA for a complex action, learning will continue to improve the PMA.

For each behavior, a triple of <A, S, R> will be defined. A is the set of primitive actions, S is a set of sensations, and R is a set of rewards. For example, for the behavior of touching, A = {left wheel forward increase speed, left wheel forward decrease speed, right wheel forward increase speed, right wheel forward decrease speed}; S = {object is bigger, object is smaller, object is in the left field of view, object is in the right field of view, object is in the center of the field of view, object is too close, contact is made}; R = {object is bigger $+1$, object is smaller -1, object is in the left field of view -1, object is in the right field of view -1, object is in the center of field of view $+1$, object is too close $+1$, contact is made $+1$} (numbers ranging from $+1$ to -1 are rewards with $+1$ denoting desirable and -1 denoting undesirable).

Below is a list of behaviors and percepts we want the robot to learn. We will provide the robot with appropriate rewards for these behaviors. Below we give a list of emergent behaviors that the robot will learn completely on its own (no rewards).

- Touch behavior: approach an object until contact is made.
- Proximity percepts: proximity to an object, derived from camera data. (e.g., see all the signals we listed above in the example of touching behavior).
- Approach behavior: approach an object until it is in proximity.
- Block percept: recognizing something in the field of view as a block.

- Push a block
- Find a block
- Unwedge a block
- Explore/map the room

When more than one agent is present, behaviors at this level may include finding or hiding from other agents, following or running from other agents, playing games (hide and seek, for instance).

Earlier we defined behaviors in terms of PMAs. We consider subsets[24] of PMAs to also be behaviors. Percepts and behaviors listed below are emergent, i.e., they are learned but the agent did not intend to learn them. The robot is never told about these behaviors. They are learned in PMAs, as in the PMA for touching objects.

- Proprioceptive movement percepts and movement behaviors: moving forward, backward, turning left forward, turning right forward, turning left backward, turning right backward, rotating left, and rotating right. Each of these is a percept as well as a behavior.

Some other percepts and behaviors situated at the Perceptuo-Motor level are listed below. They are hard-wired in the initial implementation, but could conceivably be learned or emergent as well. As a rule of thumb, we consider the perceptual representation required to perform discrimination tasks to be hard-wired, and those required to perform identification tasks to be learned (or learnable).

- Color percepts: in perceptual color space coordinates, derived from camera data.
- Color hallucination behavior: imagine the surroundings in different colors by warping the perceptual color space.
- Shape percepts: simple object shapes like square, rectangle, tall, short, narrow, wide, irregular, big, small, derived from camera data.

Also at this level, basic emotions like fear and curiosity may be implemented as particular types of behaviors (aligned with symbolic labels at the Knowledge level).

[24] Let's define a subset of a PMA to be a disjoint PMA, i.e., B is a subset of A iff all components of B are subsets of components of A and no transitions exist between actions in B and those in A-B.

A.3.3.5.4 The Knowledge Level. At the Knowledge level, all of the percepts and behaviors of the Perceptuo-Motor level are represented, but in a more symbolic fashion. For instance, colors and shapes have names. The representations at the two levels are connected via the alignment mechanism discussed above. Also at this level is a symbolic map of the room and the objects in it, and the current position of the agent. In general, planning and some learning activities can originate at this level, and reasoning about the environment and the agent's actions, perceptions, goals, desires, states, etc. is confined to this level only. Concepts of space and time would also be represented at this level, perhaps as emergent concepts from the behavior of the agent in its environment.

A.4 Concluding Remarks

We have presented a general architecture for autonomous agents that integrates behavior-based architectures with traditional architectures for symbolic systems. The architecture specifies how an agent establishes and maintains a conscious connection with its environment while mostly unconsciously processing sensory data, and filtering information for conscious processing as well as for reflexive and reactive acting. We ended our paper by instantiating the architecture with several (physical and simulated) agents embedded in their environment, in various stages of implementation. We believe our work can contribute towards integrating traditional ungrounded symbol systems with the newer physically grounded systems. Combining an elephant's body with a man's[25] mind makes for an awesome combination.

A.5 Acknowledgments

We appreciate comments made on an earlier draft of this paper by Phil Agre, Chris Brown, Stevan Harnad, Donald Nute, John Pollock, Beth Preston, William Rapaport, and Tim Smithers. Goofs are entirely attributable to the authors, of course.

[25] He-man or She-man.

Appendix B

Derivation of Color Spaces

This appendix lists some equations for deriving various color spaces from the CIE XYZ standard, and from each other, in addition to some CIE chromaticity-related equations and functions. The Mathematica code is listed verbatim below, I believe it is not too hard to read.

These are color space transforms of various kinds.

```
(* Transforms of fundamentals etc.
   Includes both linear and non-linear transforms. The former are
   represented as 3x3 matrices, the latter algorithmically. *)

BeginPackage["transforms`", {"common`", "CIE`", "chromaticity`"}]

(* The symbols appearing below, before the start of the private part, will
   be exported from the package defined above. The usage information will
   be displayed by the help command. *)

XyzToRgbTV::usage = "Linear transform from XYZ to RGB TV coordinates"
RgbTVToXyz::usage = "Linear transform from RGB TV to XYZ coordinates"
XyzToRgbCie::usage = "Linear transform from XYZ to RGB CIE coordinates"
RgbCieToXyz::usage = "Linear transform from RGB CIE to XYZ coordinates"
RgbCieToLum::usage = "Linear transform from RGB CIE to luminance"
ChromRgbCie::usage = "RGB CIE chromaticity coordinates"
XyzToRgbFCC::usage = "Linear transform from XYZ to RGB FCC coordinates"
RgbFCCToXyz::usage = "Linear transform from RGB FCC to XYZ coordinates"
RgbFCCToLum::usage = "Linear transform from RGB FCC to luminance"
ChromRgbNTSC::usage = "RGB NTSC chromaticity coordinates"
XyzToRgbNTSC::usage = "Linear transform from XYZ to RGB NTSC coordinates"
RgbNTSCToXyz::usage = "Linear transform from RGB NTSC to XYZ coordinates"
ChromRgbCRT::usage = "RGB CRT chromaticity coordinates"
XyzToRgbCRT::usage = "Linear transform from XYZ to RGB CRT coordinates"
RgbCRTToXyz::usage = "Linear transform from RGB CRT to XYZ coordinates"
```

```
RgbToI123::usage = "Linear transform from generic RGB to Ohta's I123
    coordinates"
I123ToRgb::usage = "Linear transform from Ohta's I123 to generic RGB
    coordinates"
XyzToI123::usage = "Linear transform from XYZ to I123 coordinates, using
    RGB CRT as the intermediate norm"
I123ToXyz::usage = "Linear transform from I123 to XYZ coordinates, using
    RGB CRT as the intermediate norm"
YiqToRgb::usage = "Linear transform from YIQ to generic RGB coordinates"
RgbToYiq::usage = "Linear transform from generic RGB to YIQ coordinates"
XyzToYiq::usage = "Linear transform from XYZ to YIQ coordinates, using RGB
   NTSC as the intermediate norm"
YiqToXyz::usage = "Linear transform from YIQ to XYZ coordinates, using RGB
   NTSC as the intermediate norm"
ChromRgbSony::usage = "Typical Sony monitor RGB chromaticities"
ChromWSony::usage = "Typical Sony monitor white chromaticities"
TriStimWSony::usage = "Typical Sony monitor while tristimulus values"
XyzToRgbSony::usage = "Linear transform from XYZ to RGB Sony coordinates"
RgbSonyToXyz::usage = "Linear transform from RGB Sony to XYZ coordinates"
RgbCRTToRgbSony::usage = "Linear transform from RGB CRT to RGB Sony
  coordinates"
XyzToLmsVW::usage = "Linear transform from XYZ to LMS Vos & Walraven
     coordinates"
LmsVWToXyz::usage = "Linear transform from LMS Vos & Walraven to XYZ
     coordinates"
XyzToLmsE::usage = "Linear transform from XYZ to LMS Estevez coordinates"
LmsEToXyz::usage = "Linear transform from LMS Estevez to XYZ coordinates"
XyzToLmsSP::usage = "Linear transform from XYZ to LMS Smith & Pokorny
     coordinates"
LmsSPToXyz::usage = "Linear transform from LMS Smith & Pokorny to XYZ
     coordinates"
XyzToH81::usage = "Linear transform from XYZ to Hurvich & Jameson opponent
   coordinates"
H81ToXyz::usage = "Linear transform from Hurvich & Jameson opponent to XYZ
   coordinates"
RgbCRTToH81::usage = "Linear transform from RGB CRT to Hurvich & Jameson
      opponent coordinates"
H81ToRgbCRT::usage = "Linear transform from Hurvich & Jameson opponent to
      RGB CRT coordinates"
LmsVWToH81::usage = "Linear transform from LMS Vos & Walraven to Hurvich &
     Jameson opponent coordinates"
H81ToLmsVW::usage = "Linear transform from Hurvich & Jameson to LMS Vos &
     Walraven coordinates"
LmsVWToWWAW::usage = "Linear transform from LMS Vos & Walraven to Werner &
      Wooten subject AW opponent coordinates"
WWAWToLmsVW::usage = "Linear transform from Werner & Wooten subject AW
      opponent to LMS Vos & Walraven coordinates"
RgbCRTToWWAW::usage = "Linear transform from RGB CRT to Werner & Wooten
       subject AW opponent coordinates"
WWAWToRgbCRT::usage = "Linear transform from Werner & Wooten subject AW
       opponent to RGB CRT coordinates"
XyzToWWAW::usage = "Linear transform from XYZ to Werner & Wooten subject AW
    opponent coordinates"
```

```
WWAWToXyz::usage = "Linear transform from Werner & Wooten subject AW
    opponent to XYZ coordinates"
RgbToHsi::usage = "RgbToHsi[{r_,g_,b_}] returns {h,s,i} in {[0,2 Pi],
   [0,1], [0,1]} corresponding to {r_,g_,b_} in [0,1]."
XyzToUvl::usage = "XyzToUvl[{{x_,y_,z_}, {xI_,yI_,zI_}}] returns {u,v,l} in
   {R,R,[0,100]}, corresponding to {x_,y_,z_} in R+ and
   {xI_,yI_,zI_} in R+ (the latter representing illuminant
   color). This is the CIE 1976 L*u*v* space (CIELUV)."
diffUvl::usage = "diffUvl[{u1_,v1_,l1_}, {u2_,v2_,l2_}] returns the CIELUV
  color difference between the colors given by
  {u1_,v1_,l1_} and {u2_,v2_,l2_}, both in {R,R,[0,100]}.
  This amounts to Eucidean distance in CIELUV space."
UvlToHcl::usage = "UvlToHcl[{u_,v_,l_}] returns {hue, chroma, lightness} in
   {[0,2Pi], R+, [0,100]} corresponding to {u_,v_,l_} in
   {R,R,[0,100]}, representing a color in CIELUV
   coordinates. Chroma changes with changing lightness and
   constant chromaticity."
UvlToHsl::usage = "UvlToHsl[{u_,v_,l_}] returns {hue, saturation, lightness} in
   {[0,2Pi], R+, [0,100]} corresponding to {u_,v_,l_} in
   {R,R,[0,100]}, representing a color in CIELUV
   coordinates. Saturation does not change with changing
   lightness and constant chromaticity."
XyzToAbl::usage = "XyzToAbl[{{x_,y_,z_}, {xI_,yI_,zI_}}] returns {a,b,l} in
   {R,R,[0,100]}, corresponding to {x_,y_,z_} in R+ and
   {xI_,yI_,zI_} in R+ (the latter representing illuminant
   color). This is the CIE 1976 L*a*b* space (CIELAB)."
diffAbl::usage = "diffAbl[{a1_,b1_,l1_}, {a2_,b2_,l2_}] returns the CIELAB
  color difference between the colors given by
  {a1_,b1_,l1_} and {a2_,b2_,l2_}, both in {R,R,[0,100]}.
  This amounts to Eucidean distance in CIELAB space."
AblToHcl::usage = "AblToHcl[{a_,b_,l_}] returns {hue, chroma, lightness} in
   {[0,2Pi], R+, [0,100]} corresponding to {a_,b_,l_} in
   {R,R,[0,100]}, representing a color in CIELAB
   coordinates. Chroma changes with changing lightness and
   constant chromaticity."
AblToHsl::usage = "AblToHsl[{a_,b_,l_}] returns {hue, saturation, lightness} in
   {[0,2Pi], R+, [0,100]} corresponding to {a_,b_,l_} in
   {R,R,[0,100]}, representing a color in CIELAB
   coordinates. Saturation does not change with changing
   lightness and constant chromaticity."
XyzToSVF::usage = "XyzToSVF[{{X_,Y_,Z_}, {XW_,YW_,ZW_}}] returns {F1,F2,VY}
   in {R,R,R+} corresponding to {X_,Y_,Z_} in R+ and
   {XW_,YW_,ZW_} in R+. The latter represents the XYZ
   values of white. {F1,F2,VY} are the opponent coordinates
   F1 and F2 and the lightness magnitude VY of the SVF
   uniform color space."
Rgb2Hls::usage = "Rgb2Hls[{r_,g_,b_}] returns {h,l,s} in {[0,2Pi], [0,1],
  [0,1]} corresponding to {r_,g_,b_} in [0,1]."
Hls2Rgb::usage = "Hls2Rgb[{h_,l_,s_}] returns {r,g,b} in [0,1]
  corresponding to {h_,l_,s_} in {[0,2Pi], [0,1], [0,1]}."
Rgb2Hsv::usage = "Rgb2Hsv[{r_,g_,b_}] returns {h,s,v} in {[0,2Pi], [0,1],
  [0,1]} corresponding to {r_,g_,b_} in [0,1]."
Hsv2Rgb::usage = "Hsv2Rgb[{h_,s_,v_}] returns {r,g,b} in [0,1]
```

```
  corresponding to {h_,s_,v_} in {[0,2Pi], [0,1], [0,1]}."

Begin["`Private`"]

(* The symbols appearing below are private to this package, and will not be
   exported. *)

(* LINEAR TRANSFORMS *)

(* Ency of AI *)

(* Note: EAI 1992 edition CV article says linear transforms should be
   scaled s.t. each component has a max of 1; I don't do this. May be
   better for statistical feature counting, but not for my purpose. This
   will distort the relative positions of colors in the space. *)

XyzToRgbTV := {r0TV, g0TV, b0TV} *
      {{0.587, -0.164, -0.089}, {-0.301, 0.611, -0.0087},
       {0.0178, -0.0362, 0.274}}

RgbTVToXyz := Inverse[XyzToRgbTV]

(* Compute scaling factors to preserve E response relative to Cie XYZ
   functions. This does not preserve absolute magnitudes of transformed
   primaries, so they are renormalized afterwards. The
   transforms are defined with delayed evaluation, so that the scaling
   factors can be redefined at any time and the transforms will be modified
   accordingly. Note that TV fundamentals are equalized on standard
   illuminant C, not on equal-energy white as the XYZ functions are. *)

{r0TV, g0TV, b0TV} = {1,1,1}
{r0TV, g0TV, b0TV} = 1 / (XyzToRgbTV . TriStimC)
{r0TV, g0TV, b0TV} =
  RgbNormFactorFromTxr[XyzToRgbTV, Cie31List] {r0TV, g0TV, b0TV}

(* relation Cie 1931 XYZ to Cie real primaries 700, 546.1, 435.8 nm, see
   McIlwain & Dean 1956 Eq 4-4 p. 48 *)

RgbCieToXyz1 = {{2.7690, 1.7518, 1.1300}, {1.0000, 4.5907, 0.0601},
        {0.0000, 0.0565, 5.5943}}

XyzToRgbCie1 = Inverse[RgbCieToXyz1]

XyzToRgbCie := {r0Cie, g0Cie, b0Cie} * XyzToRgbCie1

RgbCieToXyz := Inverse[XyzToRgbCie]

{r0Cie, g0Cie, b0Cie} = {1,1,1}
{r0Cie, g0Cie, b0Cie} = 1 / (XyzToRgbCie . TriStimE)  (* note eq on E *)
{r0Cie, g0Cie, b0Cie} =
  RgbNormFactorFromTxr[XyzToRgbCie, Cie31List] {r0Cie, g0Cie, b0Cie}

RgbCieToLum = {1, 4.5907, 0.0601} (* luminance contributions *)
```

```
ChromRgbCie  = {{0.735, 0.265}, (* from rogers85 *)
        {0.274, 0.717},
{0.167, 0.009}}

(* I have used ChromRgbCie to compute matching functions from the
   chromaticities, with TxrFromChrom[ChromRgbCie, ChromE], which gives the
   same results as the transformation above. That means that, most likely,
   TxrFromChrom is working ok, I can trust it to compute matching functions
   based on chromaticities. *)

(* McIlwain & Dean conversions between Cie 1931 XYZ and FCC RGB used as the
   primaries for color tv. These are equalized on standard illuminant C as
   reference white, with luminance of this white taken as unity. Eqns 4-11
   p. 62.

   After scaling, RgbFCC is vitually identical to RgbTV. *)

RgbFCCToXyz1 = {{0.608, 0.174, 0.200}, {0.299, 0.587, 0.114},
        {0.000, 0.0662, 1.112}}

(* compute inverse transform rather than using eq 4-12, which
   is apparently wrong: 1.191X should read 1.907X, perhaps 1.911X *)

XyzToRgbFCC1 = Inverse[RgbFCCToXyz1]

XyzToRgbFCC := {r0FCC, g0FCC, b0FCC} * XyzToRgbFCC1

RgbFCCToXyz := Inverse[XyzToRgbFCC]

{r0FCC, g0FCC, b0FCC} = {1,1,1}
{r0FCC, g0FCC, b0FCC} = 1 / (XyzToRgbFCC . TriStimC)
{r0FCC, g0FCC, b0FCC} =
  RgbNormFactorFromTxr[XyzToRgbFCC, Cie31List] {r0FCC, g0FCC, b0FCC}

(* luminance signal as used in color tv, eq 8-2 p. 121, without gamma
   correction *)

RgbFCCToLum = {0.30, 0.59, 0.11}

(* Sun VideoPix doc: 0.299, 0.587, 0.114 for NTSC *)

(* NTSC from Xyz chromaticity coordinates, from Hill 1990, note 3 p. 574
   After scaling, RgbNTSC is vitually identical to RgbTV. *)

ChromRgbNTSC = {{0.670, 0.330},
        {0.210, 0.710},
{0.140, 0.080}}

XyzToRgbNTSC = TxrFromChrom[ChromRgbNTSC, ChromC]

RgbNTSCToXyz = Inverse[XyzToRgbNTSC]
```

```
(* typical CRT color monitor from chromaticity coordinates, from Hill
   1990, table 16.2 p. 574, and rest of the chapter. Note this is equalized
   on CIE D65, not on C as the NTSC functions are.
   These are slightly different from the values given in rogers85. *)

ChromRgbCRT = {{0.628, 0.330},      (* rogers85: 0.628 0.346 *)
       {0.285, 0.590},      (*   0.268 0.588 *)
       {0.1507, 0.060}}      (*   0.150 0.070 *)

(* these are fairly close to chromaticities computed on positive parts only
   of RgbFCC matching curves, using this kind of computation:
   {Function[l, If[R[l] < 0, 0, R[l]]], Function[l, If[G[l] < 0, 0, G[l]]],
    Function[l, If[B[l] < 0, 0, B[l]]]} *)

(* used a transform computed from chromaticity coordinates rather than the
   one given in Hill. The differences are considerable. *)

XyzToRgbCRT = TxrFromChrom[ChromRgbCRT, ChromD65]
  (* Hill has the following:
       {r0CRT, g0CRT, b0CRT} *
       {{2.739, -1.119, 0.138}, {-1.145, 2.2029, -0.333},
        {-0.424, 0.033, 1.105}} *)

RgbCRTToXyz = Inverse[XyzToRgbCRT]

(* equalize on source D65 for white *)

{r0CRT, g0CRT, b0CRT} = {1,1,1}
{r0CRT, g0CRT, b0CRT} = 1 / (XyzToRgbCRT . TriStimD65)
{r0CRT, g0CRT, b0CRT} =
  RgbNormFactorFromTxr[XyzToRgbCRT, Cie31List] {r0CRT, g0CRT, b0CRT}


(* EAI Ohta's I1,I2,I3 space. This is a linear transform of RGB. Note the
   purpose of this trafo, described in the article. *)

RgbToI123 = {{1/3, 1/3, 1/3}, {1, 0, -1}, {-1/2, 1, -1/2}}

I123ToRgb = Inverse[RgbToI123]

XyzToI123 = RgbToI123 . XyzToRgbCRT

XyzToI123 = NormFactorFromTxr[XyzToI123, 550, 610, 450, Cie31List] XyzToI123

I123ToXyz = Inverse[XyzToI123]

(* RGB<-->YIQ transforms from mma Packages/Graphics/Colors.m *)

YiqToRgb = {{1, .95, .625}, {1, -.28, -.64}, {1, -1.11, 1.73}}
  (* rogers85 has a typo omitting the 1 from the last number:
   {{1, .956, .623}, {1, -.272, -.648}, {1, -1.105, .705}} *)

RgbToYiq = Inverse[YiqToRgb]
```

```
  (* rogers85:
        {{.299, .587, .114}, {.596, -.274, -.322}, {.211, -.522, .311}} *)

XyzToYiq = RgbToYiq . XyzToRgbNTSC

XyzToYiq = NormFactorFromTxr[XyzToYiq, 550, 610, 540, Cie31List] XyzToYiq

YiqToXyz = Inverse[XyzToYiq]

(* Sony CRT color monitor from Xyz chromaticity coordinates, from
   twj@wri.com. These are the values used in Mathematics as "approximations
   to typical color monitors", i.e. the Sony tubes as used in Mac's etc.
   Typical gamma for these tubes is 1.8.
   Agrees with values gotten from Sun. *)

ChromRgbSony = {{0.625, 0.340},
        {0.280, 0.595},
        {0.155, 0.070}}

ChromWSony = {0.283, 0.298}
TriStimWSony = {0.283, 0.298, 0.419}

XyzToRgbSony = TxrFromChrom[ChromRgbSony, ChromWSony]

RgbSonyToXyz = Inverse[XyzToRgbSony]

(* use the following transform to display RgbCRT colors on a Sony monitor *)

RgbCRTToRgbSony = XyzToRgbSony . RgbCRTToXyz

(* Vos & Walraven 1978 fundamentals. W/o scaling factors, the resulting
   functions are on luminance basis. From Wyszecki & Stiles 2nd ed, p. 612
   ff. The transformation is relative to the Vos-Judd 1978 modified CIE
   standard functions (Cie31VxBar, Cie31VyBar, Cie31VzBar). *)

XyzToLmsVW := {r0VW, g0VW, b0VW} *
      {{0.1551646, 0.5430763, -0.0370161},
       {-0.1551646, 0.4569237, 0.0296946},
       {0.0, 0.0, 0.0073215}}

LmsVWToXyz := Inverse[XyzToLmsVW]

(* equalized on E *)

{r0VW, g0VW, b0VW} = {1,1,1}; {r0VW, g0VW, b0VW} = 1 / (XyzToLmsVW . TriStimE)
{r0VW, g0VW, b0VW} =
  RgbNormFactorFromTxr[XyzToLmsVW, Cie31VList] {r0VW, g0VW, b0VW}


(* Estevez 1979 fundamentals, according to Valberg et al 1986. The
   transformation is relative to the CIE 1931
   standard functions (Cie31VxBar, Cie31VyBar, Cie31VzBar). *)
```

```
XyzToLmsE := {r0E, g0E, b0E} *
     {{0.3841, 0.7391, -0.0650},
      {-0.3471, 1.1463, 0.0870},
      {0.0, 0.0, 0.5610}}

LmsEToXyz := Inverse[XyzToLmsE]

(* equalized on E *)

{r0E, g0E, b0E} = {1,1,1}; {r0E, g0E, b0E} = 1 / (XyzToLmsE . TriStimE)
{r0E, g0E, b0E} =
  RgbNormFactorFromTxr[XyzToLmsE, Cie31VList] {r0E, g0E, b0E}


(* Smith & Pokorny 1975 fundamentals. W/o scaling factors, the resulting
   functions are on luminance basis. From Wyszecki & Stiles 2nd ed, p. 612
   ff. The transformation is relative to the Judd 1951 modified CIE
   standard functions (Cie31JxBar, Cie31JyBar, Cie31JzBar). *)

XyzToLmsSP := {r0SP, g0SP, b0SP} *
      {{0.15514, 0.54312, -0.03286},
       {-0.15514, 0.45684, 0.03286},
       {0, 0, 0.00801}}

LmsSPToXyz := Inverse[XyzToLmsSP]

(* equalized on E *)

{r0SP, g0SP, b0SP} = {1,1,1}; {r0SP, g0SP, b0SP} = 1 / (XyzToLmsSP . TriStimE)
{r0SP, g0SP, b0SP} =
  RgbNormFactorFromTxr[XyzToLmsSP, Cie31JList] {r0SP, g0SP, b0SP}


(* Hurvich 81 opponent responses *)

XyzToH81 := {gr0H81, by0H81, wb0H81} {{1, -1, 0}, {0, 0.4, -0.4}, {0, 1, 0}}

H81ToXyz := Inverse[XyzToH81]

RgbCRTToH81 := XyzToH81 . RgbCRTToXyz

H81ToRgbCRT := XyzToRgbCRT . H81ToXyz

LmsVWToH81 := XyzToH81 . LmsVWToXyz

H81ToLmsVW := XyzToLmsVW . H81ToXyz

{gr0H81, by0H81, wb0H81} = {1,1,1}
{gr0H81, by0H81, wb0H81} =
  OppNormVectorFromTxr[XyzToH81] {gr0H81, by0H81, wb0H81}

(* W&W 79, w.r.t. normalized V&W fundamentals, observer AW *)
```

```
LmsVWToWWAWOrig = {{2.38, -2.87, 0.54}, {0.93, -0.36, -1.03}, {0.85, 0.15, 0.01}}

(* this is the normalizing matrix for VW fundamentals. The argument to
   DiagonalMatrix is computed as 1 / {rBarVWMax, gBarVWMax, bBarVWMax} *)

normVW = DiagonalMatrix[{1.03759, 0.844756, 0.618546}]

(* derive WWAW fns from unnormalized VW fundamentals and from rgb *)

LmsVWToWWAW := LmsVWToWWAWOrig . normVW

WWAWToLmsVW := Inverse[LmsVWToWWAW]

RgbCRTToWWAW := LmsVWToWWAW . XyzToLmsVW . RgbCRTToXyz

WWAWToRgbCRT := Inverse[RgbCRTToWWAW]

XyzToWWAW := {gr0WWAW, by0WWAW, wb0WWAW} (LmsVWToWWAW . XyzToLmsVW)

WWAWToXyz := Inverse[XyzToWWAW]

{gr0WWAW, by0WWAW, wb0WWAW} = {1,1,1}
{gr0WWAW, by0WWAW, wb0WWAW} =
  OppNormVectorFromTxr[XyzToWWAW] {gr0WWAW, by0WWAW, wb0WWAW}

(* NON-LINEAR TRANSFORMS *)

(* EAI: HSI space used in color vision work. Note this is not a linear
   transform! Note also the singularities in this fn at low intensities.
   Modified the transform for for i below .03, otherwise get strange
   results with black, e.g. At low intensities, H and S are meaningless,
   almost random, using the unmodified transform. Assume r,g,b, in [0,1].
   This version is better for black, but still unreliable at low
   intensities, e.g. for brown. *)

RgbToHsi[{r_,g_,b_}] := Module[ {h, s, i, rn, gn, bn}, (
  i = (r+g+b)/3;
  If[i<.02, {0,0,i}, (
    {rn,gn,bn} = {r,g,b}/i;
    s = 1 - Min[{rn,gn,bn}];
    x = ArcCos[(2 rn-gn-bn) /
      (Sqrt[6] Sqrt[(rn-1/3)^2 + (gn-1/3)^2 + (bn-1/3)^2])];
    h = If[bn<=gn, x, 2 Pi-x];
    {h,s,i}
    )]
  )]


(* EAI: L*u*v* perceptually uniform color space, superseding the earliest
   uniform space UVW. Quantities denoted by postfix I in the definition
   of L*u*v* refer to the incident illumination color (!). Note this is not
   a linear transform of XYZ space. Note the singularities for low
   intensity here too. This incorporates the modification for low values of
```

```
   y/yI as described in W&S p. 165. This is CieLUV 1976. Note these
   function actually use the order U, v, l to be compatible with H81 etc,
   which have brightness as the third dimension (vertical). *)

XyzToUvl[{{x_,y_,z_}, {xI_,yI_,zI_}}] :=
  Module[ {ls, us, vs, u, v, uI, vI, d, dI}, (
    d = x + 15 y + 3 z;
    dI = xI + 15 yI + 3 zI;
    u = If[d>0, 4 x / d, 0];
    uI = If[dI>0, 4 xI / dI, 0];
    v = If[d>0, 9 y / d, 0];
    vI = If[dI>0, 9 yI / dI, 0];
    ls = If[(y/yI)>0.008856, 116 (y/yI)^(1/3) - 16, 903.3 (y/yI)];
    us = 13 ls (u-uI);
    vs = 13 ls (v-vI);
    {us,vs,ls}
    )]

(* color difference in CieLUV 1976 is just Euclidean distance *)

diffUvl[{u1_,v1_,l1_}, {u2_,v2_,l2_}] :=
  Sqrt[(u1-u2)^2 + (v1-v2)^2 + (l1-l2)^2]

(* Derive perceived lightness, chroma, and hue from uvl coordinates. Chroma
   is similar to saturation, except the latter (as defined in W&S) does not
   change with changing lightness and constant chromaticity, but the former
   does. *)

UvlToHcl[{u_,v_,l_}] := If[u==0 && v==0, {0,0,l},
  {ArcTan[u,v], Sqrt[u^2 + v^2], l}]   (* chroma *)

UvlToHsl[{u_,v_,l_}] := If[u==0 && v==0, {0,0,l},
  If[l==0, {0,0,l}, {ArcTan[u,v], Sqrt[u^2 + v^2] / l, l}]] (* sat *)


(* Cie 1976 L*a*b* space and color-difference formula, from W&S p. 166.
   Note again the order a, b, L of the components. *)

XyzToAbl[{{x_,y_,z_}, {xI_,yI_,zI_}}] :=
  Module[ {ls,as,bs}, (
    ls = If[(y/yI)>0.008856, 116 (y/yI)^(1/3) - 16, 903.3 (y/yI)];
    as = 500 (If[(x/xI)>0.008856, (x/xI)^(1/3), 7.787 (x/xI) + 16/116] -
      If[(y/yI)>0.008856, (y/yI)^(1/3), 7.787 (y/yI) + 16/116]);
    bs = 200 (If[(y/yI)>0.008856, (y/yI)^(1/3), 7.787 (y/yI) + 16/116] -
      If[(z/zI)>0.008856, (z/zI)^(1/3), 7.787 (z/zI) + 16/116]);
    {as,bs,ls}
  )]


(* color difference in CieLAB 1976 is just Euclidean distance *)

diffAbl[{a1_,b1_,l1_}, {a2_,b2_,l2_}] :=
  Sqrt[(a1-a2)^2 + (b1-b2)^2 + (l1-l2)^2]
```

```
(* Derive perceived lightness, chroma, and hue from Abl coordinates. *)

AblToHcl[{a_,b_,l_}] := If[a==0 && b==0, {0,0,l},
  {ArcTan[a,b], Sqrt[a^2 + b^2], l}]   (* chrom *)

(* same but with saturation in stead of chroma -- not in W&S? analogous to
   Uvl to Hcl, Hsl transforms above. *)

AblToHsl[{a_,b_,l_}] := If[a==0 && b==0, {0,0,l},
  If[l==0, {0,0,l}, {ArcTan[a,b], Sqrt[a^2 + b^2] / l, l}]] (* sat *)


(* SVF uniform color space, from Valberg et al 1986, app. A *)

(* helper funcions *)
v1[Y_] := If[Y<=0.0043, 0,
             ((100 Y - 0.43)^0.51)/((100 Y - 0.43)^0.51 + 31.75)]

k[VY_] := 0.140 + 0.175 VY

v2[Y_] := Module[ {VY},
  VY = 40 v1[Y];
  If[Y<=0.001 k[VY], 0,
     (((100 Y / k[VY]) - 0.1)^0.86) /
     (((100 Y / k[VY]) - 0.1)^0.86 + 103.2) ]
  ]

XyzToSVF[{{X_,Y_,Z_}, {XWhite_,YWhite_,ZWhite_}}] :=
  Module[ {S1,S2,S3,VY,p1,p2,F1,F2},
    (* center relative cone absorptions of color sample to the white
       stimulus (or the light source), aka von Kries transformation. Trafo
       A1 given in Valberg et al 86 p. 1732, differs by scaling factors
       only from XyzToLmsE. Since the sample values are scaled w.r.t. the
       white values, this doesn't matter. *)
    {S1,S2,S3} = (XyzToLmsE . {X,Y,Z}) / (XyzToLmsE . {XWhite,YWhite,ZWhite});
    (* lighness magnitude VY *)
    VY = 40 v1[Y];
    (* first opponent stage coordinates p1, p2 *)
    p1 = v1[S1] - v1[Y];
    p2 = If[S3<=Y, v1[Y] - v1[S3], v2[Y] - v2[S3]];
    (* second opponent stage coordinates F1 and F2 *)
    F1 = 700 p1 - 54 p2;
    F2 = 96.5 p2;
    {F1,F2,VY}
    ]



(* rgb to hls triplets, from Hill 90. In this hls space, all pure colors
   (red, green, blue, yellow, cyan, magenta) lie on the 0.5 lightness
   plane and are all equally saturated (1.0), with black and white at the
   bottom and top of the double cone. This is not very realistic in
```

```
   psychophysical terms. For greys, h=0 although strictly speaking it is
   undefined. Otherwise, h is given in radians, with 0 = red.

   Foley & Van Dam 1982(84) have essentially the same routine, except they
   differ on the sign of one of the terms! (see code). *)

Rgb2Hls[{r_,g_,b_}] := Module[ {mx, mn, rc, gc, bc, h, l, s}, (
  (* convert {r,g,b}, each in [0,1], to {h,l,s}, in {[0,2Pi], [0,1], [0,1]} *)
  mx = Max[r,g,b];
  mn = Min[r,g,b];
  l = (mx + mn) / 2.0; (* lightness *)
  (* compute saturation *)
  If[mx == mn, s = h = 0, ( (* grey *)
    (* chromatic color *)
    If[l<=0.5, s = (mx-mn) / (mx+mn), s = (mx-mn) / (2-mx-mn)];
    (* Hill has (2-mx+mn) for the last term, which is wrong as evidenced by
       computing some transforms to HLS and back to RGB. *)
    rc = (mx-r) / (mx-mn);      (* hue *)
    gc = (mx-g) / (mx-mn);
    bc = (mx-b) / (mx-mn);
    If[r==mx, h = bc-gc,
      If[g==mx, h = 2+rc-bc,
        If[b==mx, h = 4+gc-rc]]];
    h = 60 h;
    If[h<0, h = h+360]
    )];
  {h (Pi/180), l, s}
  )]

(* this is from Foley & vDam 82, p 619 *)

Hls2Rgb[{h_,l_,s_}] := Module[ {value, h1, r, g, b, m1, m2}, (
  (* convert {h,l,s} in {[0,2Pi], [0,1], [0,1]} to {r,g,b}, each in [0,1] *)
  value[n1_,n2_,hue_] := Module[ {hue1}, (
    If[hue>360, hue1=hue-360, If[hue<0, hue1=hue+360, hue1=hue]];
    Which[
      hue1<60, n1+(n2-n1) hue1/60,
      hue1<180, n2,
      hue1<240, n1+(n2-n1) (240-hue1)/60,
      True, n1])];
  h1 = h (180/Pi); (* convert to degrees *)
  If[l<=0.5, m2=l(1+s), m2=l+s-l s];
  m1=2 l-m2;
  If[s==0, {r,g,b} = {l,l,l},
    {r,g,b} = {value[m1,m2,h1+120], value[m1,m2,h1], value[m1,m2,h1-120]}];
  {r,g,b}
  )]


(* HSV hexcone after Smith 78, from F&vD 82, 613ff. As in HLS above,
   saturation is relative to gamut represented by model, not to Cie chart,
   i.e. not the same as purity. HSV = Hue, Saturation, Value (lightness,
   after Munsell value I suppose). For the following conversion functions,
```

```
   {r,g,b} are in [0,1], {h, s, v} in {[0, 2Pi], [0,1], [0,1]}. For
   undefined values of h, 0 is returned. *)

Rgb2Hsv[{r_,g_,b_}] := Module[ {mx, mn, rc, gc, bc, h, s, v}, (
  (* convert {r,g,b}, each in [0,1], to {h,s,v}, in {[0,2Pi], [0,1], [0,1]} *)
  mx = Max[r,g,b];
  mn = Min[r,g,b];
  v = mx; (* value *)
  If[mx != 0, s = (mx-mn) / mx, s = 0]; (* saturation *)
  If[s==0, h = 0, ( (* grey *)
    (* chromatic color *)
    rc = (mx-r) / (mx-mn);      (* distance from red *)
    gc = (mx-g) / (mx-mn);
    bc = (mx-b) / (mx-mn);
    If[r==mx, h = bc-gc, (* between Y, Magenta *)
      If[g==mx, h = 2+rc-bc, (* between Cyan, Y *)
        If[b==mx, h = 4+gc-rc]]];  (* between Magenta, Cyan *)
    h = 60 h; (* convert to degrees *)
    If[h<0, h = h+360] (* make nonnegative *)
    )];
  {h (Pi/180), s, v} (* convert to radians *)
  )]

Hsv2Rgb[{h_,s_,v_}] := Module[ {r, g, b, h1, i, f, p, q, t}, (
  (* convert {h,s,v}, in {[0,2Pi], [0,1], [0,1]} to {r,g,b}, each in [0,1] *)
  h1 = h (180/Pi); (* convert to degrees *)
  If[s==0, {r,g,b} = {v,v,v}, (
    If[h1==360, h1=0];        (* must use local variable for h *)
    h1 = h1/60; (* convert to [0,6] *)
    i = Floor[h1]; (* integer part of h *)
    f = h1-i; (* fractional part of h *)
    p = v (1-s);
    q = v (1-(s f));
    t = v (1-(s (1-f)));
    Switch[i,
      0, {r,g,b} = {v,t,p},
      1, {r,g,b} = {q,v,p},
      2, {r,g,b} = {p,v,t},
      3, {r,g,b} = {p,q,v},
      4, {r,g,b} = {t,p,v},
      5, {r,g,b} = {v,p,q}] )];
    {r,g,b}
    )]


End[]       (* private *)
EndPackage[] (* package *)
```

These are CIE chromaticity-related equations and functions.

```
(* Cie chromaticity functions and related stuff *)

BeginPackage["chromaticity`", {"common`", "CIE`", "mygraphics`"}]

(* The symbols appearing below, before the start of the private part, will
   be exported from the package defined above. The usage information will
   be displayed by the help command. *)

SetDefaultCieFns::usage = "SetDefaultCieFns[CieFnsList] sets the default
  CIE basis functions (x, y, z) to the list CieFnsList."

ChromDiagram::usage = "Parametric plot of the CIE chromaticity diagram"

ChromDiagramColor::usage =
  "Color parametric plot of the CIE chromaticity diagram"

PlotChromList::usage = "PlotChromList[{{x,y}, ...}] plots the CIE
  chromaticity diagram and points with chromaticities {x,y}."

PlotChromListColor::usage = "PlotChromList[{{x,y}, ...}] plots the CIE
  chromaticity diagram and points with chromaticities {x,y}, in color."

TxrFromChrom::usage =
  "TxrFromChrom[{{Rx, Ry}, {Gx, Gy}, {Bx, By}}, {Wx,Wy}] computes an
  XYZ to RGB transform from RGB chromaticity coordinates {{Rx, Ry}, ...},
  equalized on the white point given by chromaticity coordinates {Wx, Wy}.
  The result is normalized for max[rgb]=1."

ChromFromTxr::usage =
  "ChromFromTxr[txr] returns chromaticity coordinates {x,y} of primaries,
  given a linear transform txr from Cie XYZ color matching functions to the
  color matching functions of those primaries."

ChromFromSPD::usage =
  "ChromFromSPD[{f1, f2, ...}] returns chromaticity coordinates {x, y} of
  SPD's f1, f2, ... (functions of wavelength)."

LBlue::usage = "Typical wavelength resulting in perception of blue."
LGreen::usage = "Typical wavelength resulting in perception of green."
LYellow::usage = "Typical wavelength resulting in perception of yellow."
LRed::usage = "Typical wavelength resulting in perception of red."

PlotTxr::usage =
  "PlotTxr[matrix, label:\"\", style:rgbColors, CieFns:Cie31List] plots
  the primaries defined by matrix, a linear transform from XYZ (with basis
  functions CieFns) to the primaries, and labels the plot with label. Uses
  style to plot the individual functions."

PlotTxrP::usage =
  "PlotTxrP[matrix, label:\"\", style:rgbColors, CieFns:Cie31List]
  plots the primaries defined by matrix, a linear transform from XYZ (with
  basis functions CieFns) to the primaries, and labels the plot with label.
```

```
  Plots only the positive lobes of the primaries (useful for RGB primaries
  only). Uses style to plot the individual functions."

PlotTxrPM::usage =
  "PlotTxrPM[matrix, label:\"\", style:rgbColors, CieFns:Cie31List]
  plots the primaries defined by matrix, a linear transform from XYZ (with
  basis functions CieFns) to the primaries, and labels the plot with label.
  Plots only the major positive lobes of the primaries (useful for RGB
  primaries only). Uses style to plot the individual functions."

NIntTxr::usage =
  "NIntTxr[matrix] computes the integrals over the visible wavelength range
  of the RGB primaries defined by matrix, a linear transform from XYZ
  coordinates."

NIntTxrP::usage =
  "NIntTxr[matrix] computes the integrals over the visible wavelength range
  of the RGB primaries defined by matrix, a linear transform from XYZ
  coordinates. Uses only the positive lobes of the primaries."

NIntTxrPM::usage =
  "NIntTxr[matrix] computes the integrals over the visible wavelength range
  of the RGB primaries defined by matrix, a linear transform from XYZ
  coordinates. Uses only the major positive lobes of the primaries."

Txr2ChromGamut::usage =
  "Txr2ChromGamut[txr,range,points,label] plots the gamut of the
  primaries defined by txr (a linear transform from XYZ to the primaries)
  in a square region of the chromaticity diagram with sides (0,range), with
  a vertical and horizontal resolution of points. The plot is labeled with
  label. Actually displayed colors are approximations for CRTs. Good values
  for range and points are 0.85 and 64."

GamutNTSC::usage = "NTSC gamut as computed by function Txr2ChromGamut"
GamutCRT::usage = "CRT gamut as computed by function Txr2ChromGamut"
GamutLMS::usage = "LMS gamut as computed by function Txr2ChromGamut"

Tor2Gamut::usage =
  "Tor2Gamut[tor, brightness, hrange, vrange, points, label] computes the
  gamut of the opponent primaries defined by tor (a linear transform from
  opponent functions to rgb) at the specified brightness level. The plot is
  displayed in 3D, with x and y (opponent) coordinates going from -hrange
  to hrange and z (brightness) coordinates going from 0 to vrange. The
  x and y resolution is given by points, and the plot is labeled label.
  Varying the brightness level while keeping vrange constant can be used to
  generate successive equal brightness planes through the color space for
  animation. The acually displayed colors are approximations for typical
  CRTs."

Tor2GamutN::usage = "Like Tor2Gamut, but with RGB intensities normalized
  for max(r,g,b)=1. This looses all intensity information, but can be used
  to judge hues better than with Tor2Gamut."
```

```
AnimH81CRT::usage = "Animated plot of constant brightness planes through
  the Hurvich 81 color opponent space, transformed to CRT RGB coordinates."

AnimH81Lms::usage = "Animated plot of constant brightness planes through
  the Hurvich 81 color opponent space, transformed to LMS coordinates."

AnimWWAWCRT::usage = "Animated plot of constant brightness planes through
  the Werner&Wooten AW color opponent space, transformed to CRT RGB
  coordinates."


Begin["`Private`"]

(* The symbols appearing below are private to this package, and will not be
   exported. *)

(* default basis set to use for CIE XYZ space *)

XyzList := Cie31List

SetDefaultCieFns[CieFns_:Cie31List] := XyzList = CieFns

(* chromaticity coordinates of monochromatic light *)

z[l_] := XyzList[[3]][l] /
  (XyzList[[1]][l] + XyzList[[2]][l] + XyzList[[3]][l])
y[l_] := XyzList[[2]][l] /
  (XyzList[[1]][l] + XyzList[[2]][l] + XyzList[[3]][l])
x[l_] := XyzList[[1]][l] /
  (XyzList[[1]][l] + XyzList[[2]][l] + XyzList[[3]][l])

chrom[l_] := {x[l], y[l]}
chrom3[l_] := {x[l], y[l], 1 - x[l] - y[l]}

cPoint[{x_, y_}] := Graphics[Circle[{x,y}, 0.008]]
cPointL[l_] := cPoint[chrom[l]]

ChromDiagram := Show[ParametricPlot[{x[l], y[l]}, {l,380,750}, Frame->True,
  AspectRatio->Automatic, GridLines->Automatic, DisplayFunction->Identity],
  Graphics[{Line[{{x[380], y[380]}, {x[750], y[750]}}]}],
  DisplayFunction->Identity]]

PlotChromList[list_] := Show[ChromDiagram, Map[cPoint, list],
     DisplayFunction->$DisplayFunction]

(* generic rgb or xyz functions based on transforms *)

X[l_] := XyzList[[1]][l]
Y[l_] := XyzList[[2]][l]
Z[l_] := XyzList[[3]][l]

R[l_] := txr[[1]] . {XyzList[[1]][l], XyzList[[2]][l], XyzList[[3]][l]}
G[l_] := txr[[2]] . {XyzList[[1]][l], XyzList[[2]][l], XyzList[[3]][l]}
```

```
B[l_] := txr[[3]] . {XyzList[[1]][l], XyzList[[2]][l], XyzList[[3]][l]}

(* positive lobes only *)

RP[l_] := If[R[l]<0, 0, R[l]]
GP[l_] := If[G[l]<0, 0, G[l]]
BP[l_] := If[B[l]<0, 0, B[l]]

RPM[l_] := If[l<525, 0, RP[l]] (* major positive lobe only *)

(* inverses of xyz to rgb *)

XI[l_] := trx[[1]] . {R[l], G[l], B[l]}
YI[l_] := trx[[2]] . {R[l], G[l], B[l]}
ZI[l_] := trx[[3]] . {R[l], G[l], B[l]}

plotRGB := Plot[{R[l], G[l], B[l]}, {l,380,760}]
plotXYZ := Plot[{X[l], Y[l], Z[l]}, {l,380,760}]
plotXYZI := Plot[{XI[l], YI[l], ZI[l]}, {l,380,760}]

(* integrals of fundamentals *)

intXYZ := {NIntegrate[X[l], {l,380,760}], NIntegrate[Y[l], {l,380,760}],
   NIntegrate[Z[l], {l,380,760}]}
intXYZI := {NIntegrate[XI[l], {l,380,760}], NIntegrate[YI[l], {l,380,760}],
    NIntegrate[ZI[l], {l,380,760}]}
intRGB := {NIntegrate[R[l], {l,380,760}], NIntegrate[G[l], {l,380,760}],
   NIntegrate[B[l], {l,380,760}]}
intRGBP := {NIntegrate[RP[l], {l,380,760}], NIntegrate[GP[l], {l,380,760}],
   NIntegrate[BP[l], {l,380,760}]}
intRGBPM := {NIntegrate[RPM[l], {l,380,760}], NIntegrate[GP[l], {l,380,760}],
   NIntegrate[BP[l], {l,380,760}]}

(* compute integrals of primaries specified as an XYZ->RGB transform *)

NIntTxr[m_, CieFns_:XyzList] := (
  txr = m; XyzList = CieFns;
  intRGB )

NIntTxrP[m_, CieFns_:XyzList] := (
  txr = m; XyzList = CieFns;
  intRGBP )

NIntTxrPM[m_, CieFns_:XyzList] := (
  txr = m; XyzList = CieFns;
  intRGBPM )

(* The following function computes a transformation from Cie XYZ color
   matching functions to RGB color matching functions, given a set
   of primaries specified by chromaticity coordinates only, and the
   chromaticities of the alignment white. This function has been verified
   against the results given in Rogers 85; the only difference is the
   normalization factor. This method seems a lot easier than Rogers'. *)
```

```
TxrFromChrom[{{Rx_, Ry_}, {Gx_, Gy_}, {Bx_, By_}}, {Wx_,Wy_},
  CieFns_:XyzList] :=
  (* Compute an XYZ to RGB transform from RGB chromaticity coordinates,
     equalized on the white point given by chromaticity coordinates. The
     result is also normalized for max[rgb]=1. *)
  Module[{Rz, Gz, Bz, Wz, m, sf}, (
    Rz = 1 - Rx - Ry;
    Gz = 1 - Gx - Gy;
    Bz = 1 - Bx - By;
    Wz = 1 - Wx - Wy;
    m := Inverse[{{Rx, Gx, Bx}, {Ry, Gy, By}, {Rz, Gz, Bz}}];
    (* equalize on white *)
    m = (1 / (m . {Wx,Wy,Wz})) m;
    (* scale for max[rgb]=1 *)
    m = RgbNormFactorFromTxr[m, CieFns] m;
    m
    )]


(* This function computes chromaticity coordinates of primaries, given a
   transform from Cie XYZ color matching functions to the color matching
   functions of those primaries. Note color matching functions are NOT the
   same as SPD's, hence you cannot use the technique of function
   ChromFromSPD here. In fact the computation is a lot simpler. Use this
   function as the inverse of function TxrFromChrom. This function has been
   verified against the results given in Rogers85, p. 395 ff. *)

ChromFromTxr[txr_] := Module[ {trx = Inverse[txr]}, (
  Map[{#[[1]] / (#[[1]] + #[[2]] + #[[3]]),
       #[[2]] / (#[[1]] + #[[2]] + #[[3]])}&,
    {trx . {1,0,0}, trx . {0,1,0}, trx . {0,0,1}}]
  )]

(* Compute chromaticity coordinates from SPD's as fn of wavelength. This
   function has been verified on Cie sources A, B, C, D65, and E. The
   results are virtually the same as the chromaticities found in the
   literature. *)

ChromFromSPD[plist_, CieFns_:XyzList] := Module[{F, result, sum}, (
  XyzList = CieFns;
  Map[ Function[ el, (
    result = Map[ Function[x, NIntegrate[x, {l,380,760}]],
      {F[l] XyzList[[1]][l], F[l] XyzList[[2]][l],
       F[l] XyzList[[3]][l]} /. F->el ];
    sum = Plus @@ result;
    Take[result / sum, 2] ) ], plist ]
  ) ]


(* some typical wavelengths for spectral colors *)

{LBlue, LGreen, LYellow, LRed} = {475, 500, 580, 700}
```

```
(* for displaying fundamentals *)

PlotTxr[m_,label_:"",style_:rgbColors,CieFns_:XyzList] := (
  txr = m; XyzList = CieFns;
  Plot[{R[l], G[l], B[l]}, {l,380,760}, PlotRange->All, PlotLabel->label,
    PlotStyle->style]
  )

PlotTxrP[m_,label_:"",style_:rgbColors,CieFns_:XyzList] := (
  txr = m; XyzList = CieFns;
  Plot[{RP[l], GP[l], BP[l]}, {l,380,760}, PlotRange->All,
    PlotLabel->label, PlotStyle->style]
  )

PlotTxrPM[m_,label_:"",style_:rgbColors,CieFns_:XyzList] := (
  txr = m; XyzList = CieFns;
  Plot[{RPM[l], GP[l], BP[l]}, {l,380,760}, PlotRange->All,
    PlotLabel->label, PlotStyle->style]
  )

PlotTxo[m_,label_:"",style_:rgbColors,CieFns_:XyzList] := (
  txr = m; XyzList = CieFns;
  Plot[{R[l], G[l], B[l]}, {l,380,760}, PlotRange->All, PlotLabel->label,
    PlotStyle->style]
  )

(* displaying Cie diagram with colored outline *)

(* this function assigns RGB colors to wavelengths, scaling them up to
   maximum intensity while preserving rgb hue *)

lcolor[l_] := Module[ {r,g,b}, (
  {r,g,b} = {RPM[l], GP[l], BP[l]};
  {r,g,b} = (1 / Max[r,g,b]) {r,g,b};
  RGBColor[r,g,b]
  )]

(* only ParametricPlot3D seems to take a color argument *)
(* the purple line does not show for some reason *)
(* delayed evaluation because XyzToRgbCRT has to be defined first *)

ChromDiagramColor := (
  txr = transforms`XyzToRgbCRT; XyzList = Cie31List;
  Show[ParametricPlot3D[{x[l], y[l], 0, lcolor[l]},
    {l,380,750},
  PlotRange->{{0,.80}, {0,.85}, {-.001,.001}},
  ViewPoint->{0,0,25}, Axes->{True,True,False},
  DisplayFunction->Identity],
  Graphics3D[{RGBColor[1,0,1],
    Line[{{x[380], y[380],0}, {x[750], y[750],0}}]}],
  Background->RGBColor[.5,.5,.5],
  DisplayFunction->Identity]
```

```
  )

cPointColor[{x_, y_}] :=
  Graphics3D[{GrayLevel[0], PointSize[0.02], Point[{x,y,0}]}]
cPointLColor[l_] := cPointColor[chrom[l]]

PlotChromListColor[list_] := Show[ChromDiagramColor, Map[cPointColor, list],
  DisplayFunction->$DisplayFunction]


(* Plot RGB gamuts, given a transformation from XYZ coordinates to RGB
   coordinates. Note that when displaying gamuts for transforms other than
   XyzToRgbCRT, the acutally displayed colors are not correct since they
   are limited to what the monitor can display (of which CRT is an
   approximation). The extent of the gamut is ok, however.
   The gamuts are displayed in the Cie chromaticity space (xy coordinates),
   so colors are normalized for maximum intensity.
   To plot CRT and LMS gamuts respectively, use these expressions:
   Show[Txr2ChromGamut[transforms`XyzToRgbCRT, 0.85, 64, "CRT"],
     DisplayFunction->$DisplayFunction]
   Show[Txr2ChromGamut[transforms`XyzToLmsVW, 0.85, 64, "LMS"],
     DisplayFunction->$DisplayFunction]
*)

(* The function below uses Plot3D only because that allows colors of
   plotted points to be specified, and Plot doesn't. It plots a plane at z
   coordinate 0, and x,y determined by chromaticity coordinates. The range
   for the regular Cie chromaticity diagram is 0.85 *)

Txr2ChromGamut[txr_,range_,points_,label_,bg_:{.5,.5,.5},CieFns_:XyzList] :=
 (XyzList = CieFns;
  Plot3D[{0, Chrom2RGBColorClip[{x,y}, txr, bg, CieFns]},
    {x,0,range}, {y,0,range}, ViewPoint->{0,0,25},
    Background->ReplacePart[bg, RGBColor, 0],
    PlotPoints->{points,points}, Lighting->False, Mesh->False,
    PlotLabel->label,
    Axes->{True,True,False},
    DisplayFunction->Identity]
  )

(* The following functions compute RGB values from chromaticity coordinates
   and an XYZ->RGB transform (txr). The idea is that since chrom coords are
   related to XYZ coordinates by a constant 1/(X+Y+Z), we can use the chrom
   coords in stead of XYZ coords as input to the transform, and then scale
   the result so that the maximum component is 1. Since chrom coords carry
   no luminance info anyway, that is fine. The result should be gamma
   corrected for display of course, but Mathematica takes care of that. *)

Chrom2RGBColor[{x_,y_}, txr_, bg_:{.5,.5,.5}, CieFns_:XyzList] := (
  XyzList = CieFns;
  RGBColor[#[[1]],#[[2]],#[[3]]]& [
    (1/Max[#])#&[
      Map[If[#<0,0,#]&, txr . {x, y, 1-x-y}]]]
```

```
  )

(* Same, but turns any point with a negative coordinate into bg gray.
   Displaying points computed in this way on the same gray bg makes only
   the valid points stand out on a gray bg. *)

Chrom2RGBColorClip[{x_,y_}, txr_, bg_:{.5,.5,.5}, CieFns_:XyzList] := (
  XyzList = CieFns;
  RGBColor[#[[1]],#[[2]],#[[3]]]& [
    If[Min[#]<0, bg, #]& [
      (1/Max[#])#&[
        txr . {x, y, 1-x-y}]]]
  )

(* These are some examples of gamuts in chromaticity coordinates *)

GamutNTSC := Txr2ChromGamut[transforms`XyzToRgbNTSC, 64, "NTSC"]
GamutCRT  := Txr2ChromGamut[transforms`XyzToRgbCRT, 64, "CRT"]
GamutLMS  := Txr2ChromGamut[transforms`XyzToLmsVW, 64, "LMS"]

(* Plot Opponent gamuts, given a transformation from opponent coordinates
   to RGB coordinates. The same restriction to displayed colors applies as
   above. Opponent gamuts are displayed at constant brightness planes
   through the opponent color space, with or without intensity
   normalization. *)

(* normalized *)

Tor2GamutN[tor_, brightness_, hrange_, vrange_, points_, label_] :=
  Plot3D[{brightness, Tor2RGBColorClipN[{gr, by, brightness}, tor]},
   {gr, -hrange, hrange}, {by, -hrange, hrange},
   PlotPoints -> {points, points},
   PlotRange -> {{-hrange, hrange}, {-hrange, hrange}, {0, vrange}},
   Background -> GrayLevel[0.5], Lighting -> False, Mesh -> False,
   PlotLabel -> label, BoxRatios -> {1, 1, 2}, SphericalRegion -> True,
   DisplayFunction -> Identity]

(* The function for converting to rgb values has to limit display to
   possible points only; since the rgb transforms are scaled for a maximum
   value of one, any point which has an RGB component that is negative or
   exceeds 1 is not possible. Whatever transform is used, e.g. H81ToLms (human
   cone sensitivities), results are displayed on a monitor using its own
   gamut of course. Using Lms, good ranges are [0,1.8] for brightness and
   [-3.5,3.5] for gr/by. *)

Tor2RGBColorClipN[{gr_,by_,wb_}, tor_] :=
  RGBColor[#[[1]],#[[2]],#[[3]]]& [
    Which[
      Min[#]<0, {.5,.5,.5},
      Max[#]>1, {.5,.5,.5},
      True, (1/Max[#])# ]& [
    tor . {gr,by,wb} ]]
```

```
(* unnormalized *)

Tor2Gamut[tor_, brightness_, hrange_, vrange_, points_, label_] :=
  Plot3D[{brightness, Tor2RGBColorClip[{gr, by, brightness}, tor]},
   {gr, -hrange, hrange}, {by, -hrange, hrange},
   PlotPoints -> {points, points},
   PlotRange -> {{-hrange, hrange}, {-hrange, hrange}, {0, vrange}},
   Background -> GrayLevel[0.5], Lighting -> False, Mesh -> False,
   PlotLabel -> label, BoxRatios -> {1, 1, 2}, SphericalRegion -> True,
   DisplayFunction -> Identity]

Tor2RGBColorClip[{gr_,by_,wb_}, tor_] :=
  RGBColor[#[[1]],#[[2]],#[[3]]]& [
    Which[
      Min[#]<0, {.5,.5,.5},
      Max[#]>1, {.5,.5,.5},
      True, # ]& [
    tor . {gr,by,wb} ]]

(* use these expressions to generate an animated plot of constant-brightness
   planes through H81 and WWAW opponent space, mapped into RgbCRT space.
   Note that using Lms results in different shapes. *)

AnimH81CRT :=
  ShowAnimation[ Table[ Tor2Gamut[H81ToRgbCRT, b, 1, 2.5, 16, ""],
    {b,0,2.3,.1}]]

AnimH81Lms :=
  ShowAnimation[ Table[ Tor2Gamut[H81ToLmsVW, b, 3, 2, 16, ""],
    {b,0,1.84,.08}]]

AnimWWAWCRT :=
  ShowAnimation[ Table[ Tor2Gamut[WWAWToRgbCRT, b, 1, 1.5, 16, ""],
    {b,0,1.38,.06}]]

End[]       (* private *)
EndPackage[] (* package *)
```

Appendix C

Getting the software

The software for the application described in Chapter 8, both the select/display X Windows client and the Mathematica code, is available by anonymous ftp from

`ftp.cs.buffalo.edu (128.205.38.1), directory /ftp/pub/colornaming`

There is a README file in the directory with further instructions. For WWW[1] clients (NCSA's Mosaic, or your favorite client) the URL is

`http://www.cs.buffalo.edu/pub/colornaming`

All software is provided as is, without any expressed or implied warranty. It is guaranteed not to be useful for any commercial application, in its current form. If you have problems with it, feel free to contact me:

`lammens@cs.buffalo.edu (until 5/94)`

`lammens@arts.sssup.it (from 6/94)`

Enjoy!

[1] *Really* the greatest thing since sliced bread!

Bibliography

[Agre 1988] Agre, Philip (1988), "The dynamic structure of everyday life", Technical Report 1085, MIT Artificial Intelligence Laboratory, MIT.

[Agre & Chapman 1987] Agre, Philip E. and Chapman, David (1987), "Pengi: An implementation of a theory of activity", In *Proceedings of AAAI-87, Seattle WA*, 268–272.

[Albus 1991] Albus, James (1991), "Outline for a theory of intelligence", In *IEEE Transactions on Systems, Man, and Cybernetics, Vol. 21, No. 3*, 473–509.

[Albus et al. 1981] Albus, James; Barbera, Anthony; and Nagel, Roger (1981), "Theory and practice of hierarchical control", In *23rd International IEEE Computer Society Conference*, 18–38.

[Aleksandrov 1956] Aleksandrov, A.D. (1956), "A general view of mathematics", In Aleksandrov, A.D.; Kolmogorov, A.N.; and Lavrent'ev, M.A. (eds.), *Mathematics*, volume 1. (Dorset Press, New York), chapter I, 1–64, 1990 edition.

[Anderson 1983] Anderson, J. R. (1983), *The Architecture of Cognition*, (Cambridge: Harvard University Press).

[Anderson et al. 1991] Anderson, Scott; Hart, David; and Cohen, Paul (1991), "Two ways to act", In *ACM SIGART Bulletin.* (ACM publications), 20–24.

[Angell 1993] Angell, Ian O. (1993), "Intelligence: Logical or biological", *CACM*, 36(7).

[Ballard & Brown 1982] Ballard, Dana H. and Brown, Christopher M. (1982), *Computer Vision*, (Prentice-Hall, Englewood Cliffs NJ).

[Berlin & Kay 1969] Berlin, Brent and Kay, Paul (1969), *Basic Color Terms: Their Universality and Evolution*, (University of California Press, Berkeley CA), 1991 edition.

[Birren 1969a] Birren, Faber (1969a), *Munsell: A grammar of color*, (Van Nostrand Reinhold, New York).

[Birren 1969b] Birren, Faber (1969b), *Ostwald: The color primer.*, (Van Nostrand Reinhold, New York).

[Boolos & Jeffrey 1974] Boolos, George and Jeffrey, Richard (1974), *Computability and Logic*, (Cambridge University Press, Cambridge (England)).

[Boynton 1979] Boynton, Robert M. (1979), *Human Color Vision*, (Holt, Rinehart and Winston).

[Boynton 1990] Boynton, Robert M. (1990), "Human color perception", In Leibovic, K. N., (ed.), *Science of Vision.* (Springer, New York), chapter 8, 211–253.

[Boynton & Olson 1987] Boynton, Robert M. and Olson, Conrad X. (1987), "Locating basic colors in the OSA space", *Color Research and Application*, 12(2):94–105.

[Brooks 1985] Brooks, Rodney (1985), "A robust layered control system for a mobile robot", Technical Report 864, MIT AI Labs, MIT.

[Brooks 1987] Brooks, Rodney (1987), "Planning is just a way of avoiding figuring out what to do next", Technical Report 303, MIT AI Labs.

[Brooks 1990] Brooks, Rodney A. (1990), "Elephants don't play chess", *Robotics and Autonomous Systems*, 6:3–15.

[Brown 1982] Brown, Chrisopher M. (1982), "Color vision and computer vision", Technical Report 108, University of Rochester, Computer Science department.

[Brown 1994] Brown, Christopher M. (1994), "Vision, learning, and development", Technical Report 492, University of Rochester, Computer Science department.

[Cairo 1977] Cairo, James Edward (1977), *The Neurophysiological Basis of Basic Color Terms*, PhD thesis, State University of New York at Binghamton.

[Card et al. 1983] Card, S.K.; Moran, T.P.; and Newell, A. (1983), *The Psychology of Human-Computer Interaction*, (Erlbaum, Hillsdale, N.J.).

[Chapman 1990] Chapman, David (1990), "Vision, instruction, and action", Technical Report 1204, MIT Artificial Intelligence Laboratory, MIT.

[Connell 1992] Connell, Jonathan (1992), "SSS: A hybrid architecture applied to robot navigation", In *IEEE Conference on Robotics and Automation*, 2719–2724.

[Cowan 1983] Cowan, William B. (1983), "An inexpensive scheme for calibration of a colour monitor in terms of CIE standard coordinates", *Computer Graphics*, 17(3):315–321.

[Crane & Piantanida 1983] Crane, Hewitt D. and Piantanida, Thomas P. (1983), "On seeing reddish green and yellowish blue", *Science*, 221:1078–1080.

[Culbertson 1963] Culbertson, James (1963), *The Minds of Robots*, (U. of Illinois Press).

[Danger 1987] Danger, E. P. (1987), *The colour handbook. How to use colour in commerce and industry.*, (Gower Technical Press, Aldershot (UK)).

[Davidoff & Concar 1993] Davidoff, Jules and Concar, David (1993), "Brain cells made for seeing", *New Scientist*, 138:32–36.

[Davis & Weyuker 1983] Davis, Martin D. and Weyuker, Elaine J. (1983), *Computability, complexity, and languages: fundamentals of theoretical computer science*, (Academic Press, New York).

[De Valois et al. 1966] De Valois, Russell L.; Abramov, Israel; and Jacobs, Gerald H. (1966), "Analysis of response patterns of LGN cells", *Journal of the Optical Society of America*, 56(7):966–977.

[De Valois & De Valois 1975] De Valois, Russell L. and De Valois, Karen K. (1975), "Neural coding of color", In Carterette, Edward C. and Friedman, Morton P. (eds.), *Handbook of Perception, Vol. V: Seeing.* (Academic Press, New York), chapter 5, 117–166.

[De Valois & Jacobs 1968] De Valois, Russell L. and Jacobs, Gerald H. (1968), "Primate color vision", *Science*, 162:533–540.

[Derrington et al. 1984] Derrington, A. M.; Krauskopf, J.; and Lennie, P. (1984), "Chromatic mechanisms in lateral geniculate nucleus of macaque", *Journal of Physiology*, 357:241–265.

[Dow 1990] Dow, Bruce M. (1990), "Nested maps in macaque monkey visual cortex", In Leibovic, K.N., (ed.), *Science of Vision.* (Springer, New York), chapter 4, 84–124.

[G. Edelman 1989] Edelman, Gerald M. (1989), *Neural Darwinism : the theory of neuronal group selection*, (New York: Basic Books).

[G. Edelman 1992] Edelman, Gerald M. (1992a), *Bright air, brilliant fire. On the matter of the mind*, (Basic Books, New York).

[S. Edelman 1992] Edelman, Shimon (1992b), "Visual perception", In Shapiro, Stuart C., (ed.), *Encyclopedia of Artificial Intelligence.* (Wiley-Interscience, New York), 1655–1664.

[Firby 1987] Firby, R. James (1987), "An investigation into reactive planning in complex domains", In *Proceedings of AAAI-87*, 202–206.

[Fodor 1983] Fodor, Jerry (1983), *The Modularity of Mind*, (MIT Press).

[Foley & Van Dam 1982] Foley, James D. and Dam, Andries Van (1982), *Fundamentals of Interactive Computer Graphics*, Addison-Wesley Systems Programming Series. (Addison-Wesley, Reading MA), 1984 edition.

[Hill 1990] Francis S. Hill, Jr. (1990), *Computer Graphics*, (Macmillan, New York).

[Frumkes 1990] Frumkes, Thomas E. (1990), "Classical and modern psychophysical studies of dark and light adaptation and their relationship to underlying retinal function", In Leibovic, K. N., (ed.), *Science of Vision.* (Springer, New York), chapter 7, 172–210.

[Gat 1991] Gat, Erann (1991), "Reliable goal-directed reactive control of autonomous mobile robot", Technical report, Dept. of Computer Science, Virginia Polytechnic Institute and State University.

[GSPC-ACM 1979] Graphics standards planning committee of the ACM (1979), "Status report", *Computer Graphics*, 13(3).

[Harnad 1990] Harnad, Stevan (1990), "The symbol grounding problem", *Physica D*, 42(1-3):335–346.

[Harnad 1992] Harnad, Stevan (1992), "Electronic symposium on computation, cognition and the symbol grounding problem", E-mail symposium (ftp archive at princeton.edu:/pub/harnad/sg.comp.arch*).

[Harnad et al. 1991] Harnad, S.; Hanson, S.J.; and Lubin, J. (1991), "Categorial perception and the evolution of supervised learning in neural nets", Presented at 1991 AAAI Symposium on Symbol Grounding: Problem and Practice.

[Hausser 1989] Hausser, Roland (1989), *Computation of Language: An Essay on Syntax, Semantics and Pragmatics in Natural Man-Machine Communication*, (New York, NY: Springer-Verlag).

[Heisenberg 1988] Heisenberg, Werner (1988), "The uncertainty principle", In Newman, James R., (ed.), *The World of Mathematics*. (Tempus Books, Redmond WA), 1029–1033.

[Hering 1878] Hering, E. (1878), *Zur Lehre vom Lichtsinne*, (C. Gerold's Sohn, Vienna).

[Hexmoor 1989] Hexmoor, Henry (1989), "An architecture for reactive sensor-based robots", In *NASA Goddard conference on AI*, Greenbelt, MD.

[Hexmoor et al. 1993a] Hexmoor, Henry; Caicedo, Guido; Bidwell, Frank; and Shapiro, Stuart (1993a), "Air battle simulation: An agent with conscious and unconscious layers", In *University at Buffalo Graduate Conference in Computer Science 93 (TR93-14)*. Dept. of Computer Science, SUNY at Buffalo, New York.

[Hexmoor et al. 1993b] Hexmoor, Henry; Lammens, Johan; Caicedo, Guido; and Shapiro, Stuart C. (1993b), "Behavior based AI, cognitive processes, and emergent behaviors in autonomous agents", In *Proceedings of the International Conference of AI in Engineering (Toulouse, France, to appear)*.

[Hexmoor et al. 1992] Hexmoor, Henry; Lammens, Johan; and Shapiro, Stuart (1992), "An autonomous agent architecture for integrating perception and acting with grounded, embodied symbolic reasoning", Technical Report CS-92-21, Dept. of Computer Science, SUNY at Buffalo, NY.

[Hexmoor et al. 1993c] Hexmoor, Henry; Lammens, Johan; and Shapiro, Stuart (1993c), "Embodiment in GLAIR: A grounded layered architecture with integrated reasoning for autonomous agents", In Dankel, Douglas D., (ed.), *Proceedings of the 6th Florida AI Research Symposium*, 325–329. Florida AI Research Society.

[Hexmoor & Nute 1992] Hexmoor, Henry and Nute, Donald (1992), "Methods for deciding *what to do next* and learning", Technical Report AI-1992-01, AI Programs, The University of Georgia, Athens, Georgia, Also available from SUNY at Buffalo, CS Department TR-92-23.

[Horn 1986] Horn, Berthold K. (1986), *Robot Vision*, (MIT Press, Cambridge MA).

[Hubel & Livingstone 1990] Hubel, David H. and Livingstone, Margaret S. (1990), "Color and contrast sensitivity in the lateral geniculate body and primary visual cortex of the macaque monkey", *Journal of Neuroscience*, 10(7):2223–2237.

[Hurlbert 1991] Hurlbert, Anya (1991), "Deciphering the colour code", *Nature*, 349:191–193.

[Hurvich 1981] Hurvich, Leo M. (1981), *Color Vision*, (Sinauer Associates, Sunderland MA).

[Hurvich & Jameson 1955] Hurvich, Leo M. and Jameson, Dorothea (1955), "Some quantitative aspects of an opponent-colors theory: II. brightness, saturation, and hue in normal and dichromatic vision.", *Journal of the Optical Sociey of America*, 45(8):602–616.

[Hurvich & Jameson 1956] Hurvich, Leo M. and Jameson, Dorothea (1956), "Some quantitative aspects of an opponent-colors theory: IV. a psychological color specification system.", *Journal of the Optical Sociey of America*, 46(6):416–421.

[Hurvich & Jameson 1957] Hurvich, Leo M. and Jameson, Dorothea (1957), "An opponent-process theory of color vision", *Psychological Review*, 64(6):384–404.

[Jameson & Hurvich 1955] Jameson, Dorothea and Hurvich, Leo M. (1955), "Some quantitative aspects of an opponent-colors theory: I. chromatic responses and spectral saturation.", *Journal of the Optical Sociey of America*, 45(7):546–552.

[Jameson & Hurvich 1956] Jameson, Dorothea and Hurvich, Leo M. (1956), "Some quantitative aspects of an opponent-colors theory: III. changes in brightness, saturation, and hue with chromatic adaptation.", *Journal of the Optical Sociey of America*, 46(6):405–415.

[Jameson & Hurvich 1968] Jameson, Dorothea and Hurvich, Leo M. (1968), "Opponent-response functions related to measured cone photopigments.", *Journal of the Optical Sociey of America*, 58:429–430.

[Kaelbling 1988] Kaelbling, Leslie (1988), "Goals as parallel program specifications", In *Proceedings of AAAI-88*. Morgan Kaufman.

[Kaelbling & Rosenschein 1990] Kaelbling, Leslie and Rosenschein, Stanley (1990), "Action and planning in embedded agents", In Maes, Pattie, (ed.), *Designing Autonomous Agents*. (MIT Press), 35–48.

[Kay et al. 1991] Kay, Paul; Berlin, Brent; and Merrifield, William R. (1991), "Biocultural implications of systems of color naming", *Journal of Linguistic Anthropology*, 1(1):12–25.

[Kay & Kempton 1984] Kay, Paul and Kempton, Willett (1984), "What is the Sapir-Whorf Hypothesis?", *American Anthropologist*, 86:65–79.

[Kay & McDaniel 1978] Kay, Paul and McDaniel, Chad K. (1978), "The linguistic significance of the meaning of Basic Color Terms", *Language*, 54(3):610–646.

[Kosko 1992] Kosko, Bart (1992), *Neural Networks and Fuzzy Systems*, (Prentice Hall, Englewood Cliffs NJ).

[Laird et al. 1991] Laird, J.; Huka, M.; Yager, E.; and Tucker, C. (1991), "Robo-soar: An integration of external interaction, planning, and learning, using soar", In *Robotics and Autonomous Systems*.

[Laird et al. 1987] Laird, J. E.; Newell, A.; and Rosenbloom, P. S. (1987), "Soar: An architecture for general intelligence", *Artificial Intelligence*, 33:1–64.

[Lakoff 1987] Lakoff, George (1987), *Women, Fire, and Dangerous Things: What Categories Reveal about the Mind*, (Chicago, IL: University of Chicago Press).

[Lammens 1992] Lammens, Johan M. (1992), "A computational model of color perception and color naming: a case study of symbol grounding for natural language semantics", Dissertation proposal, SUNY/Buffalo CS department.

[Lammens et al. 1994] Lammens, Johan M.; Hexmoor, Henry H.; and Shapiro, Stuart C. (1994), "Of elephants and men", In *Proceedings of the NATO ASI on Intelligent Autonomous Agents (Trento, Italy)*. Springer Verlag, (To appear. Preprint available as CS TR-93-13, SUNY/Buffalo).

[Langley et al. 1991] Langley, Pat; McKusick, Kathleen; and Allen, John (1991), "A design for the ICARUS architecture", In *ACM SIGART Bulletin*. (ACM publications), 104–109.

[Leibovic 1990a] Leibovic, K.N. (1990a), "Vertebrate photoreceptors", In Leibovic, K.N., (ed.), *Science of Vision*. (Springer, New York), chapter 2, 16–52.

[Leibovic 1990b] Leibovic, K. N. (1990b), "Vertebrate photoreceptors", In Leibovic, K. N., (ed.), *Science of Vision*. (Springer, New York), chapter 2, 16–52.

[Lenat 1990] Lenat, Douglas B. (1990), *Building large knowledge-based systems: representation and inference in the Cyc project*, (Addison-Wesley, Reading MA).

[Levesque 1988] Levesque, Hector J. (1988), "Logic and the complexity of reasoning", *Journal of Philosophical Logic*, 17:355–389.

[Locke 1690] Locke, John (1690), *An Essay Concerning Human Understanding*, (Dover Publications, New York (Dover ed. 1959)).

[Luce 1954] Luce, Arthur Aston (1954), *Sense Without Matter or Direct Perception*, (Nelson, Edinburgh).

[Maes 1991] Maes, Pattie (1991), "Action selection", In *Proceedings of the Cognitive Science Society Conference*.

[Maloney 1993] Maloney, Laurence T. (1993), "Color constancy and color perception: The linear-models framework", In Meyer, David E. and Kornblum, Sylvan (eds.), *Attention and Performance XIV*. (MIT Press), chapter 3, 60–78.

[Maloney & Wandell 1986] Maloney, L. T. and Wandell, B. A. (1986), "Color constancy: a method for recording surface spectral reflectance", *Journal of the Optical Sociey of America*, 29–33.

[Manin 1977] Manin, Yu. I. (1977), *A Course in Mathematical Logic*, (Springer Verlag, New York).

[Marr 1982] Marr, David (1982), *Vision*, (San Francisco, CA: W. H. Freeman).

[Maturana & Varela 1987] Maturana, Humberto R. and Varela, Francisco J. (1987), *The tree of knowledge*, (Shambhala, Boston MA).

[McClelland et al. 1986] McClelland, J. L.; Rumelhart, D. E.; and Hinton, G. E. (1986), "The appeal of parallel distributed processing", In Rumelhart, David; McClelland, James; and the PDP Research Group (eds.), *Parallel Distributed Processing.* (MIT Press, Cambridge MA), chapter 1, 3–44.

[McDermott 1991] McDermott, Drew (1991), "Robot planning", Technical Report CS-861, Yale University.

[McIlwain & Dean 1956] McIlwain, Knox and Dean, Charles E. (1956), *Principles of Color Television*, (John Wiley & Sons, New York).

[Meyer & Greenberg 1980] Meyer, Gary W. and Greenberg, Donald P. (1980), "Perceptual color spaces for computer graphics", *Computer Graphics*, 14:254–261.

[Munsell 1946] Munsell, A.H. (1946), *A color notation*, (Munsell Color Co., Baltimore MD).

[Naiman 1985] Naiman, Avi (1985), "Color spaces and color contrast", *Visual Computer*, 1:194–201.

[Newell 1979] Newell, A. (1979), "Physical symbol systems", Lecture at the La Jolla Conference on Cognitive Science.

[Newhall et al. 1943] Newhall, Sidney M.; Nickerson, Dorothy; and Judd, Deane B. (1943), "Final report of the O.S.A. subcommittee on the spacing of the munsell colors", *Journal of the Optical Sociey of America*, 33(7):385–418.

[Novak & Shafer 1992] Novak, Carol L. and Shafer, Steven A. (1992), "Color vision", In Shapiro, Stuart C., (ed.), *Encyclopedia of Artificial Intelligence.* (Wiley-Interscience, New York), 192–202.

[Patterson 1986] Patterson, Roy D. (1986), "Spiral detection of periodicity and the spiral form of musical scales", *Psychology of Music*, 14(1):44–61.

[Payton 1986] Payton, David (1986), "An architecture for reflexive autonomous vehicle control", In *Proceedings of Robotics Automation*, 1838–1845. IEEE.

[Pollock 1989] Pollock, John (1989), *How to Build a Person*, (MIT Press).

[Pollock 1992] Pollock, John (1992), "New foundations for practical reasoning", In *Minds and Machines.*

[Putnam 1981] Putnam, Hilary (1981), *Reason, Truth, and History*, (Cambridge University Press, Cambridge).

[Pylyshyn 1981] Pylyshyn, Zenon W. (1981), "Psychological explanations and knowledge-dependent processes", *Cognition*, 10(1-3):267–274.

[Rapaport 1988] Rapaport, William J. (1988), "Syntactic semantics: Foundations of computational natural-language understanding", In Fetzer, James H., (ed.), *Aspects of Artificial Intelligence.* (Kluwer Academic, New York), 81–131.

[Rapaport 1992] Rapaport, W. J. (1992), "Logic", In Shapiro, Stuart C., (ed.), *Encyclopedia of Artificial Intelligence.* (Wiley–Interscience, New York), 851–853.

[Regan & Beverly 1978] Regan, D. and Beverly, K.I. (1978), "Looming detectors in the human visual pathways", In *Vision Research 18*, 209–212.

[Robertson & O'Callaghan 1986] Robertson, Philip K. and O'Callaghan, J.F. (1986), "The generation of color sequences for univariate and bivariate mapping", *IEEE Computer Graphics and Applications*, 6(2):24–32.

[Rogers 1985] Rogers, Dvaid F. (1985), *Procedural elements for computer graphics*, (McGraw-Hill, New York).

[Ronchi 1957] Ronchi, Vasco (1957), *Optics: The Science of Vision*, (Dover Publications, New York (Dover ed. 1991)).

[Rosch 1978] Rosch, Eleanor (1978), "Principles of categorization", In *Cognition and Categorization.* (Hillsdale, NJ: Lawrence Erlbaum Associates).

[Rosenblatt & Payton 1989] Rosenblatt, J. Kenneth and Payton, David (1989), "A fine-grained alternative to the subsumption architecture for mobile robot control", In *Proceedings of the International Joint Conference on Neural Networks.*

[Rumelhart et al. 1986] Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. (1986), "Learning internal representations by error propagation", In Rumelhart, David; McClelland, James; and the PDP Research Group (eds.), *Parallel Distributed Processing.* (MIT Press, Cambridge MA), chapter 1, 319–362.

[Russell 1991] Russell, Stuart (1991), "An architecture for bounded rationality", In *ACM SIGART Bulletin.* (ACM publications), 146–150.

[Scheifler & Gettys 1992] Scheifler, Robert W. and Gettys, James (1992), *X Window System (third ed.)*, (Digital Press).

[Schoppers 1987] Schoppers, Marcel J. (1987), "Universal plans for unpredictable environments", In *Proceedings 10th IJCAI*, 1039–1046.

[Schrödinger 1988] Schrödinger, Erwin (1988), "Causality and wave mechanics", In Newman, James R., (ed.), *The World of Mathematics.* (Tempus Books, Redmond WA), 1035–1047.

[Searle 1980] Searle, John R. (1980), "Minds, brains, and programs", *Behavioral and Brain Sciences*, 3:417–424.

[Shapiro 1990] Shapiro, Stuart C. (1990), "Cables, paths, and 'subconscious' reasoning in propositional semantic networks", In *Principles of Semantic Networks.* (Morgan Kaufman).

[Shapiro & Rapaport 1987] Shapiro, S. C. and Rapaport, W. J. (1987), "SNePS considered as a fully intensional propositional semantic network", In Cercone, N. and McGalla, G.

(eds.), *The knowledge frontier: essays in the representation of knowledge.* (Springer, New York), 262–315.

[Shen 1989] Shen, Wei-Min (1989), *Learning from the Environment Based on Actions and Percepts*, PhD thesis, Carnegie Mellon University.

[Shepard 1987] Shepard, Roger N. (1987), "Toward a universal law of generalization for psychological science", *Science*, 237:1317–1323.

[Simmons 1990] Simmons, Reid (1990), "An architecture for coordinating planning, sensing, and action", In *Proceedings the DARPA workshop*, 292–297.

[Slaughter 1990] Slaughter, Malcolm (1990), "The vertebrate retina", In Leibovic, K. N., (ed.), *Science of Vision.* (Springer, New York), chapter 3, 53–83.

[A. Smith 1978] Smith, Alvy Ray (1978), "Color gamut transform pairs", *Computer Graphics*, 12(3):12–25.

[B. Smith 1985] Smith, Brian C. (1985), "Prologue to "reflection and semantics in a procedural language"", In Brachmann, Ronald J. and Levesque, Hector J. (eds.), *Readings in Knowledge Representation.* (Morgan Kaufmann, San Mateo CA), 31–40.

[Suchman 1988] Suchman, Lucy A. (1988), *Plans and Situated Actions: The Problem of Human Machine Communication*, (Cambridge University Press).

[Turkowski 1986] Turkowski, Kenneth (1986), "Anti-aliasing in topological color spaces", *Computer Graphics*, 20(4):307–314.

[Valberg et al. 1986] Valberg, A.; Seim, T.; Lee, B. B.; and Tryti, J. (1986), "Reconstruction of equidistant color space from responses of visual neurones of macaques.", *Journal of the Optical Sociey of America*, 3(10):1726–1734.

[Walters 1987] Walters, Deborah K. (1987), "Selection of image primitives for general-purpose visual processing", *Computer Vision, Graphics, and Image Processing*, 37:261–298.

[Winston 1975] Winston, Patrick Henry (1985 (orig. 1975)), "Learning structural descriptions from examples", In Brachman, Ronald J. and Levesque, Hector J. (eds.), *Readings in Knowledge Representation.* (Morgan Kaufmann, San Mateo CA), 141–168.

[Wittgenstein 1953] Wittgenstein, Ludwig (1953), *Philosophical Investigations*, (Oxford: Blackwell).

[Wyszecki & Stiles 1982] Wyszecki, G. and Stiles, W. S. (1982), *Color Science*, (Wiley, New York).

[Zadeh 1971] Zadeh, L. A. (1971), "Quantitative fuzzy semantics", *Information Sciences*, 3:159–176.

[Zeidenberg 1990] Zeidenberg, Matthew (1990), *Neural Networks in Artificial Intelligence*, (Ellis Horwood, New York).

[Zeki 1993] Zeki, Semir (1993), *A Vision of the Brain*, (Blackwell Scientific, Oxford).