AN ARCHITECTURE FOR EXPLOITING QUALITATIVE, SCENE-SPECIFIC CONTEXT IN HIGH LEVEL COMPUTER VISION

by

Rajiv Chopra

A Dissertation submitted to the

Faculty of the Graduate School of the State University

of New York at Buffalo in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

June 1997

Acknowledgments

I wish to thank:

1. Dr. Rohini Srihari, my advisor, for her constant guidance and support.
2. The other members of my committee : Prof. Sargur Srihari and Prof. Deborah Walters for the interest they have shown and for their advice.
3. Dr. Jeffrey Mark Siskind for agreeing to read this dissertation as the external reader at short notice.
4. The secretaries Eloise Benzel, Eugenia Smith, Heike Berlind, Rita Hamilton, Sally Elder, Elenora Heffner and Gloria Koontz for their help with administrative matters.
5. All members of CEDAR for providing an interesting research environment.
6. To all my friends for their encouragement, companionship and understanding.

I dedicate this dissertation to my parents and my brothers for their love and faith in me.

Contents

List of Figures

Abstract

In this dissertation we present an architecture to incorporate collateral information in the high level computer vision task of object location. The declarative specification of the scene hypothesis and the domain independent control algorithm that exploits spatial information to drive the vision process, are two major contributions of this dissertation. This work has been theoretically grounded in constraint satisfaction algorithms. Apart from the application of standard CSP algorithms, several new techniques in constraint satisfaction have been developed. We provide an algorithm that performs consistency filtering for certain types of non-binary constraints. We also presente an innovative application of interval arithmetic in constraint satisfaction. Though interval constraint satisfaction is an extensive research area, we believe this to be the first attempt at using interval arithmetic to reduce the domain generation cost of finite domain CSPs. A new algorithm has been proposed here for the above. Illustration with a simple example, analysis and implementation of the algorithm have also been detailed here.

We have demonstrated the architecture in three implemented vision systems and feel that it is simple yet powerful for application in several domains.

Chapter 1

Introduction

On the back cover of the textbook *Computer Vision* by Dana Ballard and Christopher Brown are four questions, two of which are : "At what stage must domain dependent, prior knowledge about the world be incorporated into the understanding process?" and "How are world models and conceptual knowledge represented and used?" We examine these questions, their traditional solutions and propose better alternatives. An object's setting can strongly influence how that object is recognized, what it is identified as, and if it can be recognized at all. Psychological studies have shown that people cannot understand a scene in the absence of sufficient context, yet when such contextual information is present, recognition is unambiguous [Firschein and Fischler, 1987]. Objects exhibit a multitude of appearances under different imaging conditions, and many different objects may have the same image appearance. Their correct interpretation must be decided entirely by context. Such studies lead us to believe that computer vision systems will be unable to recognize and identify objects competently using only local information. Knowledge about the world can be of various types. A possible categorization of knowledge is the specificity of knowledge. We differentiate between *world knowledge*, which is universally applicable, *domain specific knowledge*, which is applicable in a restricted world, and *scene specific knowledge*, which is knowledge about a particular scene. Contextual information is not necessary for everything. Many objects have simple distinctive features which can be instantly recognized even when all contextual cues are removed or even violated. For example, an image of an office scene with a telephone on the floor is unusual but not impossible to recognize. There is a tradeoff between the specificity of contextual information and its applicability. The domain specific knowledge that telephones are normally on desks and not on floors is not applicable in this scene. Contextual information has been used by various vision systems. For example, Hawkeye used the location of a pier (obtained from a stored map of a port facility) to constrain the regions where a ship might be found [Barrow and Tenenbaum, 1977]. SPAM used a map and domain specific knowledge to find features in aerial imagery of airports [McKeown *et al.*, 1985]. Condor uses an architecture which represents context explicitly and uses it to guide all phases of computation to recognize objects in a natural outdoor scene [Strat, 1992]. However, the emphasis in the above systems was on object *recognition* based on *domain specific* knowledge. We investigate object *recognition and identification* based on *domain and scene specific* knowledge. Object recognition involves assigning a category or class to an object, as opposed to object identification which involves assigning a unique identity to an object apart from a category or class.

The motivation for this work comes partly from multimedia databases where one has access to images with related (collateral) information. Co-referencing the images and their components with the collateral information is essential for intelligent content based retrieval and browsing.

1.1 Research Contributions

We describe an architecture for context based vision that exploits *qualitative, scene-specific* knowledge. At the core is a domain independent control module that guides the interpretation process. We provide a modeling framework for encoding domain specific knowledge, and an interface for declaratively specifying scene-specific knowledge. We describe new algorithms in constraint satisfaction necessary for collateral based vision. We show how such an architecture can be realized by describing three implemented systems, PICTION, Show&Tell and SpatialFEVAHR.

The research contributions of this dissertation can be enumerated as:

1. High-level Vision:
    - Presented a novel and simple architecture for exploiting contextual information.
    - Demonstrated its utility and extensibility in three vision applications.
2. Constraint Satisfaction:
    - Developed a new consistency technique for non-binary constraints
    - Coupled discrete constraint satisfaction techniques with interval reasoning in an innovative manner to increase efficiency.
3. Spatial Reasoning:
    - Presented an inference mechanism to reason with spatial expressions in the presence of partial information
    - Demonstrated an innovative application of spatial reasoning in computer vision.
4. Multimedia Information Retrieval:

- A knowledge based technique to provide detailed cross-references between related entities in different modalities.
- A domain specific semantic indexing technique.

1.2 Organisation of the Dissertation

This dissertation has been organized as follows.

Chapter 2 discusses related research. Separate sections cover the areas of knowledge-based computer vision, constraint satisfaction and spatial reasoning that are relevant to this disseration.

Chapter 3 describes the architecture of collateral based vision. Examples from the application domain of **Show&Tell** are presented to illustrate the concepts.

Chapter 4 describes application-independent extensions proposed to constraint satisfaction techniques for use in context based vision.

Chapter 5 describes three implemented applications of collateral based vision in detail.

Chapter 6 discusses some experiments that were performed to study the basic architecture in a domain independent setting. Evaluation of components of the implemented systems that are relevant to this dissertation is also performed here.

Chapter 7 summarises the research presented in this dissertation and discusses future work.

Chapter 2

Background

Separate sections briefly introduce the areas of knowledge-based computer vision, constraint satisfaction and spatial reasoning that influence this work. We start with the general area of knowledge-based computer vision where we focus on control structures and types of knowledge that have been used in the past. Since our architecture relies on constraint satisfaction techniques, we then discuss related research in using constraint satisfaction (CSP) in computer vision. Since we have proposed some extensions to CSP techniques, we discuss research in constraint satisfaction as well. A large portion of the contextual information in each application domain we explore is spatial in nature. Therefore, the ability to reason (in a limited capacity) with and represent such spatial information is an important aspect of this research. We present pointers to the literature concerning spatial reasoning and representation, and discuss those aspects relevant to our research.

2.1 Knowledge Based Computer Vision

Image Understanding (IU) or Knowledge Based Vision has been an active area of research for the last thirty years. According to Chellappa and Kashyap [1992], it is traditionally concerned *with the design of computer systems that integrate explicit models of a visual problem domain with methods for extracting features from images and methods for matching features with domain models using a control structure.* Rather than mentioning all the work done in Image Understanding in representation, control structures, reasoning, evidence combination and uncertainty management, we highlight two points: (i) General purpose image understanding is an unrealistic, if not impossible goal, and (ii) even with domain specific knowledge in restricted domains, highly complicated representation and control structures are required to achieve acceptable results.

Research on IU has traditionally experimented with ways of integrating existing techniques into working systems and thereby discovering problems and solutions that otherwise would not have been uncovered. A typical IU system therefore would

1. extract meaningful 2-D grouping of intensity-location-time values,
2. infer surfaces, volumes, boundaries, shadows, occlusion, depth, color and motion using various types of knowledge,
3. group information into unique physical entities,
4. assign semantic labels to entities depending upon system goals and world models,
5. construct a consistent internal description and output an "interpretation"[1]

The central design issues of an IU system are representation and control.

[1] Various systems define interpretation depending upon what is convenient and appropriate to the problem domain.

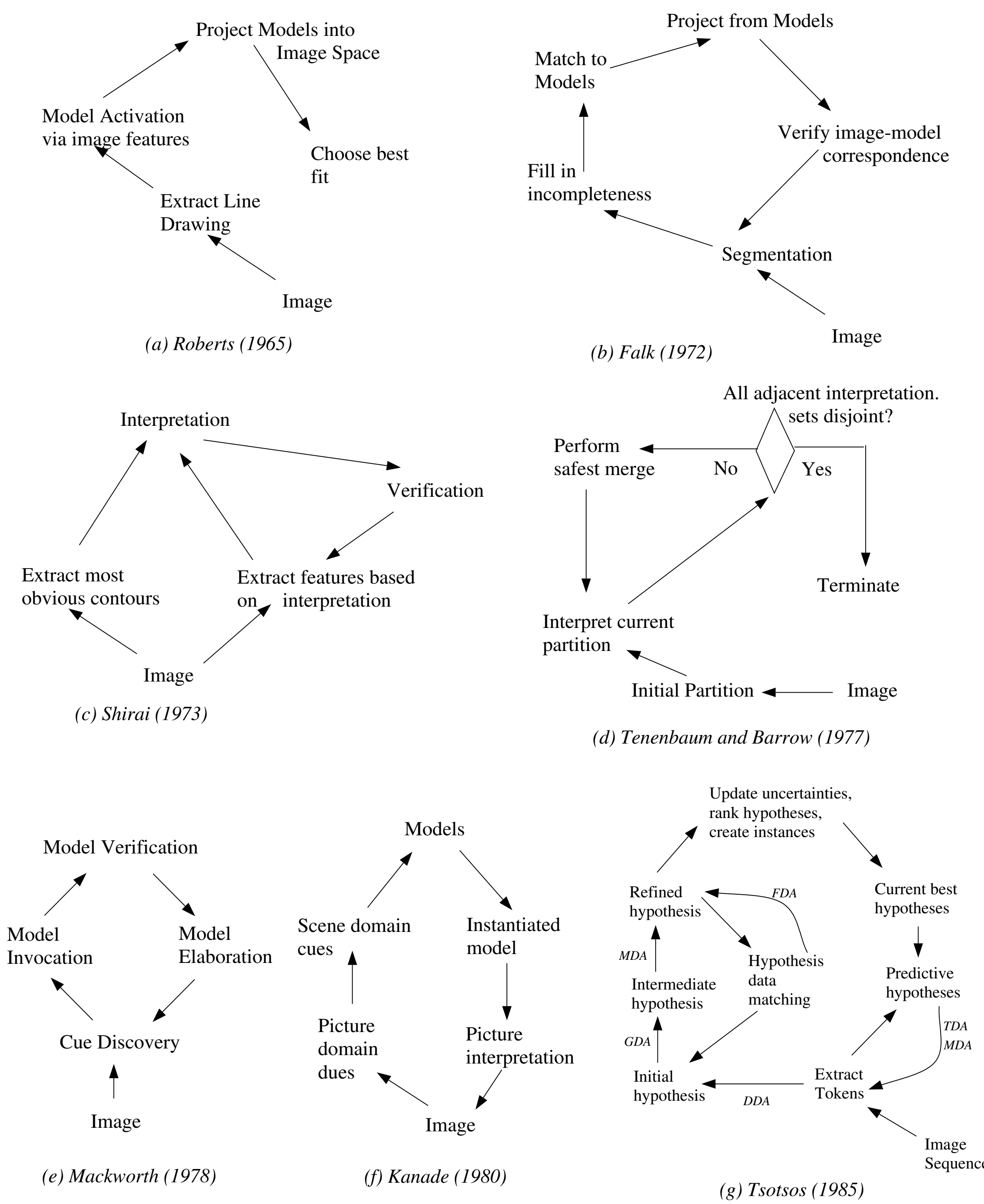
Representation: Generally, three levels of representation are considered: *Low level* includes image primitives like edges or regions; *intermediate level* includes boundaries, surfaces, volumes; and *high level* includes objects, scenes, or events. An IU system must have the ability to:

1. represent prototypical concepts,
2. organize them along aggregation, specialization and instantiation hierarchies,
3. represent and reason with spatial knowledge, which is the main type of knowledge in most vision systems,
4. reason with temporal knowledge to interpret spatio-temporal images as well as provide context for spatial information in static images.

Representational formalisms in IU followed trends in the general AI community. Minsky's frame theory [Minsky, 1975], was used in several vision systems including ACRONYM [Brooks, 1981], VISIONS [Hanson and Riseman, 1978], and ALVEN [Tsotsos, 1985]. Large collections of frames became unwieldy and were organized into semantic networks [Levine, 1978], that is, graph structures whose nodes represent objects or events and arcs represent relationships between the objects and events. The era of expert systems encouraged rule based IU system development[McKeown *et al.*, 1985]. More recently however, representation formalisms that capture uncertainties and imprecision have also been used. For example, Buxton, Howarth and Gong [Buxton and Gong, 1993; Howarth and Buxton, 1993] have investigated the use of Bayesian inference networks to represent contextual information in a dynamic visual surveillance task. Strat and Smith [Strat and Smith, 1988] developed a novel knowledge structure to store inconsistent information simultaneously as an *opinion* base. There has been recent work in building domain-independent control structures where interpretation can be formally stated based on the semantics of a standardized representation[Niemann *et al.*, 1990]. This removes the responsibility, from the IU developer, of specifying control strategies. The task becomes one of correctly representing the domain knowledge, the active knowledge sources, and the problem specification. However, many IU problems cannot easily be represented within this framework and are easier to encode using different control paradigms.

Control Requirements: An IU system typically employs a control system which selects which of the many inference, search, and matching techniques should be applied at a particular stage of processing. The basic control paradigm used in IUs is "hypothesize and test". However, since the search space in typical IUs is enormous, various search techniques such as model-directed search, goal-directed search, data-directed search, failure-directed search, hierarchical models, blackboard models and beam search have been applied. We briefly examine some of these techniques below. Since currently available computational processes are quite unreliable and visual data is inherently noisy, uncertainty measures are used in ranking hypotheses and determining the focus of attention in interpretation .

Historical Perspective on Control: Vision systems have attempted to exploit both *general* domain knowledge assumed applicable to all scenes and *model specific* knowledge of scene objects. Complex control strategies to utilize knowledge in the image interpretation task have been suggested [Firschein and Fischler, 1987; Chellapa and Kashyap, 1992]. Roberts [1965] was the first to devise a control scheme for organizing vision system components. His scheme defined a data driven best match approach shown pictorially in Fig. 2.1 (a). Falk [1972] closed the loop, (Fig. 2.1 (b)) allowing partly interpreted data to assist in the further interpretation of the scene. Shirai [1973] was the first to use a reduced image in a preprocessing stage to first find gross features for subsequent guidance in the search



(a) Roberts (1965) *(b) Falk (1972)*

(c) Shirai (1973) *(d) Tenenbaum and Barrow (1977)*

(e) Mackworth (1978) *(f) Kanade (1980)* *(g) Tsotsos (1985)*

Figure 2.1: *See paragraph* Historical Perspective on Control

of finer details as depicted in 2.1 (c). The interpretation guided segmentation (IGS) program of Tenenbaum and Barrow [1977] experimented with various sources of knowledge to guide the segmentation: unguided knowledge, interactive knowledge, both user driven and system driven; models; and relational constraints. They concluded that segmentation is improved with the application of knowledge and the more the knowledge, the faster the filtering process. 2.1 (d). The term *cycle of perception* was coined by Mackworth [1978] to denote the relations between cue discovery, model invocation, model verification and semantic interpretation (2.1 (e)). Kanade's modification of the cycle [1980] explicitly separated scene domain and image domain representations as shown in 2.1 (f). Tsotsos and coworkers [1985] further elaborated the cycle by specifying exactly at which points different hypothesis activation methods are applied (2.1 (g)). The different activation methods were driven by knowledge organization relationships and can be referred as goal-directed, model-directed, data-directed, failure-directed and temporally-directed activation methods.

Heterarchical models were developed by Badler [1975], Nevatia [1978] and Levine [1978] where a collection of separate modules performed specialized tasks and communicated with other modules as appropriate. *Hierarchical models* are similar to heterarchical models with restricted communication pathways reflecting an ordering of both processing steps and levels of abstraction. Important examples are the control structure of Barrow and Tenenbaum [1977] and that of the VISIONS system [Hanson and Riseman, 1978]. These have also been termed "pyramid models" since each level computes image properties at successively coarser resolutions and communicates only with computations in layers immediately above or below or within the layer. *Blackboard models* were first used in speech understanding and are a form of heterarchical models where communication takes place through a global data structure called a blackboard, though knowledge sources are organized hierarchically. The VISIONS system [Hanson and Riseman, 1978] incorporates this model. *Rule-based models* use inference rules to incorporate contextual information and both, data-directed and goal-directed reasoning is performed. We discuss below examples of rule-based systems where rules encode domain knowledge or models as well as specific contextual information.

Model Based Vision: IU is distinguished from model based object recognition where the main goal is to locate specific models and derive their transformation parameters in images [Chellapa and Kashyap, 1992]. However, the more important the role of top-down knowledge, the more similar the two become. Top-down knowledge effectively becomes a qualitative object model to be searched for in the image. Tsotsos has shown how contextual information computationally simplifies visual tasks [Tsotsos, 1992] Model based vision had originally relied on using known geometric models of objects to be recognized [Binford, 1982]. Using generic or parametrized models instead of precise geometric models has been researched [Brooks, 1981; Fua and Hanson, 1987], as have techniques that do not rely upon shape alone. Hawkeye [Barrow and Tenenbaum, 1977], MSYS [Barrow and Tenenbaum, 1976] and Condor [Strat, 1992] are examples of systems that avoid stored geometric models and integrate cues such as size, context, location and appearance to attempt recognition. Early work in using domain knowledge to guide vision includes the use of *appearance models* by Selfridge [Selfridge, 1982]. More recently, work by Strat and Fischler [Strat and Fischler, 1991], and others, re-affirm the advantage of using context at all levels in vision processing.

Context Based Vision Strat [Strat, 1992] provided a classification of contextual information, a formalism for representing it and a conceptual model for using context in object recognition. We propose to extend the use of contextual information for high level image interpretation tasks. Such a task might involve object identification as well as recognition.

Srihari[Srihari, 1991] showed in her thesis how collateral information from captions accompanying news photographs could be used to identify people in the photographs. A hypothesis of the image contents is generated, then a routine developed by Govindaraju [1992] to locate face candidates in the image is called and graph matching techniques are used to establish correspondence between face candidates and persons mentioned in the caption. As we shall see later, we extend the use of collateral information in this domain, as well as make the ideas developed in Srihari's thesis suitable for application in other domains.

Recently, there has been a lot of academic interest in the *Integration of Natural Language and Vision* [Srihari, 1995b; McKevitt, 1994; Srihari, 1995a]. One of the objectives of that research effort is to use the interpretation of data in one modality to drive the interpretation of data in the other. We highlight the fact that collateral based vision exploits a *reliable* hypothesis of scene contents. We obtain this hypothesis from sources other than bottom-up vision to aid the vision processing. However in the INLV community, there is sufficient interest in using bottom-up vision to generate natural language descriptions of scenes; to use vision techniques in natural language understanding and to model deep representation and semantics of language and perception. In this dissertation, we choose a far more conservative goal of examining how image interpretation can benefit from collateral information.

A Priori domain knowledge has recently been used for video processing. Bobick and Pinhanez [Pinhanez and Bobick, 1995] use an *approximate world model*, which are coarse descriptions of objects and actions in the world, as the appropriate representation of precompiled contextual information. They use this information to control a simulated moving camera using high level commands. The contextual information is derived from the script of the scene and from wide-angle low-resolution imagery. Simple view-based vision techniques are used in the appropriate context. However, the acquisition of contextual information such as action types and boundaries, and the semantics of actions is not addressed. The feasibility of this approach depends on the extraction and representation of action and temporal boundaries. Furthermore, limited common-sense reasoning capabilities will be necessary to move beyond pre-compiled domain knowledge to dynamic scene specific interpretation, even in a limited domain.

Animate Vision More recently however, the methodology and goals of image interpretation have been questioned. According to Ballard and Brown[1992], vision is a dynamic, sequential and interactive process computationally constrained and simplified by behavioral assumptions. They claim that the general problem of associating many models to many parts of the image simultaneously, as generally advocated in image understanding, is too difficult to be performed in real time. Trying to find a known object (location) or trying to identify an object whose location is known (identification) are alternate simplified problems that they suggest will eventually form efficient components of complex goal oriented visual tasks. Though most of our application domains are about static scenes, the task oriented and context driven paradigm advocated by the active vision community greatly influences our architecture.

2.2 Constraint Satisfaction

Constraint satisfaction techniques have been used in computer vision for a variety of tasks. Continuous relaxation techniques have been used for lower level noisy data matching whereas discrete relaxation techniques have been used for high level relational structure matching. [Kittler and Illingworth, 1985; Price, 1985] Framing the image features as variables (nodes), and semantic labels as values is a common aspect of all prior usage of CSP in vision(for e.g. Mapsee [Mulder *et al.*, 1988], MSYS [Barrow and Tenenbaum, 1976]).

This is because of the bottom-up approach where features from the image are extracted first and then matched with a symbolic description. However, as pointed out by Grimson [Grimson, 1992], CSP techniques become very expensive in the presence of spurious data or wrong model choices. This highlights the importance of additional information sources to reduce (remove) the ambiguity in interpretation.

Consistency algorithms, also called discrete relaxation algorithms, were first introduced by Waltz for interpreting polyhedral scenes with shadows [Waltz, 1975]. The central idea is the propagation of constraints through a network of hypotheses. Each node of the network represents a given fact with several interpretations. Labeling each node with exactly one hypothesis is called the labeling problem. Constraints on these nodes allow one to discard some of these interpretations. The labeling problem includes the graph-coloring problem and therefore is NP-hard [Garey *et al.*, 1974]. Consistency algorithms arose from a backtracking technique called *forward checking* [Haralick and Elliot, 1980] that prunes the search tree of nodes based upon incompatibility with the partial labeling constructed so far. Consistency algorithms go one step further in removing incompatible future labelings. These inconsistencies would have otherwise been discovered repeatedly by most backtracking procedures. Discrete relaxation runs in polynomial time and only produces a locally consistent solution. In the case of a tightly constrained CSP, an iterative application of the algorithm prior to backtracking can greatly reduce or totally eliminate the tree search. Such techniques have found wide application in pattern recognition and image analysis, temporal reasoning, and scheduling.

The labeling problem can also be solved using continuous relaxation techniques [Faugeras and Berthod, 1981] which, although capable of handling fuzzy constraints, are difficult to control. The user has to juggle various coefficients of the system to obtain the correct solution. Discrete relaxation handles only boolean constraints and is thus easier to manipulate. A series of algorithms, notably **AC3** [Mackworth and Freuder, 1985],

AC4 [Mohr and Henderson, 1986] and **AC5** [Hentenryck *et al.*, 1992] are available for arc-consistency. An excellent reference on constraint satisfaction problems and the various approaches to solving them can be found in [Tsang, 1993]. A generalization of AC4,that handles general non-binary constraints, called **GAC4** was presented in [Mohr and Masini, 1988].

A *static* constraint satisfaction problem [Mohr and Henderson, 1986; Mackworth, 1977] represented by (V,D,C,R) involves a set of m **nodes**, $V = \{v_1, v_2, \ldots v_m\}$, each with an associated **domain** D_i of possible **labels**. The search space consists of the Cartesian product of the nodes' domains, $D = D_1 \times D_2 \times \ldots \times D_n$.
A set of **constraints** C is specified over some subsets of these nodes, where each constraint C_p involves a subset $\{i_1, \ldots, i_q\}$ of V. C_p is labeled by a relation R_p of R, a subset of the Cartesian product $D_{i_1} \times \ldots \times D_{i_q}$ that specifies those labels of the nodes that are compatible with each other.
A **binary** constraint involves only two nodes. This notion is extensible to an **n-ary** constraint over n nodes. The task in a CSP is to assign labels to nodes such that all the constraints on the nodes are satisfied simultaneously. Depending upon the application requirements, one might need to find any *one*, *all* or *optimal* solutions. Problem reduction is a class of techniques for transforming a CSP into problems that are hopefully easier to solve. Although problem reduction alone does not normally produce solutions, it can be extremely useful when used in conjunction with search methods.

As we shall discuss later, our architecture for collateral based vision is suited for task driven interpretation, where we locate and identify a set of strongly hypothesized objects only. However it does not preclude usage in a more heterarchically organized architecture where hypotheses could be generated from both domain knowledge and image data.

2.3 Spatial Reasoning

Since the image interpretation tasks that we consider involve collateral information about objects arranged in space, representing and reasoning with spatial relations is an important component of our image understanding architecture. Over the past three decades, a tremendous amount of work has been done on the representation of spatial knowledge. Depending upon the field of study and/or the principal applications in mind, alternatives include analogical and propositional geometrical descriptions, topological and qualitative representations and linguistically motivated representations. Excellent overviews of the area of spatial reasoning are available in [McDermott, 1992]. A study by Freska and Röhrig [Freska and Röhrig, 1993] compares various research projects on spatial representation along the following dimensions:

- *Frame of reference*, having local or global scope, internal or external, Cartesian, polar etc.
- *Primitives for representation*: Points, surfaces, volumes and other geometric entities.
- *Spatial aspects represented*: Topology, arrangement, distance, orientation, size and shape.
- *Dimensionality, modularity, granularity and vagueness* of the representation.

An excellent overview of qualitative spatial reasoning is presented in [Hernández, 1994]. The "imagery debate" in cognitive science [Kosslyn *et al.*, 1981; Pylyshyn, 1981] led to the categorization of spatial representations into *pictorial* [Funt, 1980] or *propositional* [Reiter and Mackworth, 1989] variants, each with associated problems. Hybrid models that combine these approaches are seen in [Chandrasekaran and Narayanan, 1990] and [Hernández, 1994]. NLP research (primarily in the area of understanding spatial prepositions) where the focus has been on determining the context-dependent meaning of the language constructs, has also presented models for the qualitative representation of space [Herskovits, 1986; Yuhan and Shapiro, 1995].

The database community has also been interested in the representation, reasoning and management of spatial data, with a focus on efficient storage and retrieval [Topaloglou, 1993].Relational algebraic models have been proposed [Gütting, 1988] and various data structures and indices for efficient storage and retrieval of limited geometric data types are constantly being improved upon [Chang *et al.*, 1987; Jungert, 1994; Papadias and Sellis, 1993].

We mainly use spatial reasoning as a tool, depending upon the domain, to facilitate vision. Spatial representation and reasoning are active research fields with insufficient consensus on representation of space [Olivier, 1995] or shape [Kimia *et al.*, 1995]. We choose to leave spatial reasoning as a separate module in our architecture for collateral based vision, employing whichever best suits the application. Shape representation for vision is typically geometric with the primary interest being shape reconstruction or object modeling. The spatial reasoning community is more concerned with how shapes partition space and how to categorize shapes the way humans do. We need to evaluate both in the context of collateral based vision.

An early use of spatial knowledge in semantic networks was by Russell [1979] in what he called *location networks*. Many vision systems [Brooks, 1981; Reiter and Mackworth, 1989] encode domain specific geometric knowledge and rules and employ them in hypothesis generation and verification. We show how picture specific spatial information can be used in hypothesis generation and verification.

Chapter 3

Collateral Based Vision

3.1 Introduction

To solve the inherently under-constrained task of image interpretation, additional sources of knowledge have successively been added to provide the necessary constraints for the recognition of objects and understanding of scenes.

Vision systems have attempted to exploit general knowledge assumed applicable to all scenes and specific knowledge of scene objects and their appearance/configuration (model based vision). As a result, complex control strategies to utilize knowledge in the image interpretation task have been suggested. Devising control structures for knowledge based vision has been an active area of research since the mid 1960's with several refinements of the *cycle of perception* and the development of hierarchical, heterarchical and blackboard models. See [Firschein and Fischler, 1987; Chellapa and Kashyap, 1992] for the historical development of control structures in image understanding. Some vision systems [Weymouth, 1986] employ the idea of top-down control in image understanding, by providing the general context of the picture (e.g., airport scene, typical suburban street scene). where to look for the object. There has also been recent work in domain-independent control structures where interpretation can be formally stated based on the semantics of a standardized representation[Niemann *et al.*, 1990].

Our research has focused on developing a computational model for "understanding" pictures based on accompanying, descriptive text. Understanding a picture can be informally defined as the goal-driven process of locating and identifying objects whose presence in the image is suggested by the accompanying text. The ideas presented in this chapter carry the notion of top-down control one step further, incorporating not only general context but a high-level description of the actual picture.

This problem can be divided into two major sub-tasks. The first task is the processing of language input for extracting and representing useful information from collateral text accompanying an image. This information is represented as a set of constraints. The processing of captions for generating constraints is discussed in [Srihari and Burhans, 1994]. The second major task involves the design of an architecture wherein collateral information from text can be efficiently exploited in the process of image interpretation.

In describing our architecture for collateral based vision, it is useful to present working examples from an application domain based on this architecture. Therefore, we first present an overview of **Show&Tell** in Section 2. The next section describes the knowledge representation that embody the declarative scene hypothesis. Section 4 describes the image interpretation algorithm along with an illustrated example. The theoretical framework for interval based spatial reasoning including the domain independent algorithm HetCSP is presented in the next chapter on constraint satisfaction.

3.2 Show & Tell Overview

Show&Tell is a system which combines speech recognition, natural language processing and image understanding. The goal of this image annotation project is to take advantage of advances in speech technology and natural language (NL)/image understanding (IU) research to make the preparation and querying of image-related collateral information viable.

An image analyst views the image and interactively detects roads (semi-automatically). The image is then automatically partitioned based on road networks. Next, the analyst describes, in spoken language, objects or regions in the image. A state of the art speech recognition system is employed in transcribing the input .

The resulting narrative is processed by a natural language understanding component that generates visual and spatial constraints on the image. This in turn, is used by an image interpretation system in detecting and labeling areas, roads and buildings in the image. Finally, an interface to attach additional collateral to objects that have been identified is provided. The result of the processing is thus an annotated image consisting of a semantic representation of the collateral text co-referenced with the locations of objects identified in the image. Sophisticated querying based on contextual and spatial information is now enabled. The Show&Tell system is described in greater detail in Chapter 5.

3.3 Knowledge Representation

The focus of this section is on semantics of the knowledge structures required for collateral based vision rather than the KR formalism that is used to express them. A description logic language like LOOM[ISX, 1991b] presents a powerful tool for constructing such a knowledge base.

3.3.1 Semantic model:

The semantic model forms the interface between the collateral generation processes and the image interpretation processes. The collateral information is extracted from natural language text, speech commands, and deictic input from standard computer interfaces such as the mouse. We emphasize that the representation defines the interpretation process. In other words, the control of the interpretation process is is truly domain independent in so far that you can model the problem using the knowledge structures defined below. To apply our Collateral Based Vision (CBV) system to another domain, one would typically augment the semantic model with domain specific concepts and relations that are specializations of the ones provided here. We have demonstrated the applicability of these structures and the benefits of the associated control algorithm in two implemented systems *Piction*[Chopra and Srihari, 1995; Srihari, 1995c] and *Show&Tell*.

Domain Independent Model

The scene hypothesis consists of descriptions of entities of the domain world (instances of specializations of the concept `world_entity`), their visual properties and spatial relations between the entities.

The concept `conceptual_graph` (Fig. 3.1) represents all the information about an image. Its constituents are

`source`: Refers to the image file information (size, capture date, format, etc.);

`collateral_info`: Refers to the raw-analyzed textual collateral data , and

`visual_model`: Refers to the symbolic information extracted from the collateral information (scene hypothesis) with links to existing instances in the knowledge/data

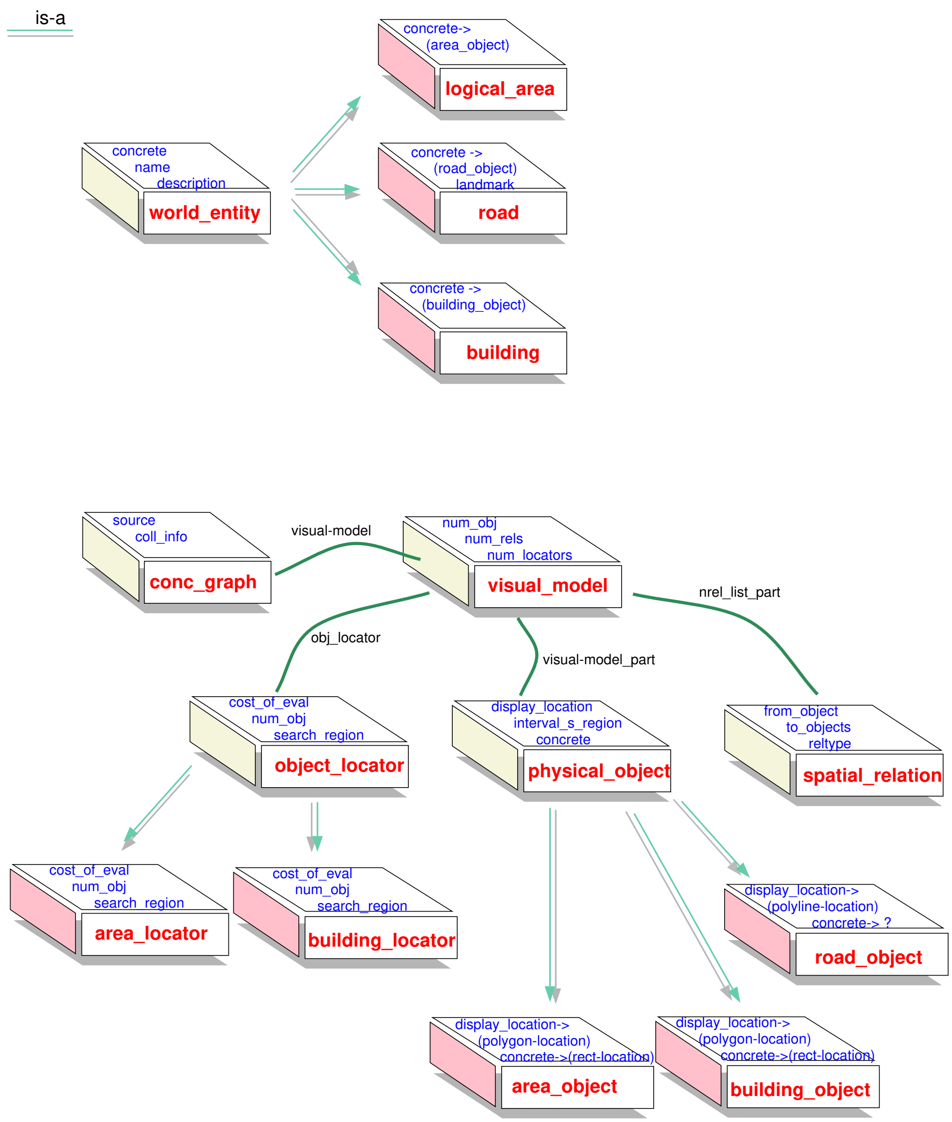


Figure 3.1: *Concept Hierarchy*

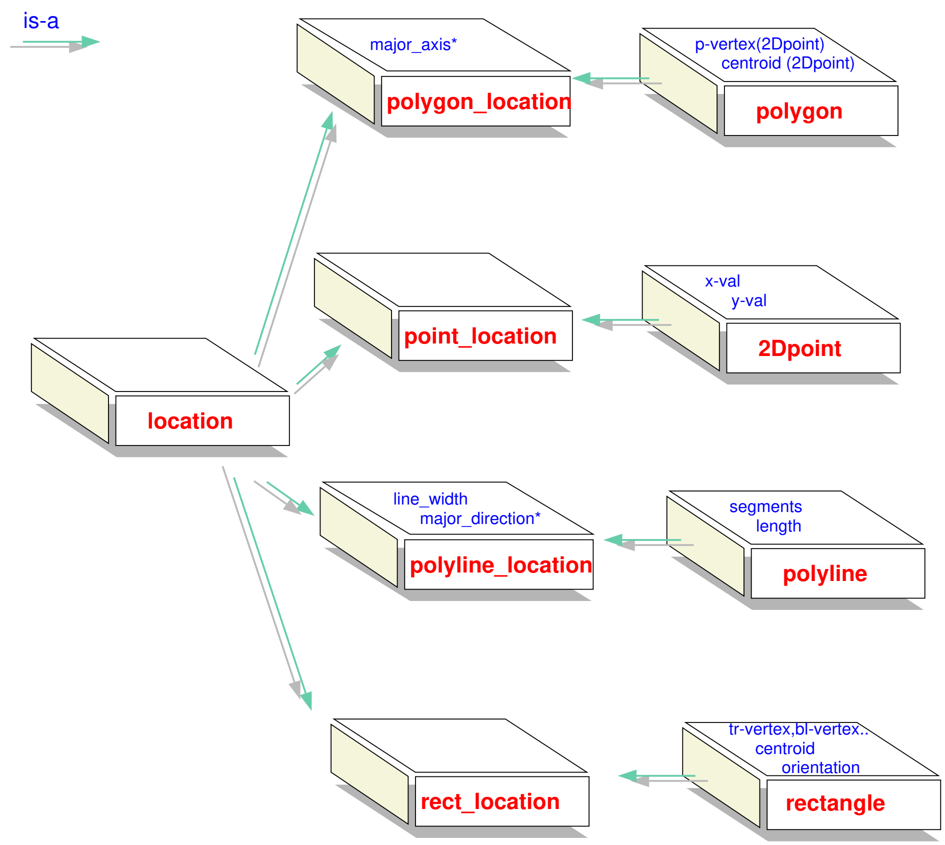


Figure 3.2: *Concept Hierarchy (contd)*

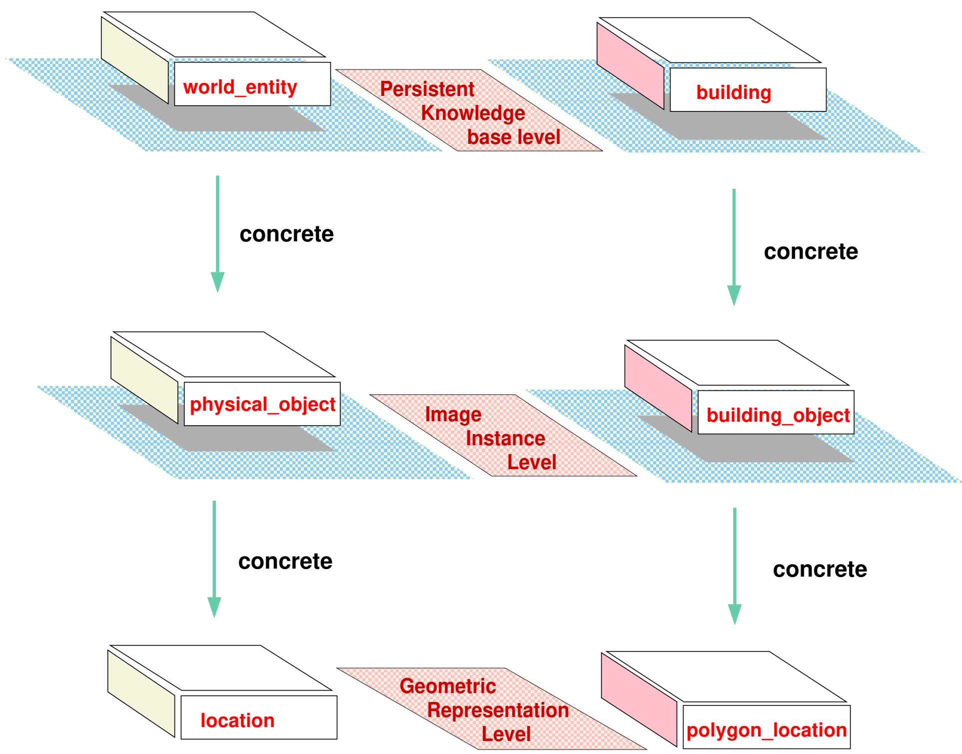


Figure 3.3: *Three level representation of objects*

base, as well links to the symbolic structures extracted from the interpretation of the image based on the scene hypothesis.

The database is populated with instances of `world_entity`. The domain world of **Piction** consists of named instances of the `person` concept, say, Bill Clinton. The database might contain more than one image featuring Bill Clinton. We represent the image specific information pertaining to the instances of `world_entity` by instances of the concept `physical_object`. We define a relation `concrete` [Niemann *et al.*, 1990] between the two levels of abstraction. For example, a `person` is made `concrete` by a `face_outline_object` in Piction (or a `human_body_object` if available object location techniques were able to recognize other body parts).

The domain world of **Show & Tell** consists of named instances of the `building`, `road` and `logical_area` concepts. These are similarly made `concrete` as `building_object`, `road_object` and `area_object` respectively.

The natural grouping of objects done by the image analyst influences the object location and identification process. When an analyst (or collateral information source) mentions a collection or group of objects of the same type, those objects are then marked to be searched for simultaneously. This effect reflects our assumptions that a non-localized search would lead to a lot of false candidates, and that it is easier for an object locator to find a specified number of similar objects than search for them individually. We define a concept `object_locator` (Fig. 3.1) to represent the information required to locate an aggregate of objects of the same type. The region of the image that the objects are predicted to be located in is the `search_region`. Its default value is the complete image extent.

For example, *"Area Alpha contains four L-shaped buildings"* would lead to the creation of an instance of the `building_locator` concept. The shape information and the number of objects of this type are also stored in this instance. The `search_region` takes the default value and upon later interpretation involving spatial reasoning, the search region may be shrunk to a smaller region of the image (possibly corresponding to the extent of Area Alpha). An image containing say, 9 buildings may contain 1 to 9 `building_locator` instances depending upon how the analyst chose to group the buildings based on spatial proximity and/or shape similarity. We discuss the interpretation process in greater detail in section 4.

As shown in Fig.3.1, we represent the spatial extent of a `physical_object` in three ways:

`display_location`: is how the spatial extent of the object is to be displayed on the screen. In our application, the `display_location` of an instance of a `building_object` is an instance of `polygon_location` and represents a two-dimensional polygonal outline.

`concrete`: is the connection of the `physical_object` to the lower level of abstraction and is the representation used for spatial reasoning. For example, an instance of a `building_object` is made concrete as an instance of `rectangle_location`. In the Show&Tell implementation, bounding boxes suffice to provide reasonable semantics to the spatial relations. For more sophisticated handling of spatial expressions, the location object may be more complex.

`interval_s_region`: is the representation of the search space of the spatial extent of the `physical_object`. It represents a set of *interval constraint variables*. The number of variables is determined by the `concrete` representation of the object. For example, 4 interval variables are required to represent the `interval_s_region` of a `building_object` made concrete by a `rectangle_location`. Since `rectangle_location` is an upright bounding box, no orientation information is required and 4 parameters are sufficient to represent it.

In applications where the `concrete` representation is a three dimensional model and the viewpoint can be specified, LOOM allows for the `display_location` to be procedurally defined to compute the project of the 3D model for display purposes.

Note that the `concrete` relation connects *three* levels of representation for the same entity (see Fig.3.3). For example, the Empire State Building would have one `building` instance in the database, several `building_object` instances, one for each image containing the building, and an equal (or lesser if building detection fails) number of `polygon_location` instances.

Spatial relations are defined as specializations of the LOOM concept `Nrel`[1]. They are defined between `physical_objects`.

Show&Tell Domain Model

The **Show&Tell** domain model consists of specializations of the concepts of the domain-independent model. These specializations are shown in pink shaded boxes in Figures 3.1, 3.2 and 3.3 . The object location algorithms we have are for qualitative *building* detection [Zhang and Srihari, 1996]. The set of spatial relations that are employed in interval based spatial reasoning are `left_of, right_of, strict_left_of, strict_right_of, above, below, strict_above, strict_below, between, included_in, includes`.

The VEIL Project at USC/ISI is a recent example of combining semantic knowledge image understanding. VEIL performs high level interpretation of scene images using the deductive capabilities of the LOOM knowledge representation system[MacGregor, 1994; ISX, 1991b; MacGregor, 1991] and provides users with semantically enriched browsing and

[1]Non binary relations (involving more than 2 objects) are not easily implemented in LOOM. We therefore choose to represent spatial relations as instances of specializations of a concept. In description logics, a concept is equivalent to a unary relation

editing facilities. A knowledge base containing symbolic descriptions of objects, relations and events is augmented with references to quantitative object models of a lower level IU system. While we also use LOOM to conceptually represent our application domain and support queries to the image database, we have augmented the inference mechanism with a constraint satisfaction based control algorithm. This algorithm effectively utilizes contextual information in controlling the object location and identification process. While we maintain an architectural separation between the high level scene interpretation and the lower level IU system, the emphasis is on *top down* control of the object location tools (of the IU system) using the high level collateral information available. The VEIL project, on the other hand, adds a higher level semantic model to the site model of the image for querying with semantically relevant entities.

3.4 Image Interpretation

Srihari[Srihari, 1995c] showed how collateral information from captions accompanying news photographs could be used to identify people in the photographs. A hypothesis of the image contents is generated, then a routine to locate face candidates in the image is called and graph matching techniques are used to establish correspondence between face candidates and persons mentioned in the caption. We extend the use of collateral information in this domain by (i) devising a uniform representation for domain and picture specific constraints, (ii) employing spatial reasoning to use partial interpretation results in guiding the application of the vision processing tools. (iii) generalizing the problem representation in an object oriented manner to deal with multiple object types (not just faces).

The idea of using partial interpretation results to localize the search for related objects has been used in several vision systems. For example, constrained search was used by Selfridge [Selfridge, 1982] to guide the search for building shadows, but with minimal spatial reasoning. Given a reference object (RO), an object to be located (LO) and a spatial relation between them, the position of the RO and the spatial relation is used to constrain the search region for the LO. Wixson and Ballard [Wixson and Ballard, 1994] give a rigorous cost-benefit analysis of this technique when there is one target object (LO) and one intermediate object (RO). Current approaches discuss visual search in isolation, that is, the RO and LO are uniquely identified. The combination of visual search with the object labeling problem as we propose, leads to choices in ROs and LOs, planning and incremental constraint satisfaction. We assume cost and reliability measures for our object locators and attribute verifiers are available.

3.4.1 Control Algorithm

We now describe in detail the control algorithm that takes the declared scene hypothesis as input and incrementally applies the vision tools to ground the scene hypothesis with quantitative values from the image. The abstract, domain independent version of this control algorithm called **HetCSP** that is suitable for testing on randomly generated data is presented in detail in section 4.2.2, page 51. We present an instance of that algorithm as it maps to the domain of Show&Tell. We initialize the incremental finite-domain constraint satisfaction problem with no variables and labels. However, the interval constraint satisfaction problem is initialized with the complete set of variables and initial values corresponding to the extremal values that the variables could take. The control algorithm loops over three stages (see Fig. 3.4):

Decision Stage: A cost/utility function is employed to decide which object class to detect *next*, given the current state of the constraint graph and labelings. The factors taken

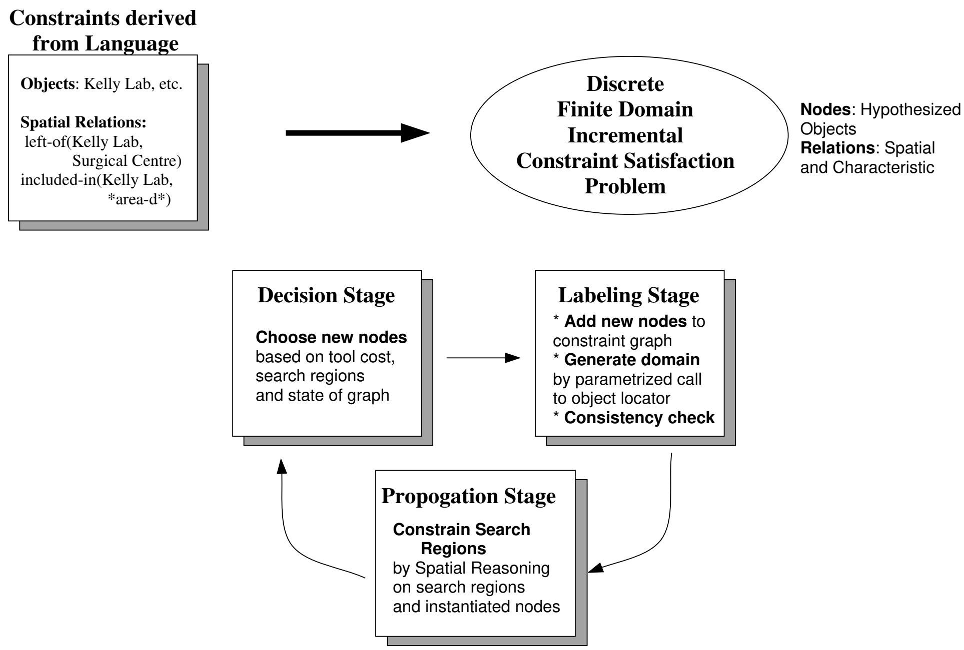


Figure 3.4: *Localizing the search for objects using spatial reasoning*

into account for this function are the tool cost, reliability, number of objects to be found, size of search region, number of constraints between this class and the objects of interest and the level of interest in this particular class of objects. Some measures are class specific (tool cost), others are instance specific (number of objects) and some are dynamic (size of search region).

Labeling Stage: The `search_region` for the corresponding `object_locator` instance is computed from the `interval_s_regions` of the objects in the group. For example, the interval search region of a 2D object corresponds to 2D bounding box. The `search_region` for the group is therefore computed as the bounding box of the previously mentioned bounding boxes. In this example (and our first implementation) this simple representation suffices, however, the search regions could be polygons or quad-trees and the computation of the *hull or union or extent* could be similarly more sophisticated. The corresponding object location algorithm uses the search region (apart from other collateral) to locate candidates for objects of the selected group. These objects, candidate labels and the constraint relations between the objects are inserted in a constraint graph representing the partial interpretation. Constraints between objects in the new group and between objects in prior groups and the new ones are satisfied to increment the partial interpretation of the image.

Propagation Stage: We update the interval variables corresponding to those finite-domain (FD) CSP variable whose list of labels has just been created in the previous stage. The bounds of the interval variables are set to the extremal values of the FD labels. For example, if an object had three candidate labels generated by the object location stage and satisfying all current constraint relations, then the `interval_s_region` for that object would be set to the bounding box (or convex hull) of the three label `physical_location` instances. The choice of the representation of the interval search region is dictated by the spatial reasoning capabilities of the implemented system. The essential idea here is of representing all allowed labels for the variable by a continuous bounded approximation that can be spatially reasoned with. Next, the interval based bounds propagation algorithm [Siskind and McAllester, 1993b] ensures that the rest of the interval variables take values that satisfy the constraints.

The effect of locating objects of a particular class can then be summarized as (i) potential identification of objects already located, (ii) labeling and partial identification of objects of the current class and (iii) constraining and ordering the search for other objects. The algorithm halts when all object groups (`object_locator` instances) have been searched for. As we have mentioned, an object locator is parameterized by the number of objects of that particular class. Since the algorithms for object-location are not always accurate, we might obtain too few or too many object candidates. Information from the spatial prediction stage is used at this point to attempt a solution. The object locator routines are called again on the constrained search region with relaxed parameters if there were initially too few candidates. If there were extra candidates, the information is used to order the candidates and the top choices are picked.

3.4.2 Example

Figure 3.5: *Result of road detection, landmark suggestion and speech recognition*

To illustrate the use of collateral information, consider the aerial image (Fig. 3.7.a) and accompanying speech input. A semi-automated road following algorithm generates a network of system-labeled road segments as described in Section 3. This network along with the image boundaries forms a planar graph. A graph algorithm finds and labels the *faces* [Harary, 1971] of this planar graph. Each internal face represents a `physical area` demarcated by roads and could correspond to a `logical area` or serve as a **landmark**. These landmarks are then used by the analyst in describing the contents of the scene. The

```
This image depicts jarvis park.
Label segments roger two roger four and
roger five as main street.
Label segment roger three as king street.
In the western half of area delta is the
baird research institute.
It consists of four buildings.
Of these the leftmost is a long rectangular
building.
Label this as the kelly laboratory.
This is used for developing and testing new
products.
Label the l-shaped building as the operations
center.
Label the large two storied building as the
headquarters building.
This is where most of the administrative
offices are located.
Label the small square building as the
communications building.
```

Figure 3.6: *Sample Speech Annotation for Image in Figure 3.5.*

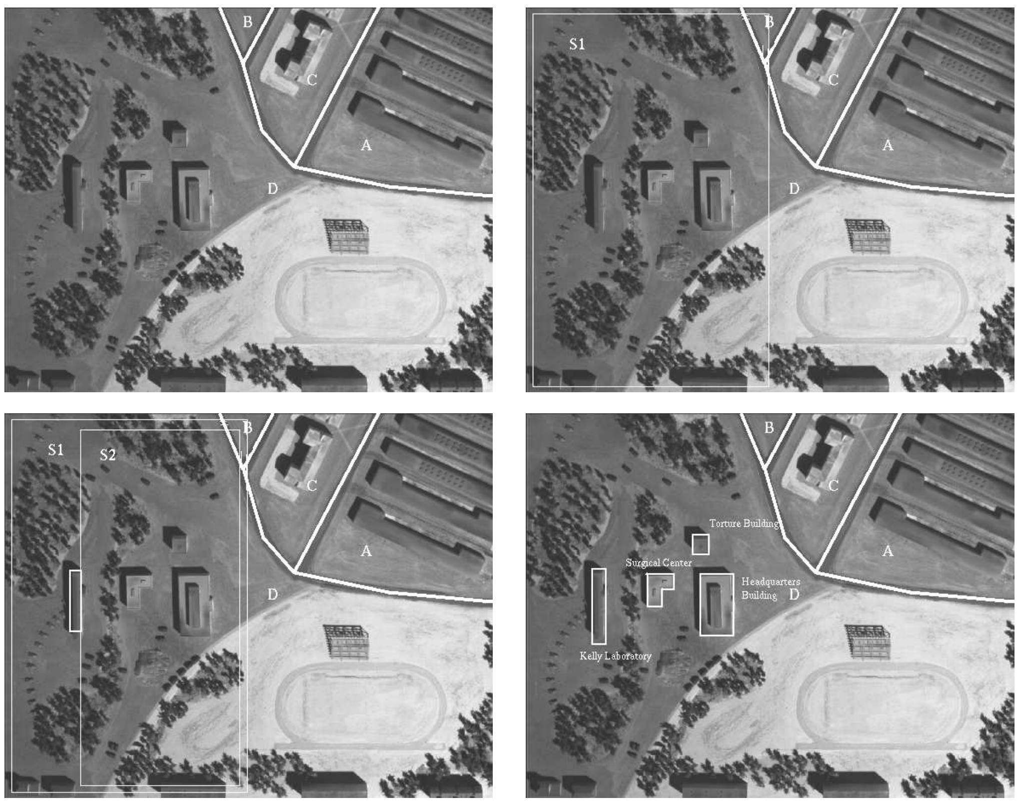


Figure 3.7: *An illustration of constrained search. Top left: Image with landmarks and suggested areas A-D. Top right: Constrained search region (S1) for Baird Research Institute. Bottom left: Kelly lab. located and constrained search region (S2) for other buildings. Bottom right: All buildings and areas located and identified.*

result of the speech recognition and language processing results in a set of (LOOM) assertions that collectively define the scene hypothesis. The set of objects to locate and identify are: three `logical_area` instances and four `building` instances. In this domain, `physical_area` instances are either outlined by the user or system *a priori* or are computed as the minimum bounding box of the objects explicitly mentioned to be contained in it. In the latter case, the `logical_area` instance is bound to the smallest search region specified for the corresponding `physical_area` till all contained objects are located. Furthermore, area objects are easier to locate than building objects. Specialized buildings are more expensive to locate than simple rectangular buildings. Such information is encoded *a priori* in the concepts for object locators.

The `logical_area` Jarvis Park is bound first to the image object *PL0*. For every system labeled physical area object that is referred to by the analyst, a `logical_area` object is created. Therefore, an instance is created for area Delta. The control algorithm then constrains the search for Baird Research Institute to the western half of area Delta (D) by altering its search region. The actual boundary of this area will be computed by the bounding box of the objects it contains and it temporarily bound to its search region *S1* (see fig 3.7). The search regions of the buildings are then constrained to *S1*. Next, the algorithm chooses to search for long rectangular buildings using the qualitative models of buildings encoded in the object locator tools. The choice of which object type to look for next is encoded in a few heuristic rules in the decision stage based on size of search region, number of objects, tool reliability, etc. Once the long rectangular building is located it constrains the search regions of the other buildings to search region S2. Finally, the four buildings are labeled and Baird Research Institute's boundary is computed.

3.5 Summary

We have presented a general technique to incorporate collateral information in the high level computer vision task of object location. The declarative specification of the scene hypothesis and the domain independent control algorithm that exploits spatial information to drive the vision process, are two major contributions of this dissertation presented in this chapter.

Chapter 4

Constraint Satisfaction: A Paradigm for Collateral Based Vision

In this chapter we describe three contributions of this dissertation in the field of constraint satisfaction that make it an appropriate paradigm for collateral based vision. These three contributions are:

1. A consistency algorithm (NAC4) for a certain class of non-binary constraints that are represented implicitly.
2. A greedy algorithm that verifies binary and non-binary constraints based on domain specific costs.
3. A constraint satisfaction technique HetCSP, that forms the basis of the domain independent control algorithm for collateral based vision. This technique is coded as an incremental, numeric, discrete and finite domain constraint satisfaction algorithm coupled with bounds propagation on intervals.

We first introduce the area of constraint satisfaction and provide the definitions necessary to present the algorithm NAC4. Next we consider expensive constraints and finally we present algorithm HetCSP.

4.1 Consistency Algorithms and Expensive Constraints

Consistency algorithms, also called discrete relaxation algorithms, were first introduced by Waltz for interpreting polyhedral scenes with shadows [Waltz, 1975] posed as a labeling problem. Traditional techniques for constraint satisfaction include chronological backtracking and backtracking with forward checking [Haralick and Elliot, 1980]. The latter prunes the search tree of nodes based upon incompatibility with the partial labeling constructed so far. Consistency algorithms go one step further in removing incompatible future labelings. These incompatible labelings would have otherwise been discovered repeatedly by most backtracking procedures. Discrete relaxation runs in polynomial time to produce a locally consistent solution. When the Constraint Satisfaction Problem (CSP) is tightly constrained, application of consistency algorithms can greatly reduce or totally eliminate subsequent backtracking. These techniques have found wide application in artificial intelligence, pattern recognition and image interpretation.

The labeling problem can also be solved using continuous relaxation techniques [Faugeras and Berthod, 1981], that are capable of handling fuzzy constraints. However, these are difficult to control because the user has to juggle various parameters of the system to obtain the correct solution. In contrast, discrete relaxation techniques handle only boolean constraints and are easier to manipulate.

Many algorithms for binary arc-consistency have been proposed: **AC3** [Mackworth and Freuder, 1985], **AC4** [Mohr and Henderson, 1986] and **AC5** [Hentenryck *et al.*, 1992]. A generalization of AC4 to handle non-binary constraints, **GAC4**, was presented in [Mohr and Masini, 1988].

Consistency algorithms for finite domains assume that constraints are enumerated as explicit relations with little or no cost, and that verifying the existence of a tuple in a relation is a unit cost operation. However, there are applications of CSP, in domains such as computer vision [Chopra and Srihari, 1995], where enumerating the complete extension of a constraint relation is neither inexpensive nor necessary. When dealing with expensive constraints, reducing the domains of the relation's nodes prior to computing the relation's extension seems an obvious improvement over computing the extension first.

4.1.1 Definitions

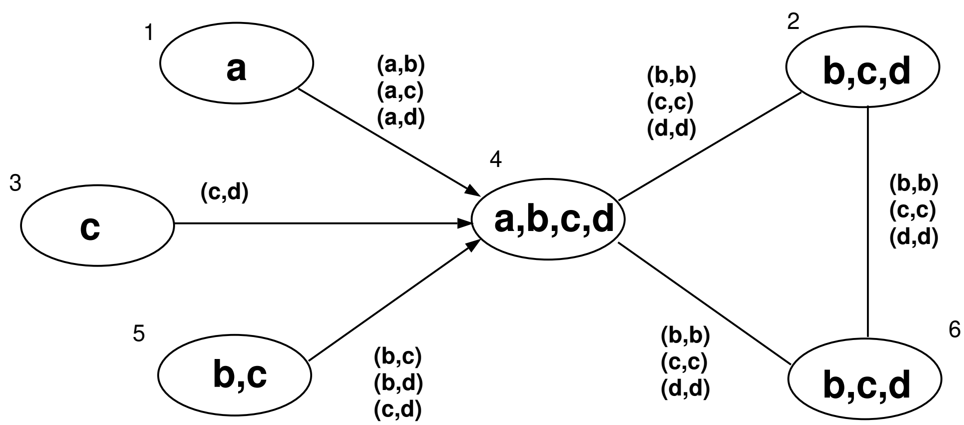


Figure 4.1: *An example of a binary CSP with constraints modeled as explicit relations. The ovals are nodes, values in them are labels and the lists of pairs are constraints.*

A *static* constraint satisfaction problem [Mackworth, 1977] is defined as a 4-tuple (V,D,C,R) where V is a set of **nodes**, $\{v_1, v_2, \ldots v_m\}$, each with an associated **domain** D_i of possible **labels**. (In Fig.4.1, adapted from [Bessiere, 1991], D_2 is $\{$b,c,d$\}$)

The search space D, consists of the Cartesian product of the nodes' domains, that is $D_1 \times D_2 \times \ldots \times D_n$.

A set of **constraints** C is specified over some subsets of these nodes, where each constraint

C_p involves a subset $\{v_{i_1}, \ldots, v_{i_q}\}$ of V. C_p is labeled by a relation R_p of R, a subset of the Cartesian product $D_{i_1} \times \ldots \times D_{i_q}$ that specifies those labels of the nodes that are compatible with each other.

A **binary** constraint involves only two nodes. This notion is extensible to an **n-ary** constraint over n nodes. The task in a CSP is to assign labels to nodes such that all the constraints on the nodes are satisfied simultaneously. Problem reduction, often referred to as achieving consistency, is a class of techniques for transforming a CSP into problems that are potentially easier to solve. Problem reduction alone does not normally produce solutions, however it can be extremely useful when used in conjunction with search methods. We define some basic notions of consistency [Tsang, 1993] below.

Node-Consistency of a node : A node is node-consistent if and only if all labels in its domain satisfy all unary constraints on that node.

Arc-Consistency of an arc : An arc (i, j) is arc-consistent if and only if nodes i and j are node-consistent and for any label in the domain of node i, there exists a label in the domain of node j such that all binary constraints on the two nodes are satisfied.

Consistency of a graph : A graph is said to be **node-** or **arc-consistent** if and only if every node or arc respectively is node- or arc-consistent.

Multiple unary constraints on a node can be equivalently represented by a single unary constraint that is their conjunction. The same technique can be applied to represent multiple binary constraints between two nodes as a single binary constraint between them. The graph in Fig. 4.1 is node consistent as no unary constraints have been specified. However, it is not arc-consistent.

We now extend the notion of arc-consistency for non-binary constraints. Instead of having constraints on pairs of nodes defining edges in a constraint graph, we have constraints over n nodes that define the **hyperedges** in a **hypergraph** . We define such constraints as **n-ary** constraints.

A constraint is a relation $R_{i,j,\ldots k}$, specifying admissible labels for the nodes $i, j, \ldots k$ over this constraint. A labeling is *arc-consistent* if and only if :

$$\forall v \in V, \forall d_v \in D_v, \forall R_{i,j,\ldots k} \text{ constraining } v,$$
$$\forall i, j, \ldots k, \exists d_i, d_j, \ldots d_k \text{ such that}$$
$$R_{i,j,\ldots k}(d_i, d_j, \ldots d_k) \text{ is true}$$

This means that for every label in the domain of every node i, there exist labels in the domains of nodes $j, \ldots k$ such that every *n-ary* constraint on these nodes is satisfied.

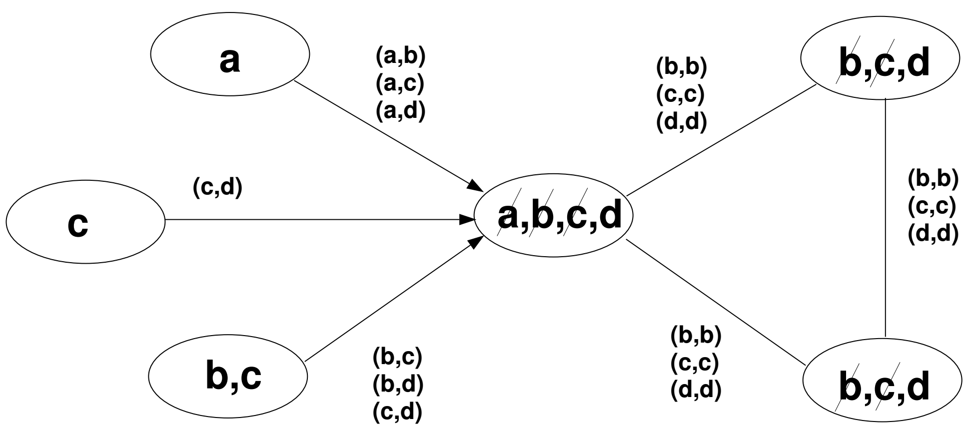


Figure 4.2: *The CSP of fig.1 after application of an arc-consistency algorithm*

4.1.2 Optimal Arc-Consistency

We briefly describe the optimal binary arc-consistency algorithm AC4 [Mohr and Henderson, 1986]. Arc-consistency is intuitively based on the notion of *support*. A label at a node is supported by a label at another node if they satisfy the constraint between the two nodes. A label b for a node i is viable if there exists at least one label at each of the other nodes j (connected to it by a constraint) that supports it. In Fig4.1, label a at node 1 is supported by labels *b,c and d* at node 4 over the constraint between nodes 1 and 4. However, *a,b and c* at node 4 do not have any support over the constraint with node 3. Only *d* at node 4 is supported (by *c*) over that constraint. If there exists a node j where no label satisfies the required relation with b at node i, then b can be removed as a potential label for node i. In Fig.4.1, *a* at node 4 is such a label since no labels at node 1 satisfy the constraint with node 4. Fig.4.2 shows the state of the CSP after achieving arc-consistency. AC4 makes the notion of support explicit by assigning a counter to each arc-label pair. These pairs are denoted by $[(i,j),b]$ and indicate the arc from i to j with label b at node i. The counters are designated by $counter[(i,j),x]$ and indicate the number of j labels that support the label b for i in the constraint (i,j). In addition, for each label c at node j, the set $S_{jc} = \{(i,b) \mid c$ at node j supports b at node i $\}$ is constructed. If c can no longer be a viable label for node j, then $counter[(i,j),b]$ must be decremented for each b supported by c. A table, M, to keep track of deleted labels and a list *List* to keep track of constraint propagation are the other structures used. *List* is initialized with all values (i,b) such that there exists some j for which $counter[(i,j),b] = 0$. The table M ensures that each node-label pair is considered only once. During the propagation phase, the algorithm takes values (j,c) in *List*, decrements each counter $counter[(i,j),b]$ $\forall$ $(i,b) \in S_{jc}$ and when a $counter[(i,j),b]$ becomes zero, it puts (i,b) in $List$. The algorithm stops when *List* is empty, which means that all values in M have non-empty supports on all the constraints. The upper bound time complexity for AC4 is O(ea^2) where e is the number of arcs and a is the cardinality of the biggest label set.[Mohr and Henderson, 1986]

GAC4 [Mohr and Masini, 1988] works on the same principle as AC4. A constraint C is a represented as a set of n-tuples $((i,a),(j,b),\dots(k,c))$. When a label a has to be removed from the set D_i of possible labels for i, all the n-tuples including the label have to be discarded. If the discarded n-tuple is the last in the constraint C supporting a particular label at a node, then that label is deleted and the effect of the deletion is propagated.

4.1.3 A New HyperArc-Consistency Algorithm: NAC4

We have seen that constraints in GAC4 are specified as sets of tuples of admissible labels, that is the relation is represented extensionally (explicitly). This may not be feasible in certain problem domains where the constraint relations might be intensionally represented and computing the extension might be very expensive or in some cases unnecessary. Intuitively, the algorithm should take advantage of prior label pruning to minimize the computation of the relation's extension.

It has been shown ([Tsang, 1993] pp.14) that any n-ary constraint can be represented as a set of binary constraints *provided* the constraints are represented as a list of acceptable tuples of labels (as in GAC4). A constraint expressed implicitly such as: node v_1 is "brighter-than" node v_2 or node v_1 is "brighter-than" node v_3 cannot be represented equivalently as a set of binary constraints unless we compute and compare the brightness values of all the labels in the domains of nodes v_1 to v_3.

NAC4 is designed for constraints that are defined by implicit relations. The notion of support, as in AC4, is extended in NAC4. A label at a node is supported by a set of labels, one at each of the other nodes involved in the constraint, and must have a minimum support from these label sets over all the constraints involving it.

We now present the algorithm NAC4 :

Algorithm NAC4

Initialization

1 $M := 0$; $S_{ib} := \emptyset$;

2 **for** $r \equiv (i, j, \ldots k) \in E$ **do**

3 **for** $b \in A_i$ **do**

4 **begin**

Total := 0;
for $(c, \ldots d) \in (A_j \times \ldots \times A_k)$ **do**
  if $R(i,b,j,c,\ldots k,d)$ **then**
  begin
  Total := Total + 1;
  $Tuple := \{(j,c), \ldots (k,d)\}$;
    for $(j',c') \in Tuple$ **do**
    Append($S_{j'c'}, ((i,b),r,Tuple)$);
  end;
  if Total = 0 **then**
  $M[i,b] := 1$; $A_i := A_i$ - $\{b\}$;
  else $Counter[r,b]$:= Total;
end;
Initialize List with $\{\ (i,b) \mid M[i,b] = 1\ \}$;

Propagation phase

while List not Empty **do**
begin
choose and remove (j,c) from List;
for $((i,b),r,T) \in S_{jc}$ **do**
  begin
  $Counter[r,b] := Counter[r,b]$ - 1;
  for $(k,d) \in T$ **do** $S_{kd} := S_{kd} - ((i,b),r,T)$;
    if $Counter[r,b] = 0$ **and**
    $M[i,b] = 0$ **then**
    begin
      Append(List,(i,b));
      $M[i,b] := 1$; $A_i := A_i$ - $\{b\}$;
    end;
end;
end

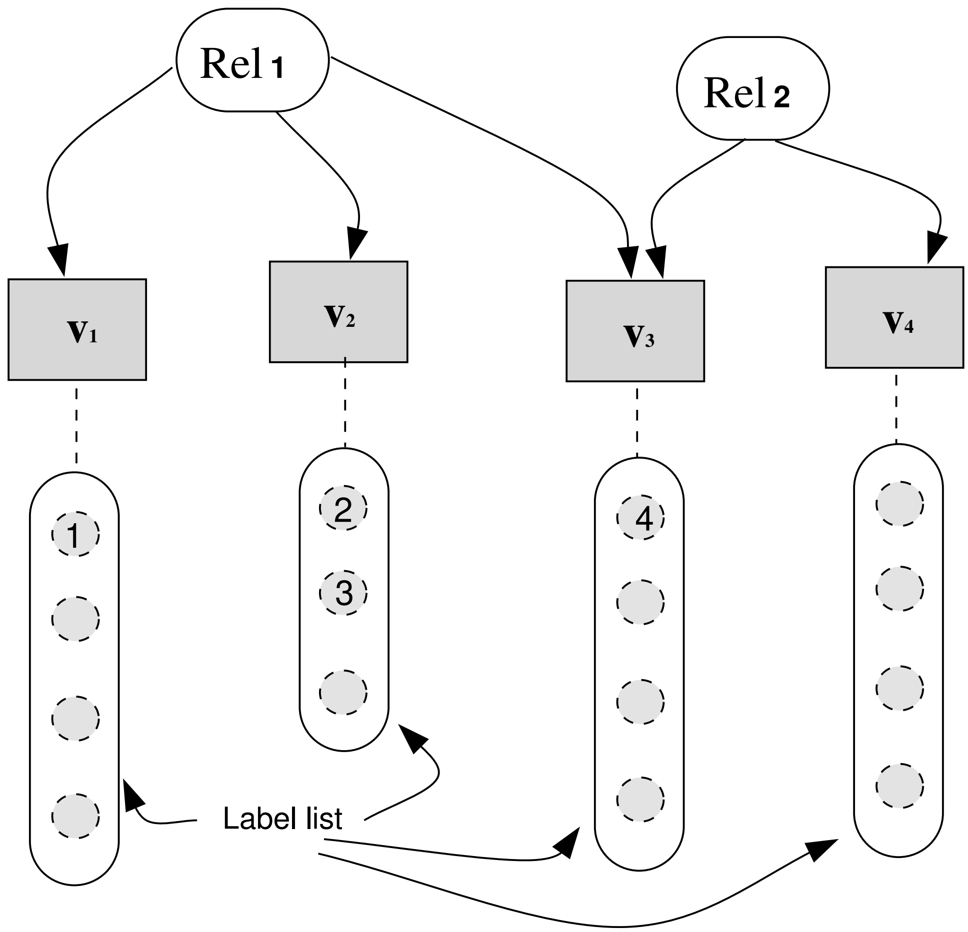


Figure 4.3: *An example of a non-binary CSP*

In the initialization stage, we construct a list of inconsistent labels, stored in M, by calculating the support for labels at each node over each n-ary relation r. The set S_{jc} ={ (i,b)} was used in AC4 to store the fact that c at j supports b at i. We now augment the information stored in S_{jc} to include the rest of nodes involved in the relation. This is achieved by storing the relation index and a pointer to the whole tuple. $S_{jc} = \{ ((i,b),r,Tuple) \mid c$ at j supports b at i in the tuple $Tuple$ of the relation r }. We therefore append the support list of each node in r whenever it supports label b at i . If a label c were to become incompatible for such a node j, then we would have a link to re-assess the support for b at i. Counters are indexed over relations but imply support for the label at the $first$ node of the relation. The propagation phase is altered (line 24) to ensure that "partner" nodes in the same relation r update their support lists. Suppose (j,c) no longer supports (i,b) over relation r, every (k,d) involved in r removes the entry indexed by (i,b) from its support list. Since the counter for b at i over r was incremented only once , multiple decrements are not necessary when $((i,b),r,T)$ is deleted from S_{kd}.

For example, in Fig.4.3, let us suppose that label 1 at node v_1 is supported by labels 2 and 3 at v_2 and label 4 at v_3 over constraint Rel_1.(In the explicit relation notation, Rel_1 would be $\{(1,2,4),(1,3,4)\}$) Now suppose label 4 cannot be assigned to v_3 due to some other constraint (possibly Rel_2) and we delete it from the label list of v_3, then we have to update the support of label 1 at v_1. We decrease the counter and delete the corresponding tuple from the support lists of every node-label pair involved in this tuple. Since there is no other label at v_3 supporting the assignment of label 1 to v_1 over Rel_1, label 1 cannot be a valid label for v_1 and must be deleted. The node-label pairs that it supports will now be examined in turn.

4.1.4 Analysis

A formal correctness proof is not presented here but can be worked out along the lines of the proof for AC4.

Correctness of NAC4

We present the steps for a complete proof of the correctness of NAC4.

1. Our induction hypothesis is that each label deleted from A_i is not admissible for any hyperarc-consistency solution. A label is removed if one of its counters goes to zero, so it has no more corresponding labels at one *hyperarc*. By induction, all the previously removed labels could not belong to any solution, so this one cannot belong to any solution.

2. The result of NAC4 is *hyperarc*-consistent: for all hyperarcs r, for all labels b at node i, we have $Counter[r,b] > 0$. Therefore, each b has corresponding label nodes $j, \ldots k$.

3. From the above two steps, we conclude that NAC4 builds the largest hyperarc-consistent solution because only those labels that provably do not participate in solutions to the original problem are removed.

Complexity

Since the algorithm has been designed for expensive implicit constraints, the domain independent assumption that the constraint verification is of unit cost is invalid. Hence the running time is decided by the application dependent cost of constraint verification. If we were to assume (wrongly) that constraint verification in line 7 of the algorithm is a unit cost operation, the worst case time complexity of the algorithm would be $O(\sum_{n=1}^{m} C_n^m na^n)$, [1] where m is the number of nodes and a is the cardinality of the largest domain. However this complexity is not of much practical importance here since this algorithm is designed for constraints whose cost of evaluation is orders of magnitude greater than a unit step of the algorithm. A precise complexity analysis would estimate the relative cost of verifying each constraint relation and weigh the cost summation correspondingly.

[1] For each of the $n \times C_n^m$ n-ary constraint, at most a^n counters are incremented. Since the counters never become negative, at most $(\sum_{n=1}^{m} C_n^m na^n)$ decrements are performed.

4.1.5 Expensive Constraints

An expensive constraint is one for which checking whether a tuple belongs to the corresponding relation is computationally expensive. Moreover, this implies that the cost of computing the full extension of a relation is proportionally expensive to the sizes of the domains of the participating nodes. A preprocessing step to order the constraints according to their domain specific costs can be added to any consistency algorithm that represents the constraints implicitly.

For example, the initialization phases of AC4 and NAC4 can be modified to select the next least expensive edge. Such a modification does not seem possible in the case of GAC4 since the initialization uses the explicit representation of the constraints to build the elegant data structures that make the algorithm optimal. The standard way to achieve arc-consistency requires that we first achieve node-consistency. However, some unary constraints may be more expensive than other constraints of higher arity. These unary constraints need not be enforced in scenarios where non-unary constraints are much cheaper to evaluate and achieve substantial problem reduction.

4.2 Coupling Interval reasoning with FDCSP

In traditional Constraint Satisfaction Problems, it is generally assumed that the variables, domains and constraints are all pre-computed inputs to a *black-box* constraint satisfaction algorithm. The obvious advantages of such an assumption is the freedom in developing generic constraint solvers and the declarative use of constraint technology. However, it is useful to examine applications of CSPs to evaluate if they can benefit further from *on-the-fly* constraint and domain specification. We consider particular classes of numerically computable discrete CSPs for which the variables' domain generation is costly. The domain generation and its cost are usually application dependent. However, in several numeric discrete CSPs, some general measures of cost can be estimated. We propose to reduce the generation cost of the discrete finite domains by combining interval narrowing techniques with local consistency techniques in discrete domains.

The motivation for this work comes from an application in high-level computer vision[Chopra and Srihari, 1995; Srihari *et al.*, 1996] where the constraint graph consisted of a hypothesis of the contents of an image. The variables of the CSP are physical objects, for instance, buildings or faces, and spatial relations between these objects formed the constraints. They presented an iterative technique called *spatial prediction* that demarcated subregions of the original image as *search regions*. Computer vision algorithms would then be applied selectively on these sub-images to generate potential values for the variables. However, there were several problems with their approach as described later in the Discussion section. We believe our algorithm overcomes those problems and can easily be applied to this application as well as others.

4.2.1 Definitions

Definition 1: A Discrete Constraint Satisfaction Problem (DCSP), (V,D,C,R), involves a set of n **variables**, $V = \{v_1, v_2, \dots v_n\}$, each with an associated **domain** D_i of possible **values**. The search space consists of the Cartesian product of the variables' domains, $D = D_1 \times D_2 \times \dots \times D_n$. A set of **constraints** C is specified over some subsets of these variables, where each constraint C_p involves a subset $\{i_1, \dots, i_q\}$ of V and is labeled by a relation R_p of R, subset of the Cartesian product $D_{i_1} \times \dots \times D_{i_q}$ that specifies which values of the variables are compatible with each other. A **solution** to a DCSP consists of an assignment of values to variables such that all constraints are satisfied.

Various approaches exist for solving discrete CSPs. A common approach is based on problem reduction strategies, and involves achieving local, that is, node or arc, consistency [Mackworth, 1977; Mohr and Henderson, 1986; Mohr and Masini, 1988; Hentenryck *et al.*, 1992]. This reduces the amount of backtracking required for global consistency.

Definition 2: An interval is informally defined as a number pair: $X = [a : b] = \{x | a \leq x \leq b\}$

Definition 3: We define a set G of functions g_i called domain generators. The function g_i associated with variable v_i maps an interval to a discrete finite domain for that variable or equivalently $g_i(I_i) \rightarrow D_i$.

These functions are typically application dependent and we focus on those applications where these generators are computationally expensive. We present below one measure of the cost associated with generating the domains. For example, in the vision application mentioned in the introduction, the object location tools would be the generators.

Definition 4: Since our objective is to reduce the cost of generating the domains of the variables, let p_i be the cost associated with generating the domain of variable v_i. We define *cost* to be a function of g_i and I_i implying that $p_i = cost(g_i, I_i)$. The total cost of generating the domains is $\mathcal{P} = \sum_{i=1}^{n} p_i$ and our objective is $min(\mathcal{P})$. If we assume that *cost* is directly proportional to $|I|$, then $max(p_i) = cost(g_i, max(|I_i|))$. In solving the CSP, assuming that a solution exists, all the domains will be generated. Note that in a traditional discrete finite domain CSP, all domains are generated initially with intervals being their maximum respective values, leading to a cost $\mathcal{P} = \sum max(p_i)$. Any algorithm that reduces the size of the intervals (assuming that the generator functions cannot be changed) achieves a reduction in $\mathcal{P}$.

Definition 5: We define our new heterogeneous CSP (V, G, C, P, V', D', C') in the following manner: The discrete finite-domain constituent (FDCSP) is a triple (V, G, C) where V is a set of n variables $\{v_1, v_2, \ldots, v_n\}$, G is a set of generators g_i that generate domains D_i for the n variables and C is a set of constraints where C_p involves a subset $\{i_1, \ldots, i_q\}$ of V and is labeled by an *implicit* relation R_p of R, subset of that specifies which values of the variables are compatible with each other.

The other constituent of our heterogeneous CSP is an interval CSP (ICSP) defined by the triple (V', D', C'). For each variable $v \in V$ there exists a variable $v' \in V'$. D' is a set of interval domains where D'_i denotes the domain for the interval variable v'_i. For example, if D_i, the domain of the FDCSP variable v_i is a set of reals, then the domain D'_i of the corresponding interval variable v'_i is an interval over reals, I_i as described by Definition 2. The set P comprises of the domain generators p_i defined above in Definition 3. The set of constraints C' consists of implicit relations corresponding to the relations over the FD variables. The semantics of these relations is the arena of interval arithmetic and is outlined in Appendix A.

Definition 6: We define **hull**$(S) = [min(S) : max(S)]$ where S is a finite set of values from a totally ordered set.

4.2.2 Algorithm HetCSP

We now present the sketch of an algorithm to reduce the cost of generating the domains of the FDCSP in an incremental manner.

Algorithm HetCSP

A0: Create an incremental FDCSP ($V'' = NULL$,
$D'' = NULL, C'' = NULL$)

A1: Ensure local consistency in ICSP (V', D', C')

A2: Choose a variable $v_i \in V - V''$ such that
$p_i = min[p(g_i, I_i)]$

A3: Generate the finite domain D''_i and update the
FDCSP such that $V'' = V'' \cup \{v_i\}$,
$D'' = D'' \cup \{D''_i\}$ and $C'' = C'' \cup \bigcup_{v_j \in V''}$

$\{c''|c'' = \rho(x_1,\ldots,x_i,\ldots,x_k) \wedge x_1,\ldots,x_k \in V''\}$

A4: Ensure local consistency in the FDCSP

A5: $\forall v_i \in V''$ compute domains D'_i of corresponding variables v'_i such that $D'_i = \mathbf{hull}(D''_i)$

A6: If $V - V'' = NULL$ quit, else **goto A1**.

Local consistency in the ICSP is ensured by one of several interval narrowing techniques[Siskind and McAllester, 1993b; Benhamou *et al.*, 1994; Benhamou and Older, 1993; Faltings, 1994; Sam-Haroud and Faltings, 1996]. Local consistency algorithms in the FDCSP is ensured to different degrees by arc- and path- consistency algorithms[Mackworth, 1977; Mohr and Henderson, 1986; Hentenryck *et al.*, 1992]. This algorithm falls under the class of consistency or problem reduction algorithms and requires a final enumeration procedure (e.g. chronological backtracking) for completeness.

Consider the example CSP shown in Figure 4.4. Frame 1 describes the initial state of the CSP with 3 finite domain variables A,B and C and their corresponding interval variables A',B' and C'. The initial interval from which the discrete finite values for A,B and C are chosen is [1:100]. The empty curly braces denote the finite domains.

Frame 2 shows the state of the CSP once the domain of A is generated. The interval is accordingly recomputed (Step A5) to [20:30] and this effect is propagated over the interval variables by narrowing B' and C' to [20:100] (Step A1).

In frame 3, the domain of variable C was generated and set to $\{22,60,80\}$. The interval C' is according narrowed to [22:80] and in the propagation step, B' is narrowed to [20:80].

Finally, in frame 4, the domain of B is generated. It is set to $\{25\}$. Arc-consistency applied to the FDCSP removes 30 from the domain of A and 22 from the domain of C. The interval domains are narrowed to reflect the final state. Since only local consistency was ensured in the incremental stages, a final pass to achieve global consistency has to be

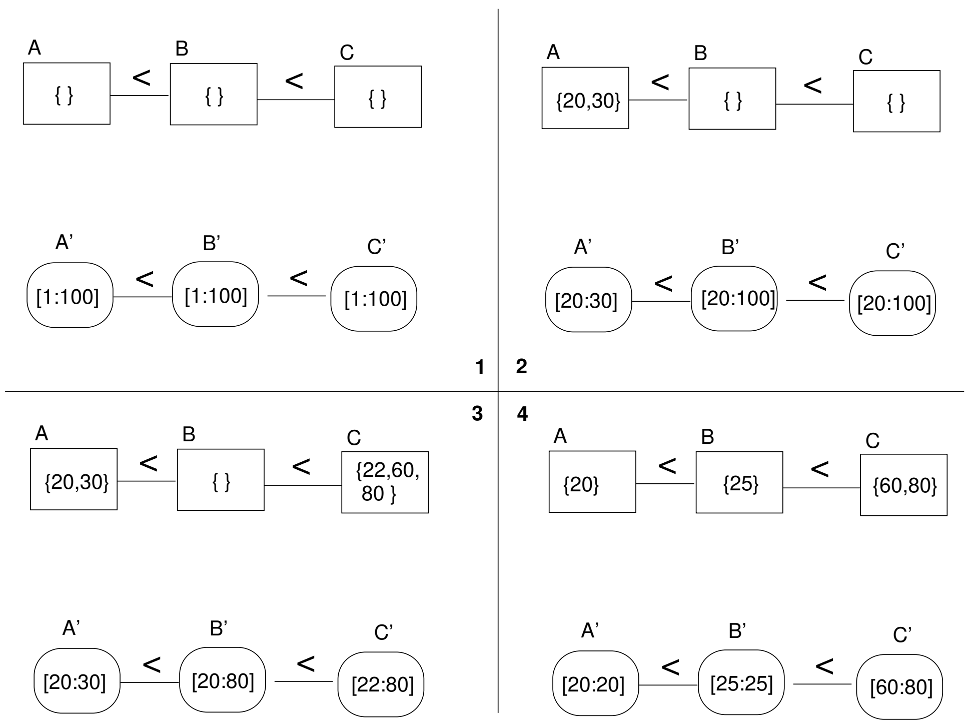


Figure 4.4: *Example: Frames 1 to 4 outline the state of the CSP with 3 finite domain variables* A, B *and* C *and their corresponding interval variables* A', B' *and* C'

performed and solutions enumerated.

4.2.3 Analysis

It is easy to see that the variable selection in Step A2 makes HetCSP a greedy algorithm. Several different heuristics could be substituted here and need to be investigated further. The focus of this algorithm is on invoking the domain generators with the minimum cost. We chose the greedy approach as it is the simplest and we make no claims about optimality.

Correctness

The correctness of HetCSP is informally proven as follows:

1. Routines for local consistency in ICSP and FDCSP exist and are correct. In other words, the local consistency routines always terminate and filter the domains (interval and discrete-finite) such that no value that participates in a correct solution is removed.

2. Discrete values that are deleted from the domains of variables are done so only by the (correct) local consistency algorithm for FDCSPs.

3. Discrete values for a variable that are not generated as a result of the corresponding interval being narrowed cannot potentially participate in a correct solution. They would be in conflict with generated values at other nodes that caused this particular interval to be narrowed.

4. HetCSP terminates trivially since it iterates n times over steps (A1-A6) that have only terminating routines.

5. HetCSP is correct since it terminates and deletes only those values from the domains of the FD variables that cannot satisfy the constraints and participate in a correct solution.

4.2.4 Discussion

A different viewpoint on domain generation is to consider a discrete constraint satisfaction problem where the domains of the variables are extremely large and the domain generators are unary constraints on these variables. However, the cost of evaluating the unary constraint is directly proportional to some positive power of the size of the domain. This view treats the problem as a discrete CSP only, whereas we view it as a combination of interval and discrete CSPs, where interval narrowing on numerical domains is computationally less expensive.

A potential problem in applying this algorithm may lie in the estimation of the domain generation cost. We make the assumption that the discrete numeric values that constitute the finite domains of the variables are generated from some numeric interval (as defined in Definition 3). We further assume that the generation cost is application specific, though proportional to some positive power of the size of the interval. It is this cost that we attempt to minimize. The observation that a single interval narrowing step can drastically reduce the interval representing the finite domain is precisely what helps reduce the (costlier application-specific) domain generation cost. Interval narrowing techniques have been used in association with finite domain CSPs in a layout generation application [Hower, 1996]. The objective there was to compute all layout possibilities in a semi-automatic system where a user places non-overlapping objects in a two dimensional frame. The application is an excellent example of the benefits of reformulating discrete finite domain CSPs using interval constraints. However, domain generation was not an issue. Can partial results be used to guide search? In the past, backtrack search algorithms have incorporated domain independent techniques and heuristics (Chapters 5-7 [Tsang, 1993]) to guide the search for solutions using partial results. However, all the above techniques are for CSPs where the domains and constraints are pre-computed or can be computed with unit cost. We believe our algorithm is the first to consider domain generation costs as minimization criteria for CSPs.

Chopra and Srihari [Chopra and Srihari, 1995; Srihari *et al.*, 1996] explored domain generation in a computer vision application where the constraint graph consisted of a hypothesis of the contents of an image. For example, the variables were buildings or faces and spatial relations between these objects formed the constraints. They presented a technique called *spatial prediction* that demarcated subregions of the original image as *search regions*. Object location algorithms would then be applied selectively on these sub-images to generate potential values for the variables. The problems with the above mentioned approach are:

1. Incomplete propagation: When *spatial prediction* does take place, it changes the search regions of **only** the variables adjacent to it in the constraint graph. The change in the search regions is not propagated to other variables' search regions.

2. Limited activation of propagation: Spatial prediction takes place only when a variable is uniquely labeled. If multiple values are generated for a variable, and partial satisfaction of the constraint graph does not lead to unique labeling for any variable, no spatial prediction takes place.

3. Domain dependence of algorithm: The above algorithm uses data structures and spatial terms particular to the high level vision problem addressed.

Algorithm HetCSP addresses the three weaknesses described above. Propagation of the interval bounds takes place in all interval variables till local consistency is achieved in the ICSP. Step A5 computes new interval bounds and propagation is activated every time a domain changes; not just when there is a single value in the domain. The presented algorithm is independent of the application domain and a mapping from the vision application to HetCSP is presented in section 3.4.1 on page 29.

4.3 Summary

In this chapter we have provided an algorithm that handles certain types of n-ary constraints and that doesn't make the assumptions of GAC4 regarding representation of the constraints. We believe the assumption of unit time complexity assigned to checking a constraint is invalid and in some cases the constraint check time easily outweighs the time spent in the rest of the algorithm. Hence the complexity of the algorithm is dependent upon the problem domain that defines the constraint checking time. Domain specific experiments or simulations need to be performed to validate our belief in the better performance of NAC4 with implicit relations over GAC4 with explicit relations.

We have also presented an innovative application of interval arithmetic in constraint satisfaction. Though interval constraint satisfaction is an extensive research area, we believe this to be the first attempt at using interval arithmetic to reduce the domain generation cost of finite domain CSPs. A new algorithm has been proposed here for the above. Illustration with a simple example, analysis and implementation of the algorithm have also been detailed here. We believe the algorithm will prove useful in the application domain that motivated this research and also believe that the presentation here is general enough for application in other areas.

Chapter 5

Applications

There are several applications which are enabled by collateral-based vision. These include diagram understanding, indexing and retrieval from text/image databases and aerial image analysis. One such application is assisting an image analyst in image understanding. Textual information can be used in mapping an image onto a predefined site model, e.g., "in the center of the image is hangar A". Another application is robot vision, specifically, using natural language to help a robot understand a scene. In diagram understanding applications, text accompanying the diagram can be used in (i) converting from raster to vector representations and (ii) provide the ability to reason with the diagram. As a simple example, if the caption reads "the x-axis represents the age in years of subjects", it restricts the OCR output for x-axis entries to numbers representing realistic values for humans. Finally, an important application is in the automatic indexing and content-based retrieval from text/image databases. It would be possible to process queries such as "find an outdoor picture with a man standing next to a red car" if collateral based vision techniques were employed.

In this chapter we present three applications that were implemented to demonstrate the potential of collateral based vision. We present them in chronological order to give the reader an impression of how the architecture of the system evolved. The three systems are

1. PICTION : A caption based face identification system.
2. Show&Tell : A semi-automated image annotation and retrieval system.
3. FEVAHR-Spatial : An autonomous mobile robot with foveal vision and a pseudo-natural language interface.

5.1 PICTION

PICTION was a system developed by Srihari [Srihari, 1991] that identified human faces in newspaper photographs based on information contained in the associated caption. Several aspects have since been modified [Srihari *et al.*, 1994; Chopra and Srihari, 1995]. The modifications to the representation and control aspects have been :

1. A *uniform* representation for domain and picture specific spatial and non-spatial constraints,
2. Improved spatial reasoning capabilities,
3. Constrained search, that is, employing spatial reasoning to use partial interpretation results in guiding the application of the vision processing tools.
4. Generalized problem representation in an object oriented manner to deal with multiple object types and aggregation.

In the first subsection, we discuss the current version of PICTION in greater detail. The modifications to the architecture through the various versions of PICTION helped us in identifying the domain independent aspects of the architecture. We therefore present the chronological evolution of the control architecture in the next subsection.

5.1.1 Overview

We describe the system *PICTION* [Srihari *et al.*, 1994], that identifies human faces in newspaper photographs based on information contained in the associated caption. Most front-page newspaper photographs tend to have captions that are factual and descriptive, qualities required for this task. PICTION is noteworthy since it provides a computationally less expensive alternative to traditional methods of face recognition in situations where pictures are accompanied by descriptive text. Traditional methods employ model-matching techniques and thus require that face models be present for all people, for them to be identified by the system.

The four main components of the model are as follows: (i) a natural-language processing (NLP) module, (ii) an image understanding (IU) module, (iii) a high level control module, and (iv) an integrated language/vision knowledge base.

The NLP module processes the caption text and expresses the result as a set of constraints. **Spatial Constraints** are geometric or topological constraints, such as left-of, above, between, inside, etc. They can be binary or n-ary, and describe inter-object relationships. They are typically used to identify/disambiguate among objects but can also be used to locate them. **Characteristic Constraints** are those which describe properties of objects. Examples include gender and hair color. Finally, **Contextual Constraints** are those which describe the setting of the picture, and the predicted objects in it. For example, the people present (explicitly mentioned in the caption), whether it is an outdoor scene, and general scene context (apartment, airport, etc).

The IU module performs two basic functions, the location and segmentation of objects,

Caption:"The All-WNY boys volleyball team surrounds the coach of the year, Frontier's Kevin Starr.Top row, from left, are Eden's Brian Heffernan, Lake Shore's Tom Kait, Sweet Home's Greg Gegenfurtner, Kenmore West's George Beiter. Bottom row, from left, are Eden's David Nagowski and Orchard Park's Rich Bland."

Figure 5.1: *Original Image and Accompanying Caption*

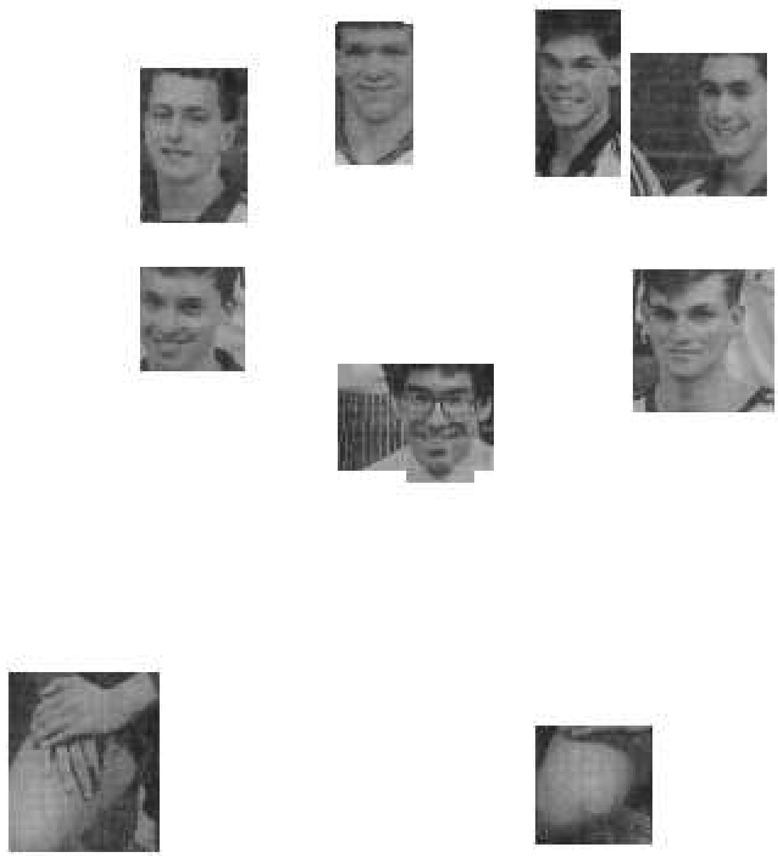


Figure 5.2: *Output of face locator for image in Fig. 5.1*

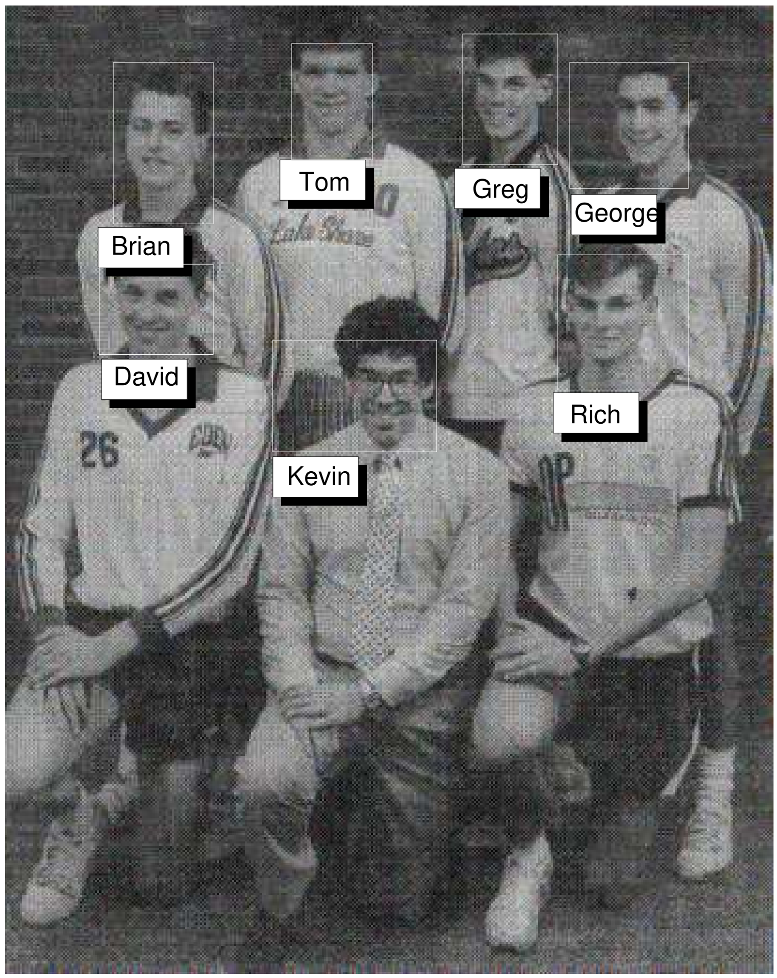


Figure 5.3: *Faces labeled after satisfying spatial constraints*

```
CONTEXTUAL:
     in-picture(name: Kevin Starr; class: human)
     in-picture(name: Brian Heffernan; class: human)

CHARACTERISTIC:
     gender(name: Brian Heffernan, male)
     gender(name: Tom Kait, male)

SPATIAL:
     surrounded-by(Kevin Starr,
                  (Brian Heffernan, Tom Kait ...))
     strict-left-of(Brian Heffernan, Tom Kait)
     below(David Nagowski, Brian Heffernan)
     in-row(number: 4;
             names: (Brian Heffernan, Tom Kait ... ))
     in-row(number: 2;
             names: (David Nagowski, Rich Bland ))
```

Figure 5.4: *A subset of the constraints generated from the caption of Figure 1.*

and the extraction of visual properties. The face-location module[Govindaraju *et al.*, 1992] is an object locator which uses edge contours as basic features, and a "springs and templates" model in a multi-resolution framework. A line detector [Burns *et al.*, 1984] is being used to construct simple detectors for 2D rectilinear objects.

Figure 5.1 is an example of a digitized newspaper photograph and the accompanying caption. The task is to correctly identify each of the seven people mentioned in the caption. Information from the caption (in the form of a constraint graph) is used to locate and label faces. Figure 5.4 illustrates some of the constraints which were derived from the caption. The number of faces and the fact that they are aligned in rows is used by a program which locates faces in the image. Identification of the faces revolved around the correct interpretation of the word 'surround'. We have interpreted it as an n-ary constraint involving minimum distances from a given candidate to the rest. The simultaneous satisfaction of this and the remaining constraints resulted in the final, correct labeling.

PICTION has been implemented in LOOM[ISX, 1991a]. which is a high level programming language and environment for constructing knowledge based systems and is based on the KL/1 family of semantic networks. It provides an object-oriented model specification language, classification based inference and multiple programming paradigms in a framework of query based assertion and retrieval. The visual routines have been implemented in C; the system runs on a Sun Sparcstation.

5.1.2 Control in PICTION

We present a control strategy which exploits (i) a confident hypothesis of the image contents and (ii) all levels of contextual information to aid the visual processing.

Our approach is to select a set of object classes of interest which we would like to locate and identify in images. For PICTION this set has been restricted to one object class, the human face. However, recognizing additional classes of objects (like hats and boxes) may greatly assist us in identifying faces.

Constraint satisfaction provides a single framework for incorporating spatial, characteristic, contextual and other general domain constraints without having to over-specify control information. However the location of objects is still performed at the IU level thereby allowing existing object detectors to be integrated into the overall model.

The Single Object Class Problem: Labeling Faces

For the single object class version of PICTION, the overall control can be summarized in a few steps :

- Constraint Graph Generation (hypothesis generation: NLP module)
- Face Candidate Generation (candidate generation : IU module)

- Consistent Labeling (verification and recognition)

Knowledge of the domain allows us to make simplifying assumptions for spatial reasoning. The assumptions concern spatial representations of objects, frames of reference, scale and rotation invariance. Spatial reasoning in this domain involves verification of predicates corresponding to spatial relations. [Burhans *et al.*, 1995]

Constraint Satisfaction as a Control Paradigm

In the previous chapter, we have already described how we employ constraint satisfaction as the basis for collateral based vision.

In PICTION, each variable v_i corresponds to an object or person hypothesis, i.e., object or person predicted to be in the picture by the constraint generator. The domain D_i for each variable is initialized to the set of all **candidates** $\{c_{i_1}, \ldots, c_{i_m}\}$ generated by a call to the face locator.

There are 3 types of constraints used in the labeling process. **Unary Constraints** typically correspond to characteristic constraints (e.g., gender, color of hair, etc.). **Binary Constraints** typically correspond to spatial constraints (e.g. left-of, above, etc.). ***N*-ary Constraints** correspond to rules obtained from the domain *a priori* but applied in a discretionary manner by the constraint satisfier, or n-ary spatial constraints like the 3-ary constraint *between*. An example of a domain dependent n-ary constraint is one which examines height relationships between people in the same row. It favors those solutions where the vertical positions of faces do not differ significantly and where there is a minimal amount of horizontal overlap.

Constraint satisfaction in a schema-based methodology has been used in computational vision, eg. the MAPSEE system[Mulder *et al.*, 1988]. Their approach is that of data driven scene labeling where image features are nodes and possible interpretations form the sets of labels. Our application of constraint satisfaction is in a goal-driven image interpretation system where a strong scene hypothesis forms the set of nodes, and located objects (features) form the sets of labels. The constraints in our system are picture specific as well as domain dependent.

HyperArc Consistency and Expensive Constraints

There are two key differences between the statement of PICTION's constraint satisfaction problem and a traditional CSP. Traditionally unary constraints are satisfied before higher order constraints. Knowing that certain binary constraints are cheaper to evaluate than unary constraints motivated a new algorithm where constraints are applied based on cost and reliability measures. Additionally, the algorithm can handle n-ary constraints without explicitly computing the complete corresponding n-ary relation [Chopra *et al.*, 1995].

Multiple Object Classes: Generalizing PICTION

So far we have seen a control strategy that exploits collateral information to identify objects of a single class in an image. This needs to be extended to more general cases involving multiple object classes. Apart from making it applicable to domains with more than one object class of interest, this capability may be required even if there is a single object class of interest. We borrow here the terminology of defining a *target* vocabulary and a *recognition* vocabulary[Strat, 1992]. Even if the target vocabulary consists of a single object class, its recognition is greatly enhanced by the development of a large recognition vocabulary. For the example in figure 5.5, the recognition vocabulary would need to contain picture frames apart from faces. Though picture frames may be specific to the newspaper photograph domain, the idea is applicable to other domains with a specific set of interesting objects and a larger set of recognizable objects.

We currently assume an unstructured collection of object locators associated with object classes is available. We are designing a hierarchy of object locators that is linked to our semantic lexicon. This lexicon is organized as a comprehensive ontology of object classes. This will enable us to recognize more object classes using the same primitive object locators. We also assume a corresponding set of image processing tools to verify and compare predetermined attributes of instances of these object classes. Domain specific knowledge allows us to make assumptions regarding object representations and frame of reference for spatial reasoning. Given these assumptions, a naive approach would be:

- Constraint Graph Generation (same as before, constraints are between objects of same/different classes)
- All Object Classes Candidate Generation (apply object locators for all object classes mentioned in the caption)
- Consistent Labeling (verification and recognition)

The above scheme is wasteful, since all object classes needn't be identified for the identification of the target vocabulary. It does not exploit locative constraints to reduce the search space and improve the performance of the object locators (candidate generators). The caption of the image in figure 5.5 is *"Sharon Bottoms and April Wade hold a picture of Tyler Doustou"*[1]. Assuming picture frames (portraits) are in our recognition vocabulary, the relevant constraints that are generated :

```
included-in(Tyler Doustou, PicFrame1);
left-of(Sharon Bottoms, April Wade)
```

[1]Obtained with permission from the Buffalo News

Using the naive approach, labeling is impeded by the failure of the face locator to find Tyler's face. The picture frame also cannot be labeled as no candidate frame object satisfies the *included-in* relation with a face candidate.

Constrained Search

To overcome the above limitations, we choose a more complex approach with some semblance to planning. The idea of using partial interpretation results to localize the search for related objects has been used in several vision systems. For example, constrained search is used by Selfridge [1982] to guide the search for building shadows, but with minimal spatial reasoning. In Section 5.1.2 we classified constraints as contextual, spatial and characteristic. When the control algorithm employs any of these constraints to disambiguate candidates, we call them *verification* constraints and when it employs them to locate candidates for an object, we term them *locative* constraints. We assume cost and reliability measures for our object locators and attribute verifiers are available.

The control algorithm loops over three stages:

- **Decision Stage** A complex cost/utility function is employed to decide which object class to detect *next*, given the current state of the constraint graph and labelings. The factors taken into account for this function are the tool cost, reliability, number of objects to be found, size of search region, number of constraints between this class and the objects of interest and the level of interest in this particular class of objects. Some measures are class specific (tool cost), others are instance specific (number of objects) and some are dynamic (size of search region).
- **Labeling Stage:** Candidates for objects of chosen group are located using the collateral information available. These objects, candidates and the constraint relations

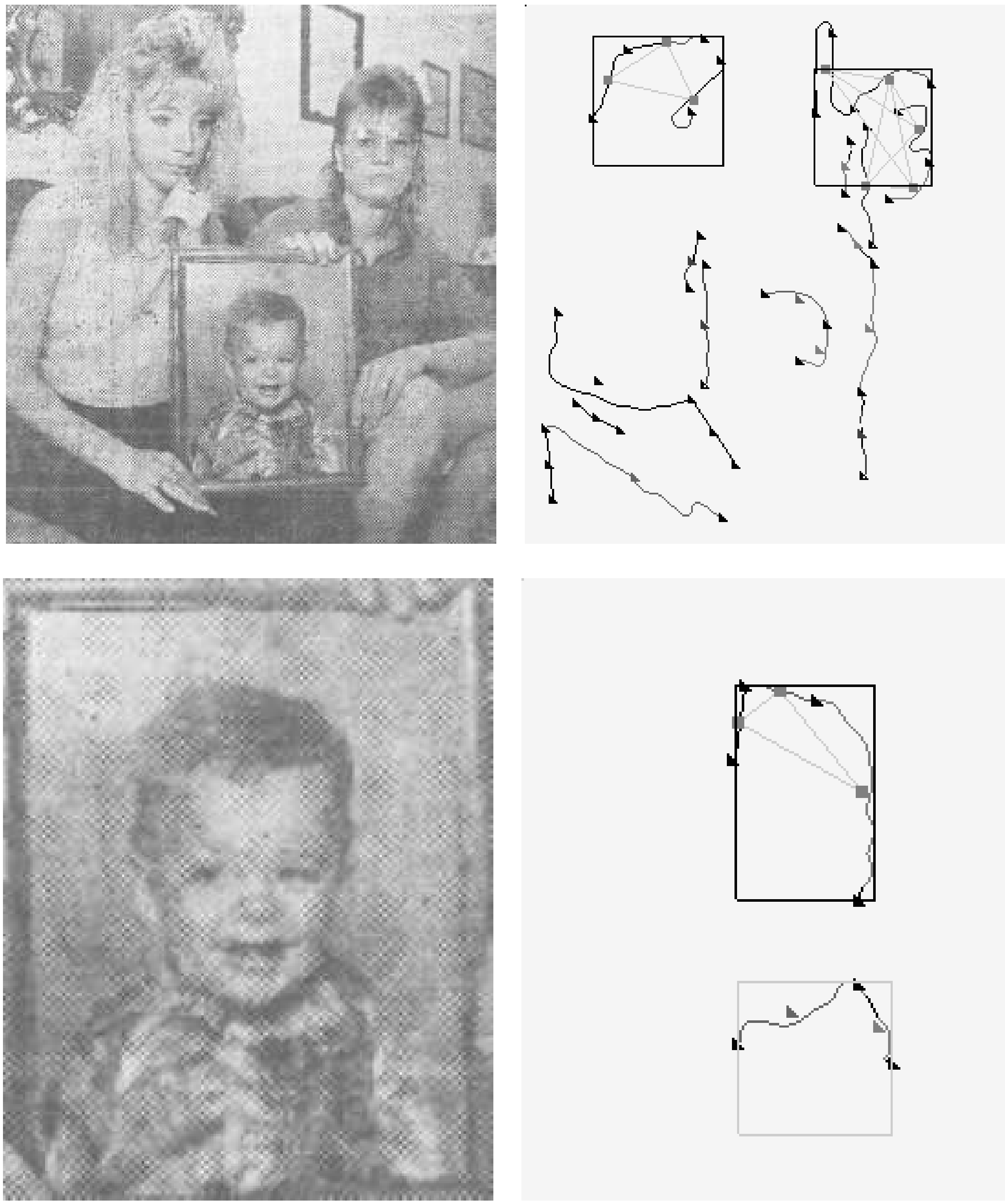


Figure 5.5: *Illustration of constrained search.* From top left: *original image and output of face locator for it; constrained search region and output of face locator for it.*

between the objects are inserted in a constraint graph representing the partial interpretation. Constraint satisfaction techniques are applied to uniquely label the objects. Constraints between objects in the new group and between objects in prior groups and the new ones are satisfied to increment the partial interpretation of the image.

- **Propagation Stage:** Any object which has been labeled uniquely in the previous module potentially constrains the search region for those objects involved in a spatial relation with it. The propagation module computes the search region using the labeled object and the type of spatial relation. So far, this *spatial prediction* has been implemented for binary spatial constraints.

The effect of locating objects of a particular class can then be summarized as (i) potential identification of objects already located, (ii) labeling and partial identification of current class objects and (iii) constraining and prioritizing the search for some other objects.

As we have mentioned, an object locator is parameterized by the number of objects of that particular class. Since the actual routines for object-location are not completely accurate, we might obtain too few or too many object candidates. Information from the spatial prediction stage is used at this point to attempt a solution. The object locator routines are called again on the constrained search region with relaxed parameters if there were initially too few candidates. If there were extra candidates, the information is used to order the candidates and the top choices are picked.

Traditionally, planning in vision has been deliberative, i.e. a plan for the entire task is constructed prior to object location based on probabilistic measures[Garvey, 1976; Ballard *et al.*, 1978]. We employ planning *reactively* to choose amongst locators using a-priori knowledge as well as the observable situation to minimize computation and maximize accuracy.

In addition to verifying spatial relations between object candidates, the **spatial reasoning** module performs the task of spatial *prediction*. In the process of interpretation, any object that becomes uniquely labeled can serve as a potential *reference* object. Spatial relations between this reference object and other objects *not yet located* are used to generate search areas for those objects. The simplifying assumption of a known fixed reference frame makes this task relatively simple. We did not need a complex spatial representation and reasoning framework in the limited domain of PICTION. However, the control architecture is modular and richer object representations and associated reasoning mechanisms can easily be incorporated.

An Example As an example of how picture-specific constraints can enable a constrained search for objects, consider Figure 5.5. The object schema for "picture frames" yields the information that it contains a frame/border and an interior and that the border is typically a rectangle or an oval. The other object schema is for faces. Based on this, the control module invokes the following operations:

- Since rectangles and ovals are easier to find than faces, the decision stage chooses to look for picture-frame objects first. It finds one candidate, a working constraint graph containing a single node is created and labeling is successful. Next, the search region for Tyler's face is constrained because of the relation *included-in*. Control loops back to the decision stage where the next best class (the only one left in this example) is chosen.

- The object locator for faces is now called to find 3 faces. The schema contains information that one face must be in a specified constrained area. Fig. 5.5 (2nd image) shows the output of the face locator. Since the number of candidates (two) is less than the known number of faces, the locator must be called again. Since neither of the candidates found is in the constrained region, the missing face must be in it, and

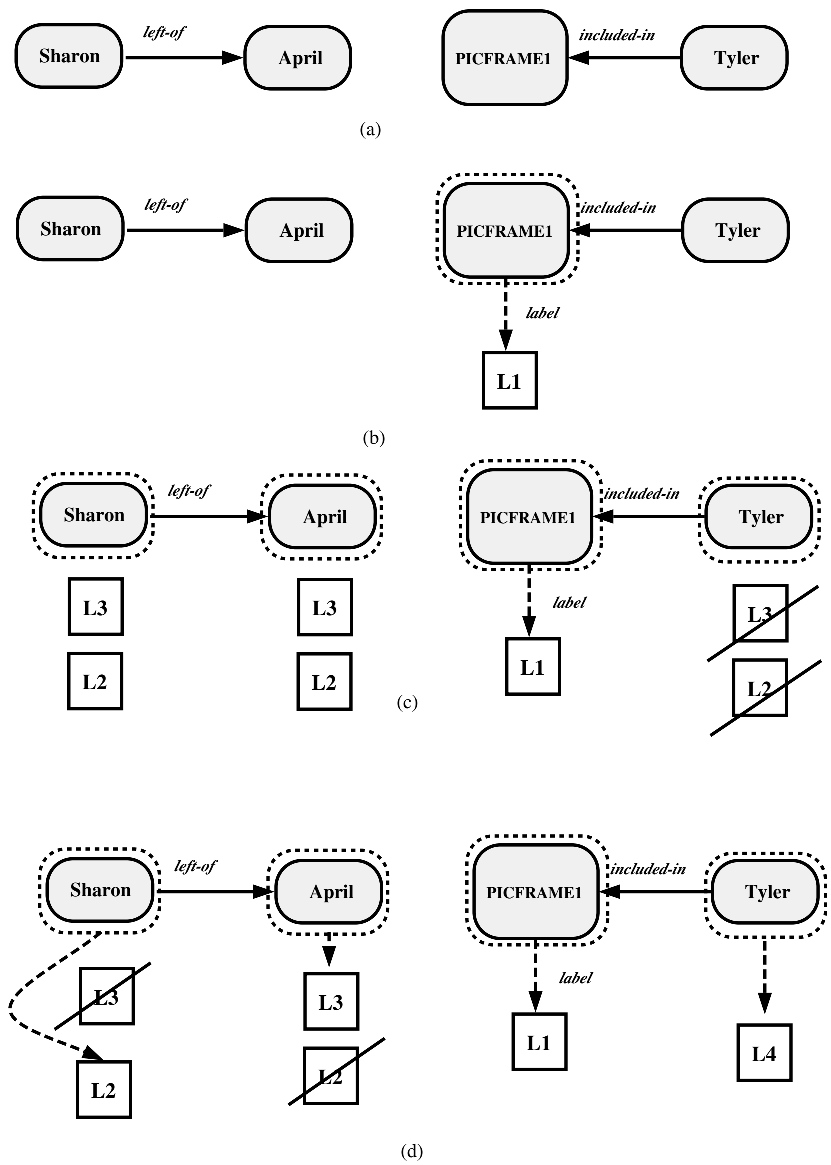


Figure 5.6: *Working Constraint Graph: (a) Complete scene constraint graph (b) PicFrame nodes are chosen in working graph, candidates generated and labeled (c) Face nodes included in w/graph, 2 candidates generated, no unique labeling (d) constrained search yields new candidate for Tyler node, leading to a successful labeling.*

the locator is called on this sub-image. The last image on the top of the page shows the output. The rectangle in black is the candidate returned by the face-locator and the grey rectangle is a weaker choice not returned in the first pass. The third face candidate is thus located.

- The constraint graph is now augmented with 3 more nodes (representing the 3 persons) and initialized with the new labels and 2 constraints (ref. to page 68) resulting in a successful labeling.

5.2 Show & Tell

Another domain where the notion of context based vision can be demonstrated is aerial image interpretation. The collateral information is provided by an image analyst in the form of a high level description of the contents. This information is used in the interpretation and indexing of the images. The annotated images are then stored in a knowledge base for content based retrieval of images and entities depicted in the images.

5.2.1 Overview

A prototype system consisting of the following modules has currently been implemented : (i) A user interface employing speech recognition, natural language processing, deictic input and visual feedback, (ii) an image interpretation module that employs the collateral provided by the analyst (and interpreted by the interface) to guide the object location and recognition tools. (iii) a simple query module with limited spatial reasoning capabilities. There are two typical phases of operation, the annotation phase where an analyst examines a new image and the query phase where retrieval is performed based on natural language and simple gesture based queries. Figure 5.7 illustrates the functional architecture of **Show&Tell**. During the annotation phase, an analyst first invokes a semi-automated road detection algorithm that displays a network of roads segments. This network along with the image boundaries forms a planar graph. A graph algorithm finds and labels the *faces* [Harary, 1971] of this planar graph. Each internal face represents a `physical area` demarcated by roads and could serve as a **landmark**. These landmarks are then used by the analyst in describing the contents of the scene. The analyst's spoken annotations are recognized using commercial speech recognition software and a language understanding module generates a hypothesis of the contents of the scene. The image interpretation module that we describe below uses the scene hypothesis to locate and label entities in the

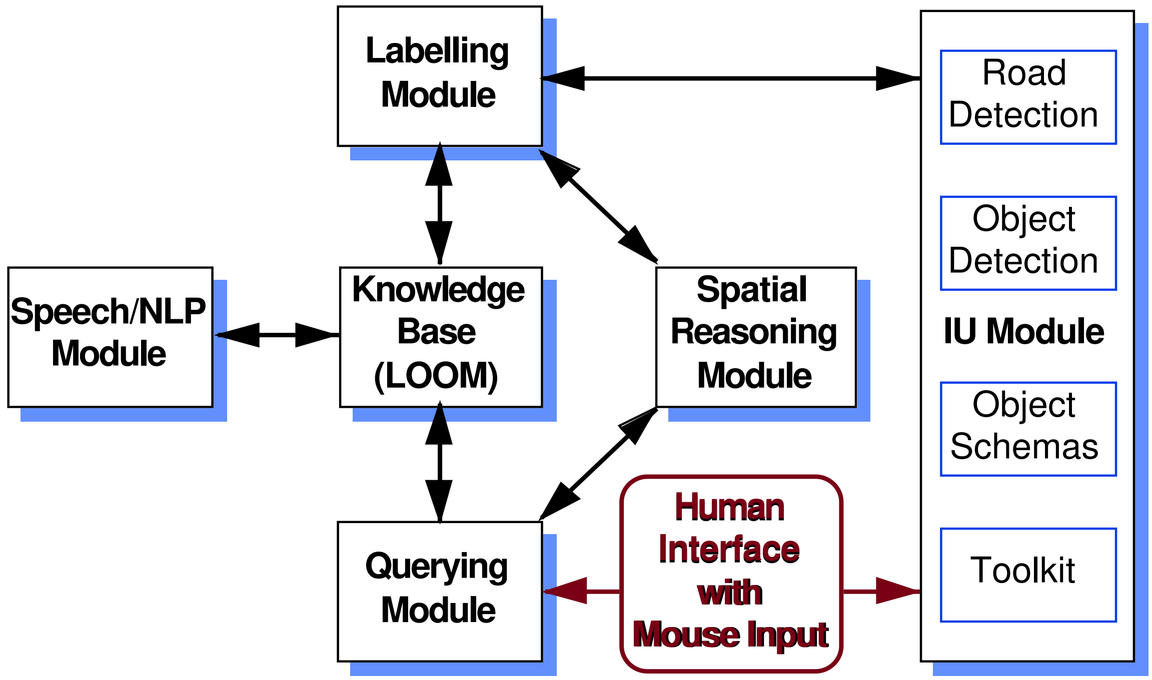


Figure 5.7: *Functional Architecture of* Show&Tell *System.*

image, that is then displayed for the analyst's confirmation. The analyst can then manually augment the annotations. In the query phase, the database user can retrieve images and entities in the images using either a natural language interface or by clicking on portions of a displayed image.

In PICTION, the caption was a *naturally* available source of collateral information that, one could say, justified its usage. However, in Show&Tell, the image analyst views the image and then annotates it through a sophisticated user interface. While one could argue that the case for image interpretation is weakened by the use of a human analyst to provide the collateral, one must keep in mind the end use of the information: content based retrieval. Since the collateral is used to annotate the images for future retrieval, we are justified in using the annotations for image interpretation and indexing.

There are several assumptions we are making in specifying the task and our proposed

Figure 5.8: *Result of road detection, landmark suggestion and speech recognition*

Figure 5.9: *Results of Image Interpretation*

Figure 5.10: *Results of Querying*

solution. From the perspective of an image analyst, this approach constitutes a healthy compromise between (i) tedious manual annotation, even when tools such as *snakes* are provided, and (ii) completely automated (image-based) interpretation. Since the task involves *co-referencing* image areas with textual descriptions, our system uses the text for dual purposes: co-referencing as well as for assisting image interpretation.

The second assumption concerns the use of preconstructed geometric site models. These have been used effectively in the RADIUS community for registering new images of a known site and subsequently for change detection. The initial version of *Show&Tell* does not use site models since the objective was investigation of linguistic context alone.

Finally, at this point we assume that an approximate shape representation for objects is sufficient for reasoning purposes. We represent objects such as buildings or areas by simple bounding boxes with orientation; polylines are used for roads, rivers. Our system does allow for more exact representations for display purposes.

5.2.2 Show&Tell Domain Model

The domain world of **Show & Tell** consists of named instances of the `building`, `road` and `logical_area` concepts. These are similarly made `concrete` as `building_object`, `road_object` and `area_object` respectively. These specializations are shown in pink shaded boxes in Figures 3.1, 3.2 and 3.3 . The object location algorithms we have are for qualitative *building* detection [Zhang and Srihari, 1996]. The set of spatial relations that are employed in interval based spatial reasoning are `left_of`, `right_of`, `strict_left_of`, `strict_right_of`, `above`, `below`, `strict_above`, `strict_below`, `between`, `included_in`, `includes`.

The natural grouping of objects done by the image analyst influences the object location and identification process. When an analyst (or collateral information source) mentions a collection or group of objects of the same type, those objects are then marked to be searched for simultaneously. This effect reflects our assumptions that a non-localized search would lead to a lot of false candidates, and that it is easier for an object locator to find a specified number of similar objects than search for them individually. We define a concept `object_locator` (Fig. 3.1) to represent the information required to locate an aggregate of objects of the same type. The region of the image that the objects are predicted to be located in is the `search_region`. Its default value is the complete image extent.

For example, *"Area Alpha contains four L-shaped buildings"* would lead to the creation of an instance of the `building_locator` concept. The shape information and the number of objects of this type are also stored in this instance. The `search_region` takes the default value and upon later interpretation involving spatial reasoning, the search region may be shrunk to a smaller region of the image (possibly corresponding to the extent of Area Alpha). An image containing say, 9 buildings may contain 1 to 9 `building_locator` instances depending upon how the analyst chose to group the buildings based on spatial proximity and/or shape similarity.

5.2.3 Example

To illustrate the use of collateral information, consider the aerial image (Fig. 5.8) and accompanying speech input (Fig. 5.11). A semi-automated road following algorithm generates a network of system-labeled road segments as described in Section 5.2.1. This network along with the image boundaries forms a planar graph. A graph algorithm finds and labels the *faces* [Harary, 1971] of this planar graph. Each internal face represents a `physical area` demarcated by roads and could correspond to a `logical area` or serve as a **landmark**. These landmarks are then used by the analyst in describing the contents of the scene. The result of the speech recognition and language processing results in a set of

(Loom) assertions that collectively define the scene hypothesis. The set of objects to locate and identify are: three `logical_area` instances and four `building` instances. In this domain, `physical_area` instances are either outlined by the user or system *a priori* or are computed as the minimum bounding box of the objects explicitly mentioned to be contained in it. In the latter case, the `logical_area` instance is bound to the smallest search region specified for the corresponding `physical_area` till all contained objects are located. Furthermore, area objects are easier to locate than building objects. Specialized buildings are more expensive to locate than simple rectangular buildings. Such information is encoded *a priori* in the concepts for object locators.

```
This image depicts jarvis park.
Label segments roger two roger four and roger five as main
street.
Label segment roger three as king street.
In the western half of area delta is the baird research
institute.
It consists of four buildings.
Of these the leftmost is a long rectangular building.
Label this as the kelly laboratory.
This is used for developing and testing new products.
Label the l-shaped building as the operations center.
Label the large two storied building as the headquarters
building.
This is where most of the administrative offices are
located.
Label the small square building as the communications
building.
```

Figure 5.11: *Sample Speech Annotation for Image in Figure 5.8.*

The `logical_area` Jarvis Park is bound first to the image object *PL0*. For every system labeled physical area object that is referred to by the analyst, a `logical_area` object is created. Therefore, an instance is created for area Delta. The control algorithm then constrains the search for Baird Research Institute to the western half of area Delta (D) by altering its

```
1.(merge-road-segments ''Main Street'' R2 R5 R4)
2.(create obj-6-conc 'building-object)
3.(create obj-6-log 'building)
4.(tell (:about obj-6-log (concrete obj-6-conc)
                          (name ''Kelly Laboratory'')))
5.(tell (:about obj-6-conc (belongs-to-image ''M7'')
                           (shape 'LONG-RECTANGULAR)))
6.(create locator-1 'building-locator)
7.(tell (:about locator-1
                (interest-level 1) (num-obj 2) (search-region PL0)
                (:filled-by instance  obj-6-conc  obj-4-conc)))
8.(tell (:about sp-rel-1
                (rel-type 'RIGHT-OF) (from-object obj-4-conc)
                (to-objects obj-6-conc)))
```

Figure 5.12: *A small subset of the constraints generated from speech input*

search region. The actual boundary of this area will be computed by the bounding box of the objects it contains and it temporarily bound to its search region *S1* (see fig 5.13). The search regions of the buildings are then constrained to *S1*. Next, the algorithm chooses to search for long rectangular buildings using the qualitative models of buildings encoded in the object locator tools. The choice of which object type to look for next is encoded in a few heuristic rules in the decision stage based on size of search region, number of objects, tool reliability, etc. Once the long rectangular building is located it constrains the search regions of the other buildings to search region S2. Finally, the four buildings are labeled and Baird Research Institute's boundary is computed.

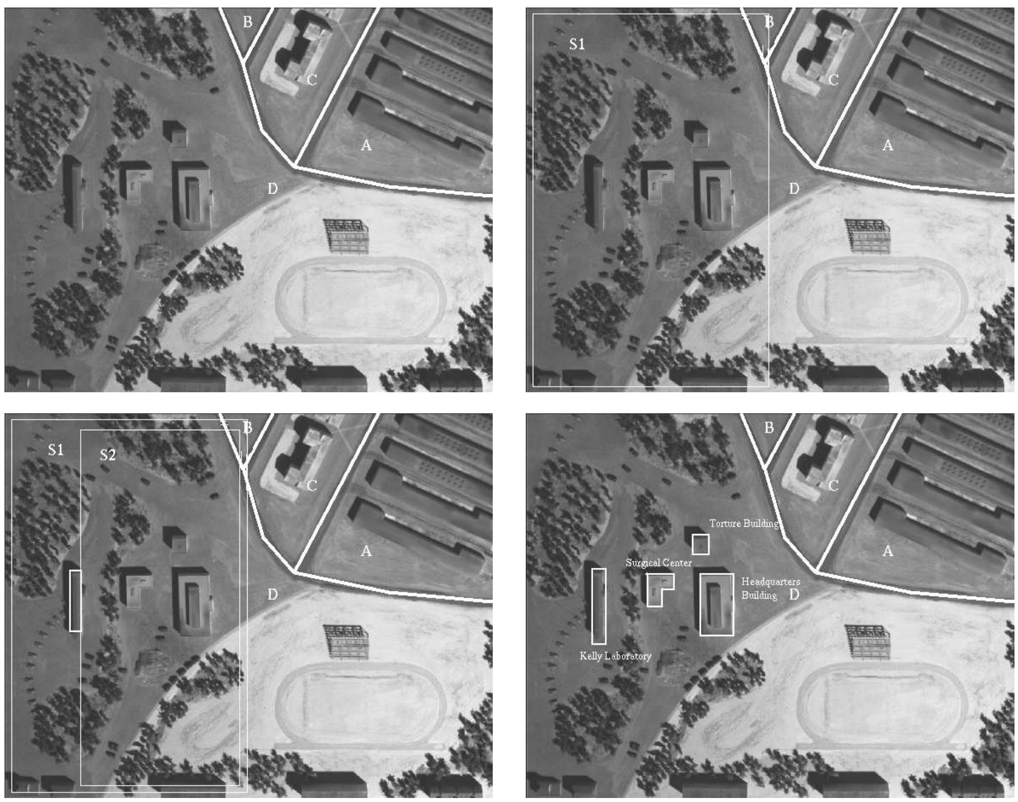


Figure 5.13: *An illustration of constrained search. Top left: Image with landmarks and suggested areas A-D. Top right: Constrained search region (S1) for Baird Research Institute. Bottom left: Kelly lab. located and constrained search region (S2) for other buildings. Bottom right: All buildings and areas located and identified.*

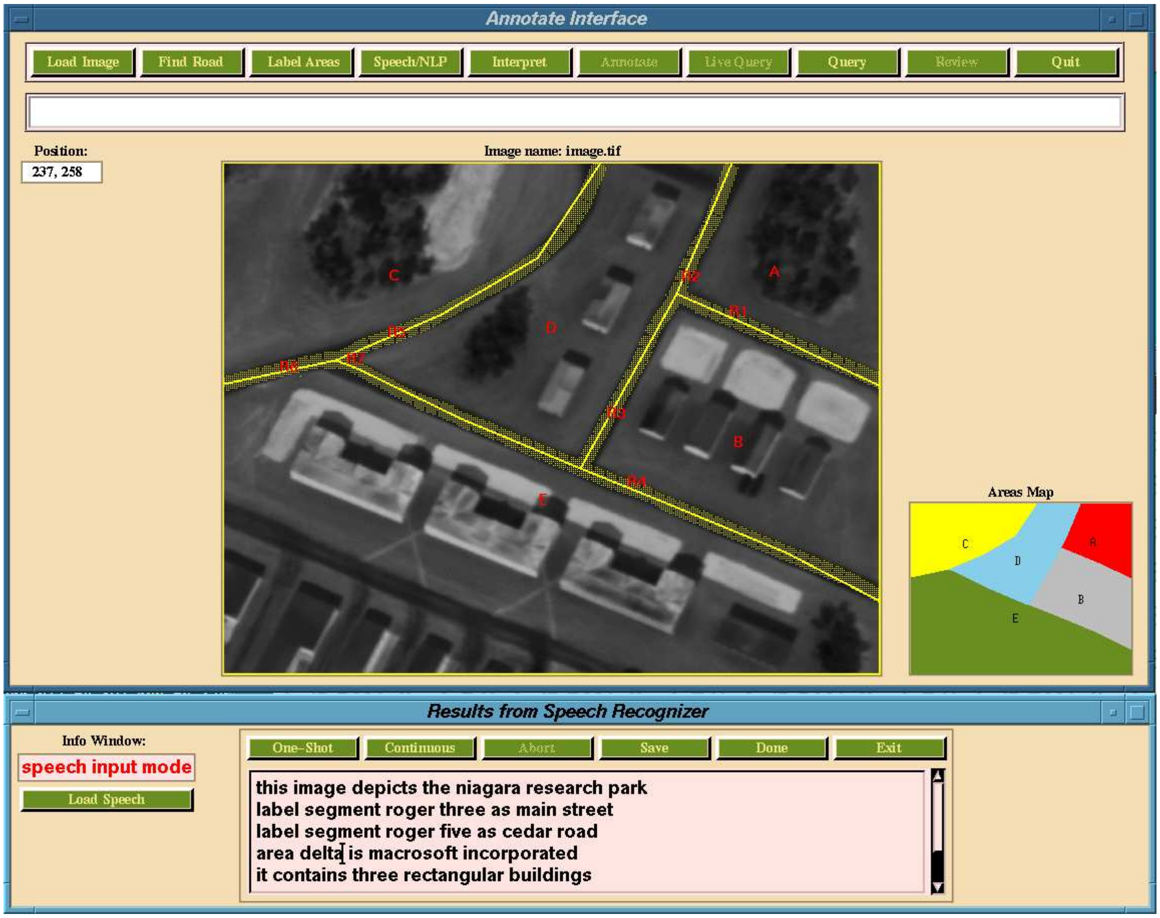


Figure 5.14: *Result of road detection, landmark suggestion and speech recognition for image M0*

```
This image depicts the niagara research park.
Label segment roger three as main street.
Label segment roger five as cedar road.
Area delta is macrosoft incorporated.
It contains three rectangular buildings.
Label them from top to bottom as storage center
and operations building and corporate headquarters
building.
```

Figure 5.15: *Sample Speech Annotation for Image in Figure 5.14.*

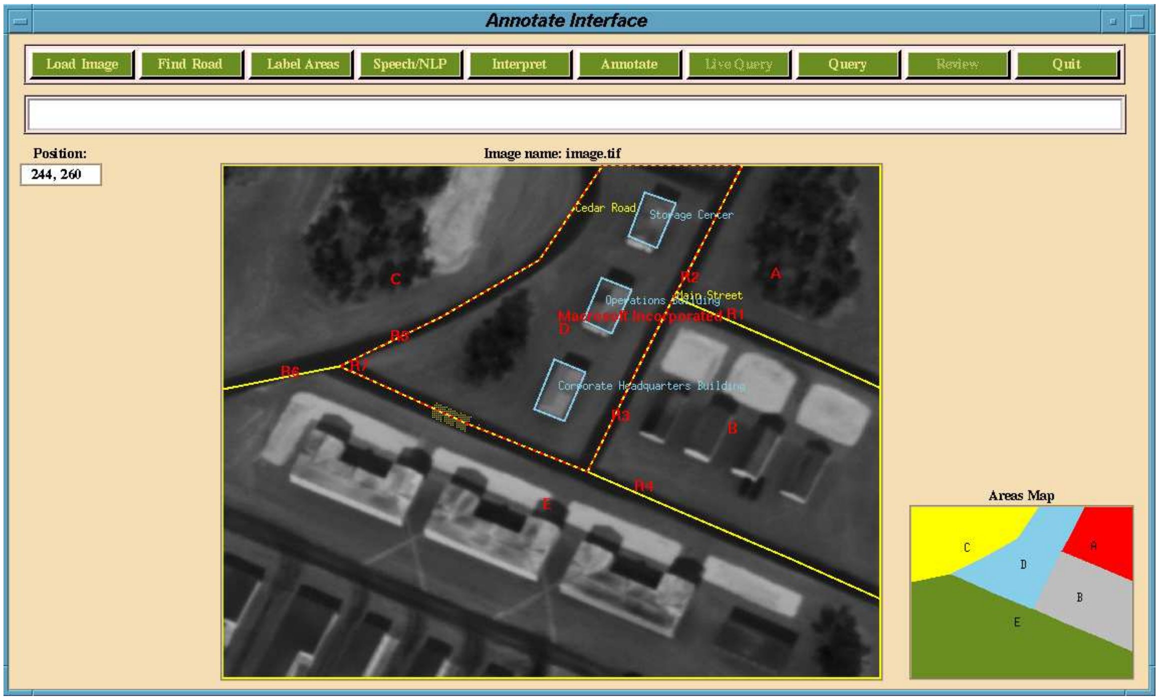


Figure 5.16: *Results of Image Interpretation*

5.3 SpatialFEVAHR

In both PICTION and Show&Tell, the sensing of the scene or image capture is passive. Once the digital image has been captured at a fixed resolution, using the collateral based vision paradigm increases the effeciency of object recognition only within the same order of magnitude. However, in applications where the world is sensed actively in real time, visual search techniques provide speed improvements that are potentially higher. For instance, an active vision sensor whose resolution and field of view can be controlled, can selectively sense in a goal driven manner. We explore spatial reasoning to guide the search for objects by an autonomous robot with vision capabilities and a psuedo-natural language interface.

5.3.1 Background

Recently there has been a lot of interest in guided autonomous mobile robots. The use of a natural language command interface is being actively investigated by many [Bandera *et al.*, 1994; Lüth *et al.*, 1994; Fuhr *et al.*, 1995]. The ability to deal with deictic gestures to localize objects has been demonstrated [Firby *et al.*, 1995]. However, no one has yet explored an interface containing locative spatial expressions that can be used for object localization. We claim that a bit of knowledge and instruction in a naturally interactive domain can go a long way.

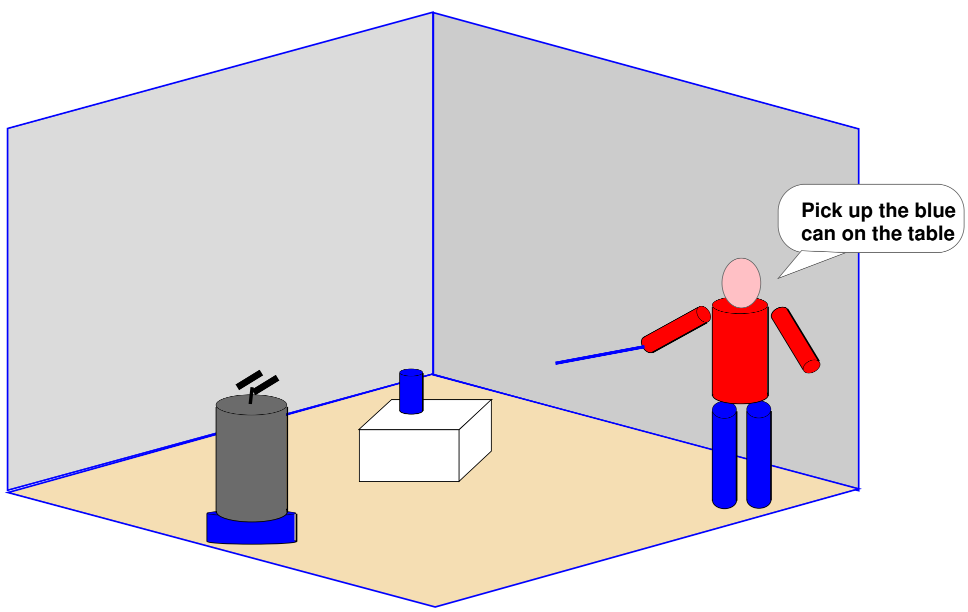


Figure 5.17: *SpatialFEVAHR Application Scenario*

5.3.2 Overview of FEVAHR

According to Bandera, Hexmoor and Shapiro [Bandera *et al.*, 1994], the objective of the Foveal Extra-Vehicular Activity Helper-Retriever (Foveal EVAHR or FEVAHR) project is to provide the EVAHR with foveal active vision and an autonomous behavioral architecture that supports its control and the refinement of skills. As part of that objective, the architecture must support cognitive functions and natural language understanding, and the machine vision functions must support wide field-of-view search and high resolution object recognition. A desirable subset of the command language understood by the agent would include *visual information*. For example, the robot should be able to understand and use effectively visual cues like *color* and *shape*. This information proves to be necessary for the robot to recognize objects in its world. It also provides a natural way for the supervisor of the robot to refer to objects. Like visual information, *spatial* information can also prove useful in recognizing objects and naturally referring to them. The current project specifications allow for a particular form of spatial reference, namely, pointing gestures.

5.3.3 Visual Search

We propose to investigate the use of spatial reasoning in visual search and object identification. For instance, spatial information is extracted from a supervisor's command to the robot. Assuming the command required the robot to locate and identify certain target object(s), this information is then used to effectively control the vision head to search for the target object(s). We focus on qualitative rather than quantitative spatial information. For example, "*the blue ball to the left of the red box*" provides qualitative locative information as opposed to "*the blue ball is at position (x, y, z)*" which provides quantitative information that may be imprecise, unavailable or difficult to obtain. Given a reference object (RO), an object to be located (LO) and a spatial relation between them, the position of the RO and the spatial relation is used to constrain the search region for the LO. Wixson and Ballard [Wixson and Ballard, 1994] give a rigorous cost-benefit analysis of this technique when there is one target object (LO) and one intermediate object (RO). Current approaches discuss visual search in isolation, that is, the RO and LO are uniquely identified. The combination of visual search with the object labeling problem as we propose, leads to choices in ROs and LOs, planning and incremental constraint satisfaction.

5.3.4 Extensions to the Command Interface

We extend the command language to provide simple spatial information in an unambiguous manner. Since understanding spatial expressions in natural language is a vast topic of research in itself, we shall restrict our command interface to handle simple spatial language only. The currently implemented grammar includes the following:

```
<command>    ::=  Stop | <action> <np>
<action>     ::=  Go To | Find | Follow
<np>         ::=  <npr>
                     | (that | this | the | a) <prepphrase>
<prepphrase> ::=  <cat> <prep> <np>
<cat>        ::=  [<adj>] <n>
```

The current lexicon is :

```
<prep>       ::=  to the left of | to the right of | behind
                  | above | in front of
<adj>        ::=  red | green | blue | black | white
<n>          ::=  robot | pen | table | ball | box
<npr>        ::=  John | me | you
```

An example command :

```
Find the blue pen on the table behind you.           (1)
```

We shall describe in the following sections how spatial information is extracted from such a command and used in visual search.

5.3.5 Spatial Representation and Reasoning

Spatial reasoning is a widely (ab)used term. The following define our usage of some of these terms:

- *Spatial representation* refers to the data structures we choose to represent coordinate systems and reference frames.
- *Shape representation* deals with the data structures to represent object shape.
- *Spatial reasoning* refers to heuristics and algorithms to perform inference on shape and spatial entities.

For example, we could choose a Cartesian world coordinate system as the spatial representation and bounding boxes as object shape representations. The bounding boxes form point sets in our spatial representation. Relations between these objects can be defined in terms of their corresponding point sets. Spatial relations that are "view dependent" are called *projective*, (e.g. left_of) and those that deal with the topologies of the related objects are called *topological* (e.g. inside). Spatial reasoning would include those operations that instantiate, verify and infer spatial relations between the objects.

In the FEVAHR domain, we are interested in:

- implementing and experimenting with different 3D spatial and shape representations,
- spatial relations expressed in different frames of reference, and

- simplifications to existing models that might be required for efficiency in a real-time environment.

A command string is parsed to generate a graph representing the extracted information, where the nodes represent objects mentioned and the arcs represent spatial relations between the nodes. The objective of a pseudo-natural language interface is not to explore ambiguous natural language but to provide an easy specification for spatial reference and relations.

Example 1 presented in the previous section will generate the following:

```
Number of objects : 3
Object number 0 : Shape : Pen    Color : Blue
Object number 1 : Shape : Table  Color : Unknown
Object number 2 : Shape : Self   Color : Unknown
Number of relations : 2
Prep: from: 1 to : 2 Type: BEHIND
Prep: from: 0 to : 1 Type: ABOVE
```

In other instances, the spatial relations might be attached to different frames of reference. For example, "*the robot behind me*" rather than "*the robot behind you*" specifies the qualitative position of the robot in a coordinate frame centered at the supervisor's position with the (y and x) axes aligned with the supervisor's front direction and right side (resp.). Transformations of volumes representing qualitative positions of objects is a form of inference that the spatial reasoning system performs.

5.3.6 Attention Control for Object Location

The foveal vision subsystem provides a set of accessible routines and variables that I propose to extend or use to implement visual search. For example, the routine `scan_entire_field_of_regard (object_features, base_angle, angle_range)` implements a 2D search for a specified object. A model database contains information to discriminate features for object recognition. We augment this database with relative cost information based on the time complexity of feature extractors, algorithms that recognize different objects, and domain specific restrictions on the relative sizes of objects. Smaller objects require higher resolution and therefore take longer to find. A simple planning algorithm will call the above-mentioned routine in succession on the intermediate objects till it successfully locates the target object.

5.3.7 Current Status

Spatial representations and a spatial reasoning algorithm have been implemented to handle simple spatial relations. For example, commands such as Example 1 have spatial relations expressed in the robot's reference frame only. We have chosen a robot centered coordinate system because the robot's position in a world coordinate system might not be known. We have not yet implemented the transformation routines required for relations in different reference frames. The control algorithm currently simulates the vision routines and calls them in the appropriate order based on the graph representation of the scene and the cost information in the model database. To complete Example 1, the symbolic behavior as simulated currently can be described as (comments in curly braces):

```
1) Target index: 0 {Target is Pen}
2) In scan_2d_field_of_regard for ObjID: 2 {Search for robot}
3) Object 2 found {Trivially, since robot knows its position}
```

```
4) Constrain Search Region : ObjID: 1
   Reln:BEHIND, Base_angle 180, base_range 90
   {Constrain the search region for the table by changing the
    base angle and range for the vision routine to search in}
5) In scan_2d_field_of_regard for ObjID: 1 {Search for table}
6) Object 1 found {Simulated for now}
7) Constrain Search Region :  ObjID: 0
   Reln:ABOVE, ground range 90, vert_range 30, ground_angle 180
   {Constrain the search region for the pen by changing the
    base angle, base range, tilt angle and range}
8) In scan_2d_field_of_regard for ObjID: 0 {Search for pen}
9) Target object found!
```

Chapter 6

Experimentation and Evaluation

There are several questions one has to answer when proposing a new technique or algorithm. What can be said about the soundness and completeness of the algorithm? How well does the algorithm perform compared to existing algorithms? How well does the algorithm or technique match the application or the real-world problem it has been applied to? What is the impact of this technique on the afore-mentioned problem? What are its limitations and shortcomings? In chapters 3 and 4 we have attempted to answer some of the above questions.

In this dissertation, we have proposed several new algorithms, designed an extensible architecture and implemented our ideas in several applications. An evaluation of each of this contributions needs to be done separately.

6.1 Algorithms

Algorithms are usually evaluated on correctness and its performance characteristics: average and worst-case. We have presented two algorithms (NAC4 and HetCSP) in constraint satisfaction in chapter 4 along with algorithmic analysis. We now present experimental analysis of the algorithm **HetCSP** presented in chapter 4, page 51.

6.1.1 Algorithm HetCSP

Implementation

This algorithm has been implemented in Allegro Common Lisp and uses a constraint package included with the public domain software *SCREAMER* [Siskind and McAllester, 1993b], for enforcing local consistency on interval variables.

Generation of Test Cases

The algorithm was tested on randomly generated CSPs. The generation of the finite domains was simulated. The routine to generate random CSPs took four parameters: number of variables in the CSP : *num-var*, mean number of values for each variable : *num-val*, continuous interval from which discrete values are chosen: *int-domain*, and approximate variance in the number of values : *var-val*. The output of the above routine is a triplet of arrays:

IntDomains : an array of num-var interval variables initialized to the interval *int-domain*

FDDomains : an array of num-var discrete finite sets of values within the interval. This is the simulated *Generator* of finite domains.

Constraints : a 2D array (size num-var $\times$ num-var) of constraints between variables. For simplicity, in this experiment we define a single constraint $<$. Without any loss in generality, we generate the constraints such that if Constraints[i,j] exists, then $i < j$. Since the $<$ relation is transitive, this restriction prevents cycles in the constraint relationships. Note however that i < j does not imply that a constraint exists between them. The assignment of constraints is randomly generated.

Experimentation

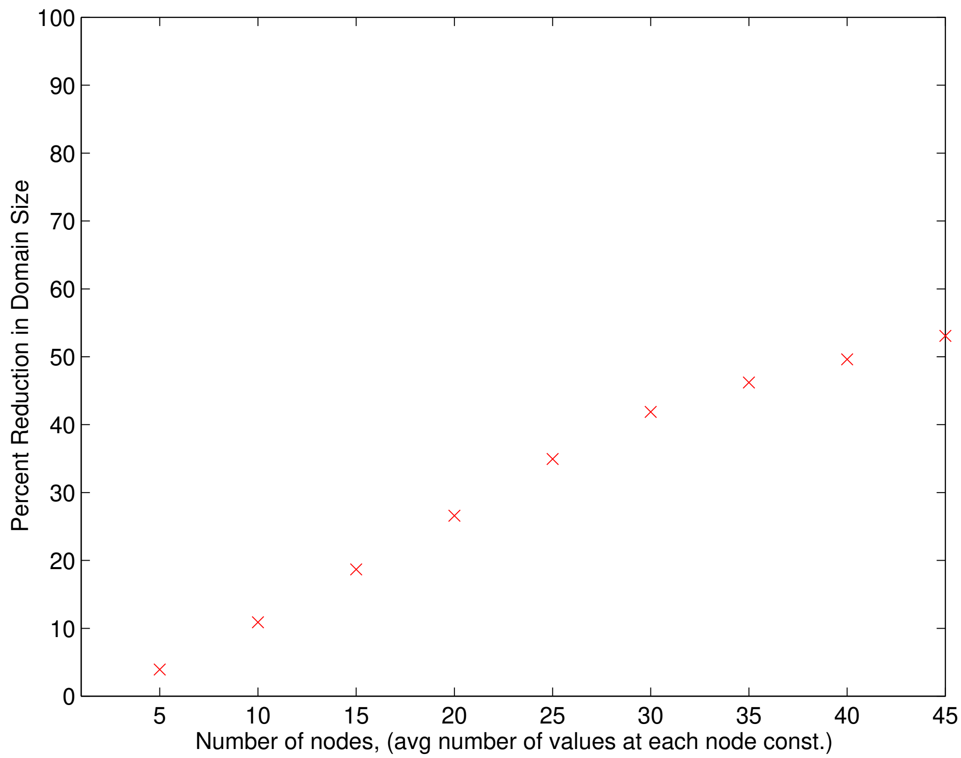


Figure 6.1: *Experiment 1: Each point represents 100 randomly generated CSPs that had at-least one or more possible labelings. The mean number of values at each node was 30 and the approximate variance was 3. Since the mean number of values at each node is constant and the number of constraints to be satisfied increases, there is an increase in the reduction of the domain sizes.*

Figures 6.1 to 6.4 display the results of four experiments that were performed to analyze the algorithm. We present the percentage reduction in domain size versus several parameters. Each point represents the average of 100 randomly generated CSPs that had at-least one or more possible labelings. For each CSP, the original domain size is calculated as the sum of the sizes of each variable's domain. If the order of magnitude of the cost of generating the domain of a variable is $\Theta(|\ I_i\ |)$ then the percentage reduction in domain size gives the reduction in generation cost. If the generation cost is proportional to higher powers of the domain size, then the savings are greater. In all the experiments the intervals are initialized to [0:100].

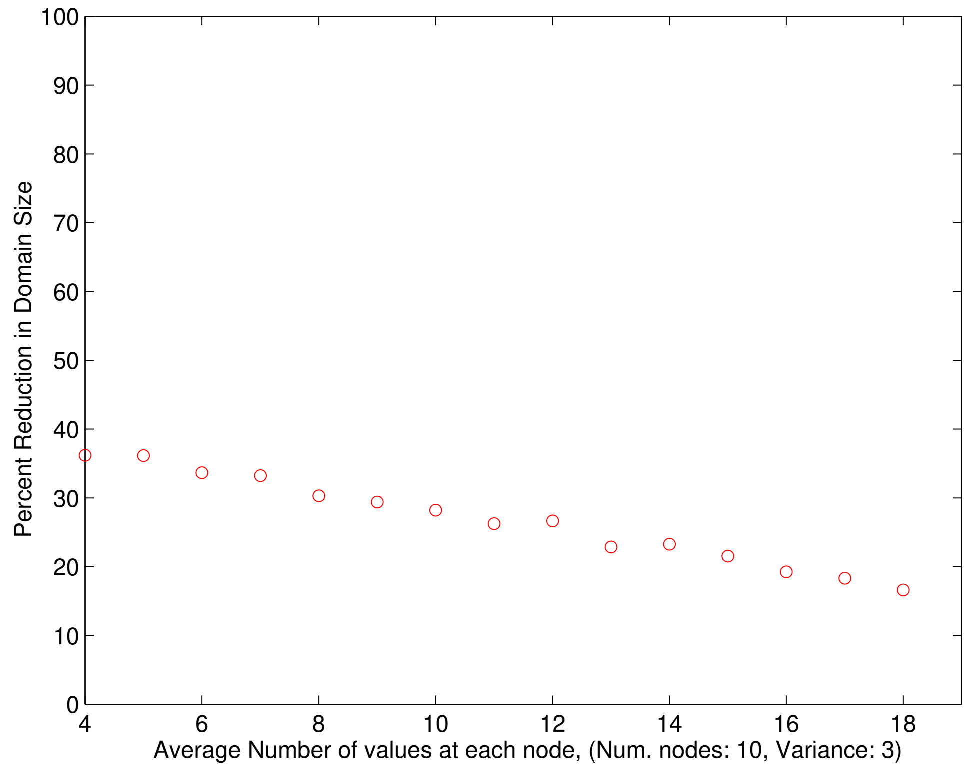


Figure 6.2: *Experiment 2: Each point represents 100 randomly generated CSPs that had at-least one or more possible labelings. Since the interval size is constant, as the mean number of values per node increases, the reduction in the domain size decreases.*

In Exp. 1, the mean number of values at each node was 30 and the approximate variance was 3. Since the mean number of values at each node is constant and the number of constraints to be satisfied increases, there is an increase in the reduction of the domain sizes. In experiment 2, as the average number of values per node increases, the reduction in the domain sizes decreases. Since the initial interval is constant, a larger number of values from the same interval leads to larger intermediate intervals and hence a lower total reduction in domain sizes.

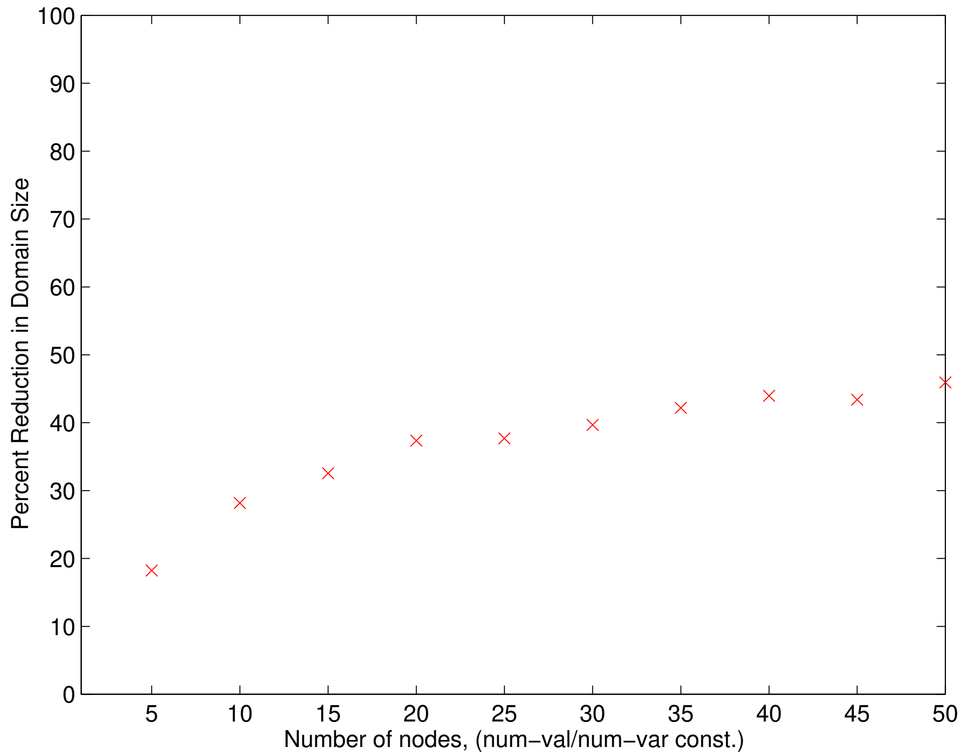


Figure 6.3: *Experiment 3: Each point represents 100 randomly generated CSPs that had at-least one or more possible labelings. By keeping the mean number of values per variable constant, we are able to counter the effect of fewer values per variable and measure how the algorithm scales up with increasing number of variables (in contrast with Experiment 1).*

We measured the effect of increasing the number of variables in experiment 1 . However, there is a side effect of the average number of values per variable decreasing. Therefore, in Experiment 3, to counter balance that side effect, we increase the number of values per variables as we increase the number of variables. We find that the reduction in domain sizes increases at a slower rate than in Exp. 1. We are also able to measure the scalability of the algorithm with increasing number of variables. We find through Experiment 4 that the variability in the relative sizes of the domains does not affect the domain reduction. Since the values are randomly chosen from the same initial interval domain, fewer values does not imply smaller interval sizes.

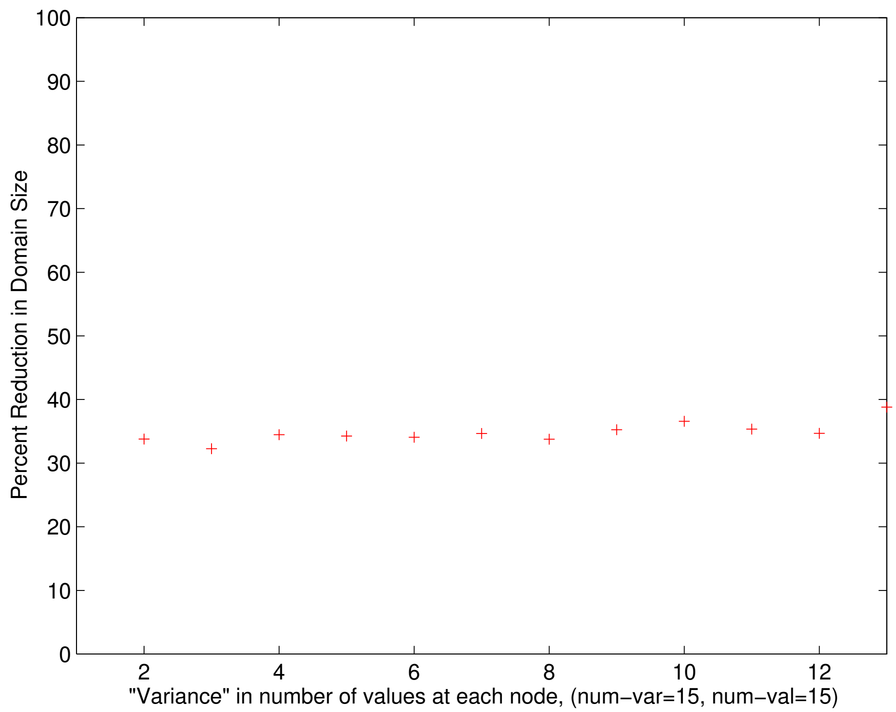


Figure 6.4: *Experiment 4: Each point represents 100 randomly generated CSPs that had at-least one or more possible labelings. The variability in the relative sizes of the domains does not affect the domain reduction.*

6.2 Architecture

Collateral based vision (CBV) has been demonstrated in three applications. In evaluating the role of CBV in the real-world problems that it addresses in these applications, one must keep in mind that large systems have components that have different testing criteria.

PICTION can roughly be modularized into 3 components: The natural language module, the interpretation module and the object location module. A natural language module that processes sentences from a subset of the English grammar needs to analyzed on whether it generates the *correct* semantics for the sentences it is designed to process and then on how large a subset of natural English that it can process. On the other hand, the object location module that applies low-level vision routines for model based object recognition and location must be evaluated on its precision and recall. In other words, how many candidates does it miss or incorrectly identify.

However, an architecture that unites the two modules needs to be tested on all the above criteria while cleanly separating the errors of both. PICTION has been tested with over a hundred test cases. However, the testing of the language and object location modules was pertinent there. The interpretation module was able to represent the scene, drive the object location routines and improve the efficiency. What was more important in the evaluation of the interpretation module was the added expressiveness (non-binary constraints and the ability to handle relative characteristic constraints) and the ability to handle partial labelings. Since multiple object classes were not natural in the PICTION domain, the increased efficiency due to visual search did not have a great impact.

Show&Tell has unfortunately not been tested in its entirety on many images: the bottlenecks to testing being the speech and language modules. However, the interpretation module has been tested on several examples and its impact is as expected: Increased efficiency in object location due to spatial constraints where existing; ability to easily model and represent the application domain for information retrieval.

SpatialFEVAHR was implemented as a proof-of-concept application since the expected payoff was highest in this domain. Once again, due to lack of real world data, we are unable to measure the impact of CBV. However, in related work[Wixson and Ballard, 1994] where one intermediate object was used to locate the target, speed improvements of the order of 7 times where reported. We believe our work that employs potentially multiple intermediate objects should lead to speed improvements of 7 correspondingly multiplied. However, differences in application scenarios might alter that significantly. What is equally important is the fact the SpatialFEVAHR presents a framework for extracting spatial information from the existing natural interface to the robot and in employing the information in a vision task.

Collateral based vision is a new paradigm and it is difficult to quantify the impact of such techniques. The abstract formulation of the control algorithm in HetCSP has been tested experimentally in the previous section. Increases in efficiency have been demonstrated. Due to lack of data from the Show&Tell and SpatialFEVAHR domains, it was extremely difficult to test the architecture with real data.

How much of an impact does our technique have on scene labeling? Firstly, it enables the detailed level of co-referencing that was not possible before. Secondly, it improves the efficiency by using partial information to eliminate impossible labelings.

6.2.1 Limitations

We have claimed that the control algorithm is independent of the representation insofar as the domain problem can be represented using the structures mentioned in chapter 3. Obviously, there are several limitations on the kinds of vision problems that this architecture is suited for. Nevertheless, we feel that there exist several real-world application domains where bottom up vision is difficult and qualitative collateral information regarding scene contents is readily available. Some of the limitations can be enumerated as:

1. The main limitation is the need for an **accurate** scene hypothesis. The control algorithm can handle incomplete scene hypothesis very well as well as error-prone object location tools. However, it assumes that the collateral information provided is correct. There is no mechanism of extracting the maximal subset of the hypothesis that has positive re-inforcement from the scene data. Since the motivation of this work is the acknowledged difficulty in bottom-up scene understanding, we do not attempt to include semantic information from bottom-up scene understanding to update our scene hypothesis.

2. Several spatial expressions are not incorporated in the applications because the representation and semantics of those expressions are still a recent research topic. Examples of such expressions are *along*, *across* etc.

3. The current representation of interval variables lends itself to the bounding box representation for multi-dimensional intervals. This is an impediment in the representation of and reasoning with complex spatial expressions. However, the drawback is not very serious since the use of spatial reasoning in collateral based vision is to localize search rather than to extract deep semantics of spatial expressions.

4. Temporal reasoning will be a necessary capability of any high-level vision system that processes image sequences (video). Our current architecture for collateral based vision doesn't have any temporal reasoning capabilities.

5. Total ordering of the apriori cost of invoking each object location tool must be available to the control algorithm. This might be difficult to estimate without sufficient knowledge of the domain in which the tools are to applied.

Chapter 7

Conclusions and Further Research

7.1 Conclusions

In this dissertation we have presented an architecture to incorporate collateral information in the high level computer vision task of object location. The declarative specification of the scene hypothesis and the domain independent control algorithm that exploits spatial information to drive the vision process, are two major contributions of this dissertation. This work has been theoretically grounded in constraint satisfaction algorithms. Apart from the application of standard CSP algorithms, several new techniques in constraint satisfaction have been developed. We have provided an algorithm that performs consistency filtering for certain types of non-binary constraints. We have also presented an innovative application of interval arithmetic in constraint satisfaction. Though interval constraint satisfaction is an extensive research area, we believe this to be the first attempt at using interval arithmetic to reduce the domain generation cost of finite domain CSPs. A new algorithm has been proposed here for the above. Illustration with a simple example, analysis and implementation of the algorithm have also been detailed here.

We have demonstrated the architecture in three implemented systems and feel that it is simple yet powerful for application in several domains. There are however several issues that we have not addressed. We discuss some of the future work in the next section.

7.2 Future Work

There are several directions that this research can proceed along. We mention some of the issues that can be explored further:

1. The cost function evaluated for different classes in the *decision* stage is currently based on weights computed by trial and error. The encoding of relative tool costs and measures should be made more declarative or learned from a training set. Declarative specification of a priori costs compared to training in a specific domain should be investigated.

2. Though we have explored multiple reference frames in the SpatialFEVAHR domain, alternative representations that can be investigated are reference frame independent representations [Fuhr *et al.*, 1995]and object-centric representations.

3. The constrained search paradigm is currently being applied only for spatial relations between objects. We need to explore its application to other visually relations, for example, if darker-than(A,B) and A is uniquely labeled, then that constrains the value of the characteristic constraint (brightness) of B, and could be used for a locative purpose.

4. We have identified some problems with the incremental constraint checking coupled with visual prediction. For example, Consider an object A that is uniquely and incorrectly labeled with instance L1 and is used to constrain the search for some other

object B. The object locator tool for object B might be misled by this false search region and there doesn't seem any obvious way to recover from this error. One should explore the ideas of "justification" in truth maintenance systems to overcome this difficulty. This problem is however alleviated in the case where object location is conservative and generates false positives along with the actual label. In that case, however, the advantages of constrained search are reduced since the occurrences of uniquely labeled objects is reduced.

5. In the Show&Tell example, the roads did not serve as landmarks, but that does not preclude their usage in visual search. Spatial reasoning techniques to use "ribbon" shaped objects for spatial verification and prediction need to be investigated.

6. There are several exciting extensions to SpatialFEVAHR that we envisage. Apart from providing a robust implementation of complex spatial references and transformations, we propose to extend this work to control the *robot body* and not just the vision *head* in visual search. For example, a spatial reference may be made to an object not in the current field of regard of the robot. The robot would then have to infer potential vantage points which could lead to object location. There is already an interest in the active vision community for active control of the camera for object recognition. This work supplements that effort by using qualitative high level knowledge in a related task of object location. Integrated strategies for recognition and location that use such instance specific high level information as well as domain specific optimized strategies for recognition are future proposed research efforts.

Appendix A

Interval Arithmetic and Constraints

We present the basic notions of rational interval arithmetic (IA) [Moore, 1966], and interval arithmetic applied to constraint satisfaction [Davis, 1987; Benhamou and Older, 1993; Hyvönen, 1989; Hyvönen, 1992].

We consider $\Re \bigcup \{-\infty, +\infty\}$, the set of real numbers augmented with the two infinity symbols, and the natural extension of the relation $\leq$ to this set. The **interval** is defined as a set of real numbers represented as a number pair. For every $a, b \in \Re, a \leq b$, we use the following notations for intervals:

$[a : b] = \{x \in \Re | a \leq x \leq b\}$,

$[a : b) = \{x \in \Re | a \leq x < b\}$

$(a : b] = \{x \in \Re | a < x \leq b\}$,

$(a : b) = \{x \in \Re | a < x < b\}$

$[a : +\infty) = \{x \in \Re | a \leq x\}$,

$(a : +\infty) = \{x \in \Re | a < x\}$

$(-\infty : b] = \{x \in \Re | x \leq b\}$,

$(-\infty : b) = \{x \in \Re | x < b\}$

$(-\infty : +\infty) = \Re$

In IA, basic algebraic relations can be defined as:

$A + B = \{a + b | a \in A, b \in B\}$

$A - B = \{a - b | a \in A, b \in B\}$

$A * B = \{a * b | a \in A, b \in B\}$

$A/B = \{a/b | a \in A, b \in B\}, if 0 \notin B$

For the above set of relations we can define the limits of the resultant interval in terms of the original intervals as :

$[a : b] + [c : d] = [a + c : b + d]$

$[a : b] - [c : d] = [a - d : b - c]$

$[a : b] * [c : d] = [min(ac, ad, bc, bd) : max(ac, ad, bc, bd)]$

$[a : b]/[c : d] = [a : b] * [1/d : 1/c], if 0 \notin [c : d]$

For a detailed description of the properties of interval arithmetic, one can refer [Moore, 1966; Alefeld and Herzberger, 1983]. The definition of *solution functions* or *narrowing functions* [Benhamou and Older, 1993] associated with every n-ary relation of $\Re$ is the basis of interval constraint satisfaction. The main properties of the narrowing functions are

1. Contractance: The narrowed intervals are smaller than the initial intervals
2. Correctness: Every valid solution in the theoretical real numbers lies in the narrowed intervals
3. Monotonicity: The narrowing preserves the inclusion and
4. Idempotence: The narrowed intervals have to be computed only once.

Several algorithms based on the iterative application of narrowing functions have been proposed in the literature [Davis, 1987; Benhamou and Older, 1993; Hyvönen, 1989; Siskind and McAllester, 1993a; Benhamou *et al.*, 1994; Faltings, 1994; Sam-Haroud and

Faltings, 1996]. The differences in the algorithms lies primarily in the set of relations over $\Re$ for which practical narrowing functions have been constructed and in the representation of the space of solutions. Invariably the algorithms terminate trivially and are independent of the order of application of the constraints.

Bibliography

[Alefeld and Herzberger, 1983] G. Alefeld and J. Herzberger. *Introduction to Interval Computations*. Academic Press, 1983.

[Badler, 1975] N. Badler. Temporal scene analysis: Conceptual descriptions of object movements. Technical Report 80, Dept. of Computer Science, University of Toronto, 1975.

[Ballard and Brown, 1992] Dana H. Ballard and Christopher M. Brown. Principles of animate vision. *CVGIP: Image Understanding*, 56(1):3–21, July 1992.

[Ballard *et al.*, 1978] Dana Ballard, Cristopher Brown, and J. Feldman. An approach to knowledge directed image analysis. In *Computer Vision Systems*, pages 271–282. Academic Press, 1978.

[Bandera *et al.*, 1994] Cesar Bandera, Henry Hexmoor, and Stuart Shapiro. Foveal machine vision for robots using agent based gaze control:final report. Technical Report 613-9160001, Amherst Systems, September 1994.

[Barrow and Tenenbaum, 1976] Harry G. Barrow and Jay M. Tenenbaum. Msys: A system for reasoning about scenes. Technical Note 121, Artificial Intelligence Center, SRI International, Menlo Park, CA, April 1976.

[Barrow and Tenenbaum, 1977] Harry G. Barrow and Jay M. Tenenbaum. Experiments in model-driven scene segmentation. *Artificial Intelligence*, 8(3):241–274, June 1977.

[Benhamou and Older, 1993] Frédéric Benhamou and William J. Older. Applying Interval Arithmetic to Real, Integer and Boolean Constraints. *Logic Programming: The ALP Newsletter*, 6(2):13–14, May 1993. (Extended Abstract).

[Benhamou *et al.*, 1994] F. Benhamou, D. McAllester, and P. Van Hentenryck. CLP(Intervals) Revisited. In *Logic Programming - Proceedings of the 1994 International Symposium*, pages 124–138, 1994.

[Bessiere, 1991] C. Bessiere. Arc consistency in dynamic constraint satisfaction problems. In *Proceedings of AAAI-91*, pages 221–226, 1991.

[Binford, 1982] T.O. Binford. Survey of model-based image analysis systems. *International Journal of Robotics Research*, 1(1):18–64, 1982.

[Brooks, 1981] Rodney Brooks. Symbolic Reasoning among 3-dimensional models and 2-dimensional images. *Artificial Intelligence*, 17:285–349, 1981.

[Burhans *et al.*, 1995] Debra Burhans, Rajiv Chopra, and Rohini Srihari. Domain specific understanding of spatial expressions. In *Proceeding of the IJCAI-95 Workshop on Representation and Processing of Spatial Expressions*, Montreal, August 1995.

[Burns *et al.*, 1984] J. Burns, A. Hanson, and E. Riseman. Extracting straight lines. Technical Report 84-29, University of Massachusetts, Amherst, 1984.

[Buxton and Gong, 1993] Hilary Buxton and Shaogang Gong. Bayesian nets for mapping contextual knowledge to computational constraints. In *British Machine Vision Conference*, Guildford, England, 1993.

[Chandrasekaran and Narayanan, 1990] B. Chandrasekaran and N. H. Narayanan. Towards a theory of common-sense visual reasoning. In K.V. Nori and C. E. V. Madhavan, editors, *Foundations of Soft. Tech. and Theoretical Computer Science, Tenth Conference*, Lecture Notes in Computer Science, pages 388–409. Springer-Verlag, 1990.

[Chang *et al.*, 1987] Shi-Kuo Chang, Q.-Y. Shi, and C.-W. Yan. Iconic indexing by 2D-strings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(3):413–427, 1987.

[Chellapa and Kashyap, 1992] R Chellapa and R.L. Kashyap. Image understanding. In Stuart C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, volume 1. John Wiley, 2nd edition, 1992.

[Chopra and Srihari, 1995] Rajiv Chopra and Rohini Srihari. Control Structures for Incorporating Picture-Specific Context in Image Interpretation. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI95)*, pages 50–55, Montreal,Canada, August 1995. Morgan Kaufmann.

[Chopra *et al.*, 1995] Rajiv Chopra, Rohini Srihari, and Anthony Ralston. Hyperarc consistency and expensive constraints. Technical Report 95-21, SUNY at Buffalo, 1995.

[Davis, 1987] E. Davis. Constraint Propogation with Interval Labels. *Artificial Intelligence*, 32(3):281–331, 1987.

[Falk, 1972] G. Falk. Interpretation of imperfect line data as a three dimensional scene. *Artificial Intelligence*, 3(2):101–144, 1972.

[Faltings, 1994] Boi Faltings. Arc-consistency for continuous variables. *Artificial Intelligence*, 65(2):363–376, 1994.

[Faugeras and Berthod, 1981] O. Faugeras and M. Berthod. Improving consistency and reducing ambiguity in stochastic labelling: an optimization approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4:412–423, 1981.

[Firby *et al.*, 1995] R. James Firby, Roger E. Kahn, Peter N. Prokopowicz, and Michael J. Swain. An architecture for vision and action. In *Proceedings of IJCAI-95*, volume 1, pages 72–79, Montreal, Canada, 1995.

[Firschein and Fischler, 1987] O. Firschein and M.A. Fischler. *Readings in Computer Vision, Issues, Problems, Principles, and Problems*. Morgan-Kaufmann, 1987.

[Freska and Röhrig, 1993] C. Freska and R. Röhrig. Dimensions of qualitative spatial reasoning. In Piera Carrete and M. G. Singh, editors, *Proceedings of the III IMACS International Workshop on Qualitative Reasoning and Decision Technologies*, pages 483–492, Barcelona, 1993.

[Fua and Hanson, 1987] Pascal Fua and Andrew J. Hanson. Using generic geometric models for intelligent shape extraction. In *Proceedings: DARPA Image Understanding Workshop*, pages 227–233, Los Angeles, CA, 1987.

[Fuhr *et al.*, 1995] Thomas Fuhr, Gudrun Socher, Christian Scheering, and Gerhard Sagerer. A three-dimensional spatial model for the interpretation of image data. In *IJCAI Workshop on Representation and Processing of Spatial Expressions*, 1995.

[Funt, 1980] B. V. Funt. Problem solving with diagrammatic representations. *Artificial Intelligence*, 13(3):201–230, 1980.

[Garey *et al.*, 1974] M.R. Garey, D.S. Johnson, and L. Stockmeyer. Some simplified np-complete problems. In *Proc. 6th Annual ACM Symp. Theory Comput.*, pages 47–63, 1974.

[Garvey, 1976] J. D. Garvey. Perceptual strategies for purposive vision. Technical Note 117, AI Center, SRI International, 1976.

[Govindaraju *et al.*, 1992] Venu Govindaraju, Sargur N. Srihari, and David B. Sher. A Computational Model for Face Location based on Cognitive Principles. In *Proceedings of the*

10th National Conference on Artificial Intelligence (AAAI-92), pages 350–355, 1992. San Jose, CA.

[Grimson, 1992] W.E.L. Grimson. The cost of choosing the wrong model in object recognition by constrained search. *IJCV*, 7(3):195–210, April 1992.

[Gütting, 1988] R. Gütting. Geo-relational algebra: A model and query language for geometric database systems. In J.W. Schmidt et.al., editor, *Advances in Database Technologies EDBT88, International Conference on Extending Database Technology*, pages 506–527. Springer-Verlag, 1988.

[Hanson and Riseman, 1978] A. Hanson and E. Riseman. Visions: A computer system for interpreting scenes. In A Hanson and E. Riseman, editors, *Computer Vision Systems*, pages 303–334. Academic Press, New York, 1978.

[Haralick and Elliot, 1980] R.M. Haralick and G.L. Elliot. Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, 14:263–313, 1980.

[Harary, 1971] Frank Harary. *Graph Theory*. Addison-Wesley, 1971.

[Hentenryck *et al.*, 1992] P. Van Hentenryck, Y. Deville, and C. Teng. A Generic Arc-Consistency Algorithm and its Specializations. *Artificial Intelligence*, 57:291–321, 1992.

[Hernández, 1994] Daniel Hernández. *Qualitative Representation of Spatial Knowledge*. Number 804 in LNAI. Springer- Verlag, 1994.

[Herskovits, 1986] Annette Herskovits. *Language and Spatial Cognition*. Cambridge University Press, 1986.

[Howarth and Buxton, 1993] Richard Howarth and Hilary Buxton. Selective attention in dynamic vision. In *Proceedings of IJCAI-93*, Chambery, France, 1993.

[Hower, 1996] Walter Hower. Bottom-up Layout Generation. *Informatica*, 20(1):57–63, 1996.

[Hyvönen, 1989] Eero Hyvönen. Constraint Reasoning Based on Interval Arithmetic. In *Proceedings of IJCAI 1989*, pages 1193–1198, 1989.

[Hyvönen, 1992] E. Hyvönen. Constraint Reasoning Based on Interval Arithmetic: The Tolerance Propogation Approach. *Artificial Intelligence*, 58:71–112, 1992.

[ISX, 1991a] ISX Corporation. *LOOM Users Guide, Version 1.4*, 1991.

[ISX, 1991b] ISX Corporation. *LOOM Users Guide, Version 2.0*, 1991.

[Jungert, 1994] Erland Jungert. Qualitative spatial reasoning from the observer's point of view—towards a generalisation of symbolic projection. *Journal of Pattern Recognition*, 27(6), 1994.

[Kanade, 1980] Takeo Kanade. Survey: Region segmentation: Signal vs semantics. *Computer Vision, Graphics and Image Processing*, 13:279–297, 1980.

[Kimia *et al.*, 1995] B.B. Kimia, A.R. Tannenbaum, and S.W. Zucker. Shapes, shocks, and deformations i: The components of 2-dimensional shape and the reaction-diffusion space. *IJCV*, 15(3):189–224, July 1995.

[Kittler and Illingworth, 1985] J. Kittler and J. Illingworth. Relaxation labelling algorithms – a review. *IVC*, 3:206–216, 1985.

[Kosslyn *et al.*, 1981] S. M. Kosslyn, S. Pinker, G. Smith, and S. Schwartz. On the demystification of mental imagery. In N. Block, editor, *Imagery*, pages 131–150. MIT Press, 1981.

[Levine, 1978] M. Levine. A knowledge-based computer vision system. In A Hanson and E. Riseman, editors, *Computer Vision Systems*, pages 335–352. Academic Press, New York, 1978.

[Lüth *et al.*, 1994] T.C. Lüth, Th. Längle, G. Herzog, Eva Stopp, and U. Rembold. Kantra: Human-machine interaction for intelligent robots using natural language. In *3rd IEEE Intl. Workshop on Robot and Human Communication*, pages 106–111, Nagoya, Japan, 1994.

[MacGregor, 1991] Robert MacGregor. The evolving technology of classification-based knowledge representation systems. In *Principles of Semantic Networks:Exploration in the Representation of Knowledge*, pages 385–400. Morgan Kaufmann, Los Angeles, CA, 1991.

[MacGregor, 1994] Robert M. MacGregor. A Description Classifier for the Predicate Calculus. In *Proceedings of the Twelfth National Conference on Artificial Intelligence, (AAAI 94)*, pages 213–220, 1994.

[Mackworth and Freuder, 1985] A.K. Mackworth and E.C. Freuder. The complexity of some polynomial network consistency algorithms for the constraint satisfaction problem. *Artificial Intelligence*, 25:65–74, 1985.

[Mackworth, 1977] A.K. Mackworth. Consistency in Networks of Relations. *Artificial Intelligence*, 8:99–118, 1977.

[Mackworth, 1978] Alan Mackworth. Vision research strategy: Black magic, metaphors, mechanisms, miniworlds, and maps. In A. Hanson and E. Riseman, editors, *Computer Vision Systems*, pages 53–60. Academic Press, New York, 1978.

[McDermott, 1992] D. V. McDermott. Spatial reasoning. In Stuart C. Shapiro, editor, *Encyclopedia of Artificial Intelligence*, volume 1. John Wiley, 2nd edition, 1992.

[McKeown *et al.*, 1985] David M. McKeown, W. A. Harvey, and J. McDermott. Rule-based interpretation of aerial imagery. *IEEE Pattern Analysis and Machine Intelligence*, 7(5):570–585, September 1985.

[McKevitt, 1994] Paul McKevitt, editor. *Workshop on the Integration of Natural Language and Vision, AAAI-94*, 1994.

[Minsky, 1975] Marvin Minsky. A Framework for Representing Knowledge. In Patrick H. Winston, editor, *The Psychology of Computer Vision*, pages 211–277. McGraw-Hill Book Company, New York,NY, 1975.

[Mohr and Henderson, 1986] R. Mohr and Thomas C. Henderson. Arc and Path Consistency Revisited. *Artificial Intelligence*, 28:225–233, 1986.

[Mohr and Masini, 1988] R. Mohr and G. Masini. Good Old Discrete Relaxation. In *Proceedings of ECAI-88*, pages 651–656, 1988.

[Moore, 1966] R. E. Moore. *Interval Analysis*. Prentice Hall, 1966.

[Mulder *et al.*, 1988] Jan A. Mulder, Alan K. Mackworth, and William S. Havens. Knowledge Structuring and Constraint Satisfaction: The Mapsee Approach. *IEEE PAMI-10*, 10(6):866–879, 1988.

[Nevatia, 1978] R. Nevatia. Characterization and requirements of computer vision systems. In A. Hanson and E. Riseman, editors, *Computer Vision Systems*, pages 81–88. Academic Press, New York, 1978.

[Niemann *et al.*, 1990] H. Niemann, G. F. Sagerer, S. Schroder, and F. Kummert. Ernest: A Semantic Network System for Pattern Understanding. *Pattern Analysis and Machine Intelligence*, 12(9):883–905, 1990.

[Olivier, 1995] Patrick Olivier, editor. *IJCAI95 Workshop on the Representation and Processing of Spatial Expressions*, 1995.

[Papadias and Sellis, 1993] Dimitris Papadias and Timos Sellis. The semantics of relations in 2D space using representative points: Spatial indexes. In *Spatial Information Theory. A Theoretical Basis for GIS. European Conference, COSIT'93*, volume 716 of *Lecture Notes in Computer Science*, pages 234–247. Springer, September 1993.

[Pinhanez and Bobick, 1995] C. Pinhanez and A.F. Bobick. Using approximate models as source of contextual information for vision processing. In *Context95*, page xx, 1995.

[Price, 1985] K.E. Price. Relaxation matching techniques - a comparison. *PAMI*, 7(5):617–623, September 1985.

[Pylyshyn, 1981] Z. Pylyshyn. The imagery debate: Analog media vs tacit knowledge. In N. Block, editor, *Imagery*, pages 151–206. MIT Press, 1981.

[Reiter and Mackworth, 1989] R. Reiter and A.K. Mackworth. A Logical Framework for Depiction and Image Interpretation. *Artificial Intelligence*, 41:125–155, 1989.

[Roberts, 1965] L. Roberts. Machine perception of three dimensional solids. In J. Tippet and coworkers, editors, *Optical and Electro-optical Information Processing*, pages 159–197. MIT Press, Cambridge, Mass., 1965.

[Russell, 1979] D. M. Russell. Where do I look now? In *Proceedings of the PRIP*, pages 175–183, August 1979.

[Sam-Haroud and Faltings, 1996] D. Sam-Haroud and Boi Faltings. Consistency Techniques for Continuous Constraints. *Constraints Journal*, 1, 1996.

[Selfridge, 1982] Peter G. Selfridge. *Reasoning About Success and Failure in Aerial Image Understanding*. PhD thesis, U. of Rochester, 1982.

[Shirai, 1973] Y. Shirai. A context sensitive line finder for recognition of polyhedra. *Artificial Intelligence*, 4(2):95–119, 1973.

[Siskind and McAllester, 1993a] Jeffrey Mark Siskind and David Allen McAllester. Nondeterministic Lisp as a Substrate for Constraint Logic Programming. In *Proceedings of the Eleventh National Conference on Artifical Intelligence*, July 1993.

[Siskind and McAllester, 1993b] Jeffrey Mark Siskind and David Allen McAllester. Screamer: A Portable Efficient Implementation of Nondeterministic Commonlisp. Technical Report IRCS–93–03, University of Pennsylvania Institute for Research in Cognitive Science, 1993.

[Srihari and Burhans, 1994] Rohini K. Srihari and Debra T. Burhans. Visual Semantics: Extracting Visual Information from Text Accompanying Pictures. In *Proceedings of AAAI-94*, pages 793–798, 1994. Seattle, WA.

[Srihari *et al.*, 1994] Rohini K. Srihari, Rajiv Chopra, Debra Burhans, Mahesh Venkataraman, and Venugopal Govindaraju. Use of Collateral Text in Image Interpretation. In *Proceedings of the ARPA Workshop on Image Understanding*, pages 897–905, Monterey, CA, 1994.

[Srihari *et al.*, 1996] Rohini Srihari, Zhongfei Zhang, Mahesh Venkatraman, and Rajiv Chopra. Using Speech Input for Image Interpretation and Annotation. In *Proceedings of the ARPA Image Understanding Workshop*, Palm Springs, CA, February 1996.

[Srihari, 1991] Rohini K. Srihari. Extracting Visual Information from Text: Using Captions to Label Human Faces in Newspaper Photographs (Ph. D. thesis). Dept. of Computer Science Technical Report 91-17, State University of New York at Buffalo, 1991.

[Srihari, 1995a] Rohini K. Srihari, editor. *Computational Models for Integrating Language and Vision, AAAI-95 Fall Symposium Series*, 1995.

[Srihari, 1995b] Rohini K. Srihari. Computational models for integrating linguistic and visual information: A survey. *Artificial Intelligence Review (special issue on integration of NLP and Vision)*, 8(5):349–369, 1995.

[Srihari, 1995c] Rohini K. Srihari. Use of Collateral Text in Understanding Photos. *Artificial Intelligence Review (special issue on integration of NLP and Vision)*, 8(5):409–430, 1995.

[Strat and Fischler, 1991] Thomas M. Strat and Martin A. Fischler. Context-Based Vision: Recognizing Objects Using Information from Both 2-D and 3-D Imagery. *IEEE PAMI*, 13(10):1050–1065, 1991.

[Strat and Smith, 1988] Thomas Strat and Grahame Smith. Core knowledge system: Storage and retrieval of inconsistent information. In *Proceedings of the DARPA Image Understanding Workshop*, pages 660–665, Cambridge, MA, April 1988.

[Strat, 1992] Thomas M. Strat. *Natural Object Recognition*. Springer-Verlag, 1992.

[Topaloglou, 1993] T. Topaloglou. Storage management for knowledge bases. In *Proceedings of the 2nd International Conference on Information and Knowledge Management (CIKM-93)*, Washington, D.C., November 1993.

[Tsang, 1993] Edward Tsang. *Foundations of Constraint Satisfaction*. Academic Press, 1993.

[Tsotsos, 1985] J. Tsotsos. Knowledge organisation and its role in the interpretation of time-varying data:the alven system. *Computational Intelligence*, 1(1):16–32, 1985.

[Tsotsos, 1992] J.K. Tsotsos. On the relative complexity of active vs. passive visual search. *IJCV*, 7(2):127–141, January 1992.

[Waltz, 1975] D. Waltz. Understanding line drawings of scenes with shadows. In P.H. Winston, editor, *The Psychology of Computer Vision*. McGraw-Hill, New York, 1975.

[Weymouth, 1986] T.E. Weymouth. *Using Object Descriptions in a Schema Network for Machine Vision*. PhD thesis, University of Masschussetts at Amherst, 1986.

[Wixson and Ballard, 1994] Lambert E. Wixson and Dana H. Ballard. Using intermediate objects to improve the efficiency of visual search. *IJCV*, 12:209–230, 1994.

[Yuhan and Shapiro, 1995] Albert H. Yuhan and Stuart C. Shapiro. Computational representation of space. In Judith Duchan, Gail Bruder, and Lynne Hewitt, editors, *Deixis in Narrative: A Cognitive Science Perspective*, pages 191–226. Lawrence Erlbaum Associates, 1995.

[Zhang and Srihari, 1996] Zhongfei Zhang and Rohini Srihari. Qualitative building detection from monocular aerial images. Technical report, CEDAR, SUNY Buffalo, 1996.