CONSTRAINED CONTOURING IN THE POLAR COORDINATES

Shriram Revankar and David Sher

State University of New York at Buffalo

Buffalo, NY 14260

Abstract

A constrained contour is an outline of a region of interest, obtained by linking edges under the constraints of connectivity, smoothness, image context and subjective or user specified constraints. We discuss a constrained contouring algorithm in polar coordinates to trace closed contours. The algorithm finds optimal contour locations in all radial direction according to the constraints of context, distance from approximate contour and the image gradient. These optimal edge-points ordered according to their angular coordinates. We treat these edge points as the nodes of a graph of all possible contours. The links of the graph are weighted so that the shortest path between a pair of nodes is the smooth contour that traces maximum number of edge-points, and the shortest cycle in the graph gives the optimal contour.

1 Introduction

A contour is an outline a region of interest, and it is useful for the analysis of shape, form, size and motion of an object. Contouring is also useful for semantic interpretation of a scene. Although a contour is usually associated with traces of points of change in brightness and color gradients, sometimes it is also associated with the traces in the regions where there is no apparent change in image brightness or color, *e. g.* contours that extract the occluded objects, or the contours that extract textured regions. Furthermore, contours are sometimes subjective [1], and are influenced by the context of the image [2]. In this paper, we describe a method to trace closed contours of regions of interest. The contour is constrained by the connectivity and smoothness constraints, in addition to contextual constraints. The subjective or user specified constraints are provided by an approximate contour generated by the user or by an external process.

A conventional edge operator that detects the change in the brightness or color gradient, facilitates only a partial detection of a contour. Edge operators are regional, and it is difficult to design them to maintain global features such as shape and size. Therefore, an edge operator can not satisfactorily perform the tasks of contouring a desired region. However the output of a suitable edge operator can be used reduce the search space for selecting a satisfactory contour. The early work in contour tracing involved search for connectivity in the neighborhood of edge points and refining them [3, 4]. In several other methods, a contour is viewed as a path through a graph formed by linking the edge elements together according to heuristic linkage rules [5]. These methods may not yield the required boundaries because contours are context dependent, and subjective [1, 2]. Furthermore, the edge elements correlate only partially with the contours, because edge operators often detect false positives, and false negatives.

1.1 Deformable Contouring

To generate satisfactory contours researchers have proposed providing external constraints, and physical curvature restrictions to edge selection. Constraints are usually set according to the *a priori* domain knowledge. Kass, Witkins and Terzopoulus [6] suggested *snakes*, a *deformable contouring* model to select the best set of edges from the all possible edge locations, so that a set of global constraints are satisfied. The basic premise of their work is that an approximate contour is placed near the desired boundary by some other mechanism. Their use energy minimization methodology to refine the shape and location of the approximate contour under the constraints of connectivity, bending and domain dependent image features. Since our work closely relates to ideas of this work, we will briefly describe this work.

A contour is defined in the parametric form $V(s) = (x(s), y(s))$, s in some interval I. The energy of a contour $E_{contour}$ is defined as a summation over the entire contour, of three separate energy terms, namely, the internal energy term E_{int}, the external constraint term E_{ext} and the image energy term E_{image}.

$$E_{contour} = \int_I E_{int}(V(s)) + E_{image}(V(s)) + E_{ext}(V(s)) \tag{1}$$

The researchers have suggested finding a minimum energy ($E_{contour}$) contour by variational calculus technique [6]. A more efficient and robust way to achieve energy minimization by dynamic programming has been suggested by Amini, Weymouth and Jain [7]. A greedy algorithm for faster evaluation of the minimum energy contour is suggested by William and Shah [8].

In these methods the internal energy of the contour is defined by:

$$E_{int} = \alpha(s)|V_s(s)|^2 + \beta(s)|V_{ss}(s)|^2 \tag{2}$$

where $V_s(s)$ and $V_{ss}(s)$ are first and second order derivatives of the contour $V(s)$ with respect to the parameter s, and α, β determine the allowable bending and shearing at a point.

The image energy (E_{image}) is defined as a weighted combination of the three functionals:

$$E_{image} = w_{line}E_{line} + w_{edge}E_{edge} + w_{term}E_{term} \tag{3}$$

The weights were suitably chosen depending on the domain knowledge. The line energy term is proportional to the intensity of the point. For edge energy (E_{edge}) Kass *et al.* experimented with negative magnitudes of a simple gradient operator, and a zero crossing based operator. The termination energy (E_{term}) at a point is determined as the incremental change in the gradient in the direction perpendicular to the gradient direction.

The external constraint energy term E_{ext} is defined in terms of distances from the points suggested by a user. The energy magnitudes are proportional to square of the distances from the desired points, or are proportional to the negative square of the distance from an undesired point.

A similar work where *a priori* knowledge of the shape of the object is used to develop deformable templates has been reported by Yuille *et al.* [9].

In our *constrained contouring* algorithm we treat the above internal, external and image energy terms as the constraints on the possible contour location and shape, and we select a smooth contour that satisfies the constraints. To find the optimal contour we enumerate the all the contours that satisfy the constraint, and select the one that passes through the most number of locally optimal edge points. Our algorithm is efficient because we need to search only the space of contours that satisfy the constraints, instead of all possible contours.

2 Contouring in the Polar Coordinates

We transform the contour and the image features to the polar coordinates without any loss of data. Using the domain dependent constraints, and subjective or user specified constraints, we determine the optimal edge-points in all the radial direction. We use the third constraint of contour connectivity and smoothness to find a smooth contour that passes through the maximum number of optimal radial edge-points. To select such a contour efficiently we express the contour as a graph with nodes being edge points and links reflecting the constraints, and use modified shortest path algorithms to determine the optimal contour.

2.1 Motivation

The following observations motivated us to study the contouring problem in polar coordinates.

- many interesting contouring problems involve tracing closed contours enclosing a region.
- a large fraction of the regions of interest are of the shape such that radial lines from an internal point intersect the border only once (*e.g.* See Figure 3). Such contours form single valued curves in polar coordinates.
- The smoothest closed contour is a circle, with curvature or allowable bending being inversely proportional to the radius. Hence it is natural to visualize deforming a circle for a closed contour rather than deforming a line.
- Researchers have noted that shrinking and bunching-up of contour points where image forces are higher while processing in rectangular coordinates [8]. In polar coordinates, such problems can be avoided, because for a closed contour deformation for point need be given only in the radial direction. rotational, or angular deformation or freedom is not necessary.
- Often it is possible to design local edge detectors to locate the precise points of changes. If a smooth contour is made to pass through maximum number of these edge points, it will be precise at all the points where data is clear,

and false positive and false negative edge points will not contribute in the final contour positioning.

2.2 Transformation to polar coordinates

Let $P = \{p_1, p_2, p_3, \ldots, p_{N-1}, p_N\}$ be a simple polygon with vertices $p_i = (x_i, y_i)$, in rectangular coordinates, representing the initial contour input by some other process. To analyze in the polar coordinates, we transform the initial contour along with the image data to polar coordinates (r, θ) so that no data is lost.

For no loss transformation, we select the angular resolution:$\Delta\theta$, and the radial resolution:Δr, such that a unit translation in the rectangular coordinates at any point in the image causes at least a $\Delta\theta$ angular change or Δr radial change in the polar coordinates. If (1×1) is the pixel size in the rectangular coordinates, then $(\Delta\theta \times \Delta r)$ is the pixel size in polar coordinates.

To determine the angular unit, $\Delta\theta$ we inscribe the image in a circle of radius r_{max}, which is the maximal radial distance from the selected origin to the point in the image. Then we determine the angular resolution $\Delta\theta$ as the angle subtended by the radial lines passing through two neighboring pixels on the parameter of the circle. It is given by follows(see Figure 1a):

$$\Delta\theta = tan^{-1}(\frac{1}{r_{max}}) \tag{4}$$

where r_{max} is the maximum radial distance in the image from the selected origin.

We select the radial resolution Δr such that, for any angular deviations less than $\Delta\theta$ a translation of one pixel in rectangular coordinate causes a shift of at least one pixel in the polar coordinates. Without loss of generality, let $(0, 0)$ be the selected origin. The radial distance r_1 of a point (x_1, y_1) is $r1 = \sqrt{x_1^2 + y_1^2}$. For an angular

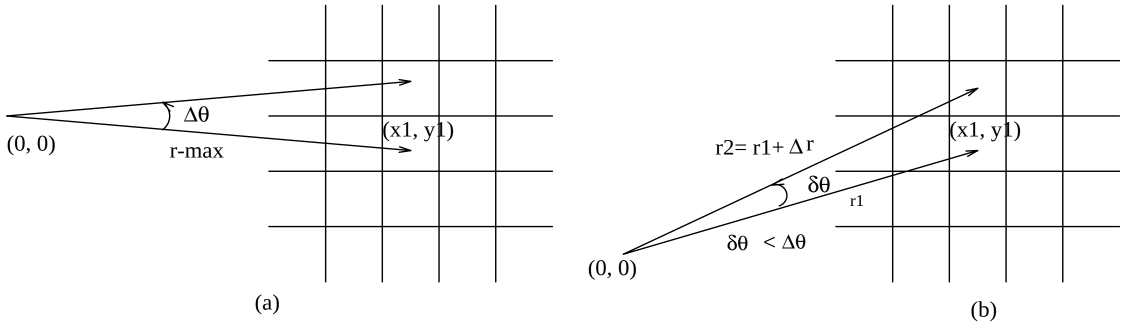


Figure 1: schematic describing selection of (a) angular, and (b) radial resolutions.

deviation $\delta\theta < \Delta\theta$ the smallest change in r is caused by a unit change in either x or y coordinate. Let the new point in the rectangular coordinate be $(x_1, y_1 + 1)$, and the corresponding radial distance is $r_2 = \sqrt{x_1^2 + (y_1+1)^2}$. Therefore the corresponding change in the radius is:

$$\Delta r = (r_2 - r_1) = \sqrt{x_1^2 + (y_1+1)^2} - \sqrt{x_1^2 + y_1^2}$$

This difference decreases with increase in x_1 and y_1, and we can show that for large x_1 or y_1 (See Figure 1b) it is bounded:

$$\Delta r = \lim_{x \ or \ y \longrightarrow (>>1)} \sqrt{x_1^2 + (y_1+1)^2} - \sqrt{x_1^2 + y_1^2} \geq \frac{1}{\sqrt{2}} \tag{5}$$

Therefore, by choosing $\Delta r = \frac{1}{\sqrt{2}}$ we cover all the data points that could have got merged by the selected angular resolution.

Using Δr and $\Delta\theta$ as the units in the polar coordinate we transform the input contour and the image data into polar coordinates. This assures no data loss. An example of the transformation is shown in Figure 2, where we transformed the image and contour by anti-clock wise scan starting from x-axis direction. The selected origin

Figure 2: Input contour on the original image and its polar coordinate transformation

is the point $(\overline{x}, \overline{y})$, and is given by:

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \tag{6}$$

$$\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{7}$$

If r is the distance of a point (x, y) from the point $(\overline{x}, \overline{y})$ and the line joining these two points subtends an angle θ, with respect to the x-axis, then it is transformed to polar coordinates by the relations, $x = r\cos\theta$, $y = r\sin\theta$. Since we will be concentrating mainly on closed contours, the selected origin is usually quite close to the center of gravity of the region.

Owing to the higher resolution of the polar domain, a continuous contour in cartesian domain can yield discontinuous contour in the polar domain. We connect such discontinuities by linear interpolation in the polar domain. This transformation and interpolation provides a contour in polar coordinates $[P' = \{p'_1, p'_2, p'_3, \ldots, p'_n\}, p'_i = (r_i, \theta_i)]$ such that contour points p'_i have equiangular separation of $(\theta_i - \theta_{i-1}) = \Delta\theta$.

2.3 Internal Constraints: Shearing and Bending

Nonrigid shape has been satisfactorily modeled in terms of *deformable models* based on the physical principles governing the dynamics of nonrigid bodies [10]. A contour outlining a nonrigid shape is therefore modeled as an object with membrane and thin plate like characteristics [6]. While the membrane property determines the *shearing* component, the thin plate property determines the bending component of the contour.

For digital contours in rectangular coordinates, researchers have used the distance between the adjacent points as the reasonable approximation of *shearing,* or the first order term $V_s(s)$ of the internal energy described in equation (2) [7, 8].

$$[V_s(s)]^2|_i = [x_i(s) - x_{i-1}(s)]^2 + [y_i(s) - y_{i-1}(s)]^2 \tag{8}$$

By substituting, $x_i = r_i \cos\theta_i,\ y_i = r_i \sin\theta_i$, we have

$$[V_s(s)]^2|_i = r_i^2 + r_{i-1}^2 - 2r_i r_{i-1}\cos(\theta_i - \theta_{i-1}) \tag{9}$$

We have transformed contour so that the angular difference between two consecutive contour points is $(\theta_i - \theta_{i-1}) = \Delta\theta$, which is small. Therefore $\cos(\theta_i - \theta_{i-1}) \longrightarrow 1$, and we rewrite the above equation as follows:

$$V_s(s) = \sqrt{r_i^2 + r_{i-1}^2 - 2r_i r_{i-1}} = (r_i - r_{i-1}) \tag{10}$$

Because all the adjacent contour points are separated by the angular difference of $\Delta\theta$, we can reasonably approximate $r'|_i = \frac{dr}{d\theta}|_i = r_i - r_{i-1}$. Therefore from the above equation(10) we have:

$$V_s(s)|_i = r'|_i \tag{11}$$

Bending at point on the curve is characterized by its curvature at that point. The second order term V_{ss} of internal energy equation (2) accounts for the curvature component. Curvature of a contour at a point is defined as the rate of change of inclination of the tangent to some base line ϕ, with respect to the arc length s, and is given by:

$$k = \frac{d\phi}{ds} \tag{12}$$

This relation can be written in polar coordinates as follows:

$$k = \frac{r^2 + 2r'^2 - rr''}{[r^2 + r'^2]^{3/2}} \tag{13}$$

where

$$r' = \frac{dr}{d\theta} \qquad r'' = \frac{d^2r}{d\theta^2}$$

For digital contours represented with equiangular points we can approximate $r'|_i$ by $r'_i = (r_i - r_{i-1})$ and $r''|_i$ by $r''|_i = (r_{i+1} + r_{i-1} - 2r_i)$. Therefore:

$$|r''_i| \leq 2 \cdot \max(|r'_{i+1}|, |r'_i|) \tag{14}$$

Unlike cartesian domain, in polar domain we can relate the smoothness with respect to size of the object. A large class of natural and artificial objects have shears or r' components that are much smaller than their size, in their digital representation. In Figure 3 we show some objects and their largest $\frac{r'}{r}$ ratio. These numbers can also be interpreted as the ratio of thickness (w) of the contour to the minimum radius of the object at a corner, because for a continuous contour $r' \leq w$. Therefore we can not make this statement if the contour thickness is comparable to the size or minimum diameter of the region it outlines. This assumption about the ratio is true for only those class of objects that do not have multiple radial values in any direction from

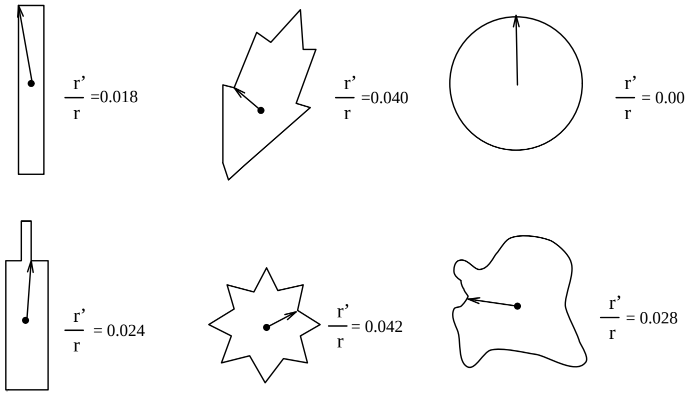


Figure 3: Digitized contours and their maximum $\frac{r'}{r}$ ratios.

the origin.

Hence we can safely assume that r' is much smaller than r, and so is r''. From equation 11 we have $V_s(s) \approx r'_i$, and from the curvature equation (13) we have:

$$\left. V_{ss}(s)\right|_i \approx \frac{1}{r_i} \qquad \frac{r'}{r} << 1 \tag{15}$$

Therefore in polar domain we can control the smoothness of a contour by constraining the ratio radial difference and the radius of curvature. For example in Figure 4 we

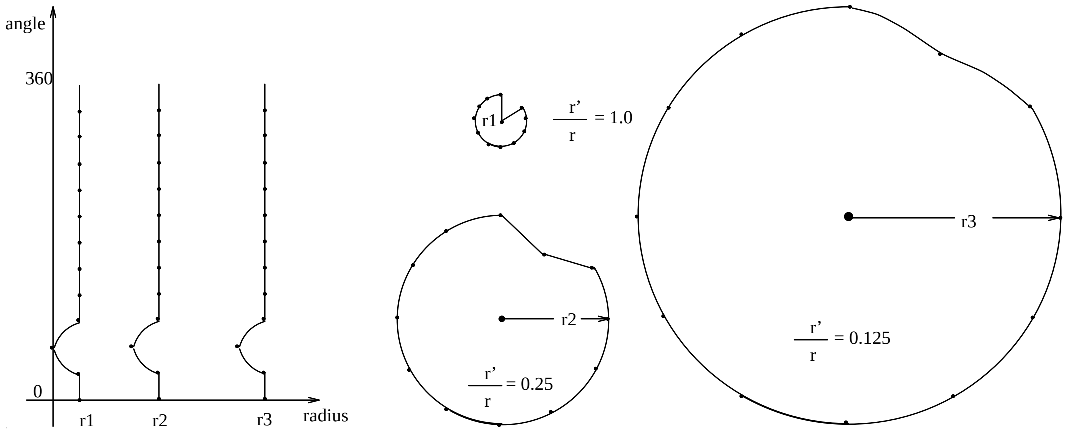


Figure 4: Deformed circles and their polar coordinate representations. All three circles have the same r' values.

show polar and cartesian representation of deformed circles of three different radii. In polar domain, all of them have the same r' values, but different r values, namely r1, r2 and r3. The smoothness of their cartesian representation is dependent on the $\frac{r'}{r}$ ratios. A constraint on this ratio, $\tau_i = \frac{r'_i}{r_i}$ is used in the selection of the optimal constrained contour described in sections 3 and 4.

2.4 External or Contextual Constraints

External constraints can be domain dependent, or can be derived from the process that places the rough contour. For example in supervised segmentation [11] the user observes an approximate segmentation and circles a contour to refine the segmentation. The edges of the reference segmentation can provide an external constraint. If (r_s, θ) is the point on the boundary of the approximate segmentation, then the likelihood of locating the refined contour at location (r, θ) is given by:

$$E_s(r,\theta) = \begin{cases} 1 \quad if \quad r = r_s \\ \frac{c}{(r_s - r)^2} \quad otherwise \end{cases}$$

where c is an empirical constant.

Similarly we define $E_c(r, \theta)$, the likelihood of locating the final contour as constrained by the distance from the input contour.

Further, if we have knowledge about the external contouring process, that can also provide us an additional constraint, $E_I(r, \theta)$. For example, in interactive contouring, we use human *attention distribution map* [12, 13] borrowed from psychophysical studies to constrain the positions taken by the final contour [11].

In addition, we can add contextual constraints for edge selection. For example in iterative refinement is envisaged, we can make the weightage to $E_c(.)$ proportional to the level of iteration.

We define the total likelihood of locating the final contour at location (r, θ) as the sum of these likelihoods.

$$E_{ext}(r,\theta) = E_s(r,\theta) + E_c(r,\theta) + E_I(r,\theta) \tag{16}$$

2.5 Image or Domain Dependent Constraints

Image constraints are the gradient of the image, line, termination, etc. These are clearly domain dependent. While in line images, and un textured images, simple gradient operator can provide edges, in textured images, primitives like termination and corners may not be effective. In medical images, corner detection is often not necessary.

Further, a large number of image domains generate objects of interest brighter or darker than the background. For such images, the gradients being considered can be limited to either only increasing gradients or decreasing gradients as observed from the origin in the polar domain. This would help contouring along a thin object, so that the final contour does not alternate depending on the gradient strength.

Let the total likelihood of locating the optimal contour at location (r, θ) owing to image constraints be $E_{image}(r, \theta)$.

2.6 Selection of Regional Edge-Points

A point (r, θ) on the input contour can only be deformed to the points in the radial direction θ. These points form the neighborhood of deformation of (r, θ).

The locally optimal points for the final contour at $Q = \{q_1, q_2, \ldots, q_n\}, q_i = (r_i, \theta_i)$, are given by:

$$q_i = (r_i, \theta_i) \ni \forall_{j \in [0,\dots,r_{max}]} [E_{image}(r_i, \theta_i) + E_{ext}(r_i, \theta_i)] \geq [E_{image}(r_j, \theta_i) + E_{ext}(r_j, \theta_i)] \tag{17}$$

3 Characterization of the Optimal Contour

Given an ordered set of constrained regionally optimal points $Q = \{q_1, q_2, \dots, q_n\}$, we define the optimal constrained contour as the *smoothest contour that passes through the maximum number of regionally optimal points* q_i*, while satisfying the smoothness constraint* $\tau(i)$*.*

To visualize the intuitive idea behind this characterization, in Figure 5 we show a schematic of the optimal contour for the displayed set of points. In the figure, solid line

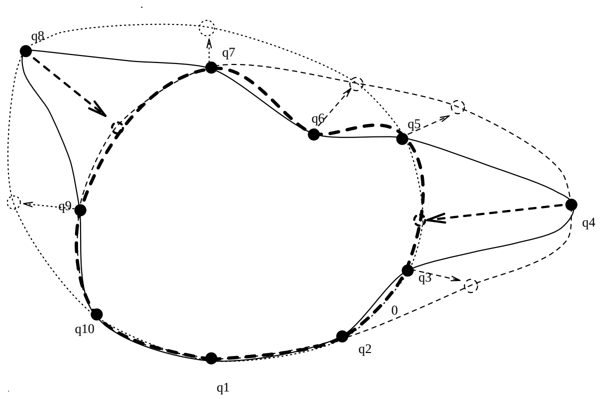


Figure 5: Schematic of the optimal contour in a graph of edge-points.

contour is the rough contour that does not satisfy the smoothness constraint. There are three broken line contours that include maximum possible number of points while satisfying the imposed smoothness constraints. Two of them are shown in thin broken lines, and both includes 6 points. The third smooth contour excludes these two points and passes through 8 points. This contour is considered optimal.

We solve this problem of selecting the optimal contour by arranging the points q_i

as the nodes of a fully connected graph. Each link of the graph is weighted equal to the number of points it skips if it were selected in the chosen path, and if the link satisfies the smoothness constraint, otherwise the link get ∞ weight. The shortest cycle of such a graph gives a contour that is smooth and skips the least number of nodes. That contour is selected as the optimal contour.

Optimal Contour: let $G = (V, E, W)$ be the graph with a set of vertices $V = \{q_1, q_2, q_3, \ldots, q_{n-1}, q_n\}$ be a set of optimal edge points located at increasing angles with respect to the origin and a reference radial direction. Let $E = \{E_{i,j} | i, j \in [1 \ldots n]\}$ be the set of links connecting the vertices and $W = \{W_{ij} | i, j \in [1 \ldots n]\}$ be their weights. Let Π be the set of all ordered subsets of V (set of valid paths in V).

The problem is to find a subset $P_o \in \Pi$, $P_o = \{q_i, q_{i+k_1}, q_{i+k_2}, q_{i+k_3}, \ldots, q_{i+k_l}\}$, where $(k_1 < k_2 < k_3 \ldots < k_l)$, such that:

1. a closed curve joining $(q_i \Longrightarrow q_{i+k_1} \Longrightarrow q_{i+k_2} \Longrightarrow \ldots q_{i+k_l} \Longrightarrow q_i)$ satisfies the smoothness constraints.

2. $|P_o| \geq |P_a| \; \forall P_a \in \Pi$.

We call the closed curve of P_o, the *optimal contour.*

4 Optimal Path Algorithm

To find the optimal path we assign weight $w_{ij} \in W$, to each link $e_{ij} \in E, \forall_{i,j \in [1..n]}$ as follows:

$$W_{ij} = \begin{cases} |i-j| - 1 & if \; [\frac{(r_i - r_j)}{r_i}] < (\tau(i) \cdot |i-j|) \\ \infty & o/w \end{cases} \tag{18}$$

We show examples of this weighting scheme in Figure 6. According to equation (18), the links of the graph have 0 value if they are adjacent to each other and satisfy the smoothness, τ constraint. If a link skips w number of nodes, in the anti-clock wise direction then the weight of that node is w. In Figure 6, the left side graph represents

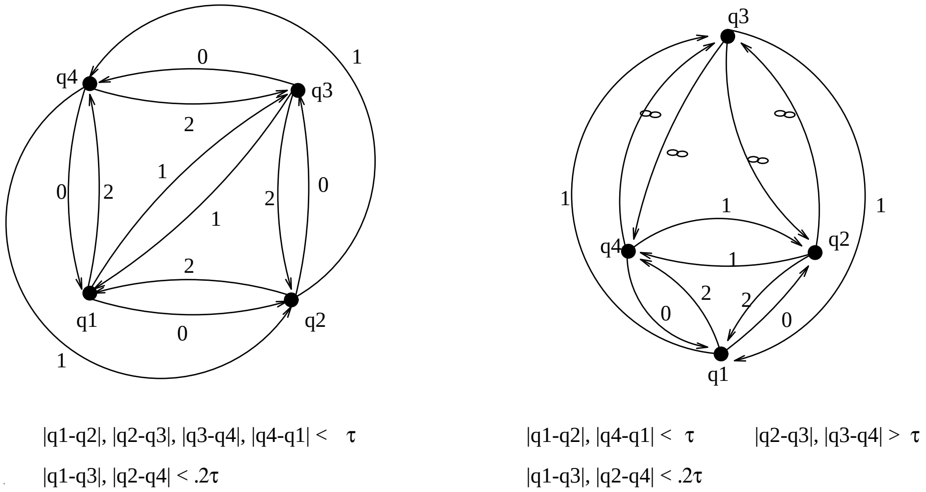


Figure 6: A schematic of graph showing the assignment of weights.

the nodes that completely satisfy the smoothness constraint, therefore all the links to anti-clock wise successors are of zero weight, and the rest of the arcs have the weight equal to number of nodes they bypass. In the right side graph, the point q_3 does not satisfy the smoothness constraints with its neighbors, however the link $q_1 \Longrightarrow q_3$ satisfies the smoothness constraint. So the links between those two points have unit weight, indicating that each link bypasses on intermediate point.

To find the shortest cycle starting from q_1, we split the node into two nodes q' and q'', with the same (r, θ) value. q'_1 is used as the starting node, and it inherits all the outgoing links of q_1. q''_2 inherits all the incoming arcs of q_1, and in addition a new link of weight $n-1$ links q'_1 to q''_1, indicating that the link skips $n-1$ number of intermediate nodes. Now the shortest path from q'_1 to q''_1 gives the shortest cycle that includes the point q_1. Therefore we find the optimal contour containing the node q_1, using the following modified shortest path algorithm.

Algorithm:

- *Input:* $G = (V, E, W)$ is a graph
- *Output:* $P = \{P_1, P_2, \ldots, P_n, P_{n+1}\}$ are the lists that hold the trace of the vertices, and $L = \{L_1, L_2, L_3, \ldots, L_n, L_{n+1}\}$ are the sum of the weights of the corresponding traces. P_{n+1} holds the shortest cycle.

Conditional-Closed-Contour(G, P, L)
$\forall i, L_i = \infty$
$L_1 = 0$
$P_1 = q_1$
$q_{n+1} = q_1''$ for each $q_i \in \{q_2, q_3, q_4, \ldots, q_n, q_{n+1}\}$
    for each $q_j \in \{q_1, q_2, \ldots, q_{i-1}\}$
        if $(L_j + W_{ij}) < L_i$ then
            $L_i = L_j + W_{ij}$
            $P_i = P_j$
            $P_i = append(P_i, q_j)$
        endif
    endfor
endfor

claim 1: The algorithm returns P_{n+1}, the shortest cycle that passes through q_1.

proof: Follows directly from the correctness of the shortest path algorithms for directed acyclic graphs. The algorithm finds the shortest path between the nodes q_1' and q_1'', and both q_1' and q_1'' represent the same point, therefore the shortest path is a cycle. $\square$

claim 2: The algorithm returns a smooth closed contour that passes through maximum possible number of edge points starting from q_1.

proof:

1. The algorithm returns a closed contour : from (*claim 1*).
2. The contour is smooth because:

(a) all finite weight links in the graph are constrained by specified smoothness factor τ.

(b) the contour passes through finite weight links only, if $w_p = \sum_i w_{pi}$ is the total weight of the path,

$$w_p \leq (n-1)$$

because the weight of the link $q_1' \longrightarrow q_1''$ is $(n-1)$.

3. The contour passes through maximum number of points: (proof by contradiction)

   - Let the contour P_{n+1} pass through Λ number of points. Let there be a path P' with total weight $w_{P'}$, between q_1' to q_1'', that passes through Λ', number of nodes, and $(\Lambda' > \Lambda)$.

     From equation (18) w_p, the number of nodes skipped along the path from q_1' to q_1''.

     $\Longrightarrow \Lambda = n - w_P, \quad and \quad \Lambda' = n - w_{P'}$

     $\Longrightarrow w_P > w_{P'}$

     $\Longrightarrow$ P' is shorter than the shortest path - **(contradiction)**. $\square$

This algorithm iterates n times, with iteration i having a nested iteration of size $i-1$. Clearly, the number of operations is asymptotically proportional to $\frac{n(n-1)}{2}$, and therefore the algorithm is of asymptotic complexity $O(n^2)$, with n proportional to the length of the contour.

The algorithm generates an optimal contour given that q_1 belongs to the optimal contour. This may not be true. Therefore we run this algorithm with all the points as the source points, and pick the contour that passes through the maximum number of edge-points. The following algorithm describes the selection of the optimal contour:

Algorithm:

- *Input:* $G = (V, E, W)$ is a graph
- *Output:* P_{opt} is the optimal path.

Optimal-Contour(G, P_{opt})
$P_{opt} = \{\}$
length=$|P_{opt}|$
for each $q_i \in \{q_1, q_2, q_3, q_4, \ldots, q_n, q_1'\}$
    Generate a graph G_i with q_i as the starting node
    Conditional-Closed-contour(G_i, P)
    if(length $< |P_{n+1}|$)
        length $= |P_{n+1}|$
        $P_{opt} = P_{n+1}$
    endif
endfor

This algorithm iterates the $O(n^2)$ algorithm n times, and thus it is a $O(n^3)$ algorithm.

5 Modification for Known Discrete Neighborhood

Sometimes it is possible to know the range of movement for point. For example, if the image is of heart, knowing the dynamics of heart motion, it is possible to know the limits displacement of a wall section. If for a point q_i on the contour, we know a set of values $\{q_{i1}, q_{i2}, q_{i3}, q_{i4}, \ldots, q_{im}\}$ that it can take, then we can rewrite the optimal-contour algorithm with reduced complexity.

Optimal-Contour(G, P_{opt})
$P_{opt} = []$
length$=|P_{opt}|$
for each $q_{ij} \in \{q_{i1}, q_{i2}, q_{i3}, \ldots, q_{im}\}$
 Generate a graph G_{ij} replacing q_i by q_{ij}
 Conditional-Closed-contour(G_{ij}, P)
 if(length $< |P_{n+1}|$)
  length $= |P_{n+1}|$
  $P_{opt} = P_{n+1}$
 endif
endfor

This is clearly a $O(mn^2)$ algorithm, and for small m's this provides considerable speedup. This algorithm can be improved further if possible neighborhoods of all the points on the contour were known. We can use the following $O(m^2n)$ algorithm.

Optimal-Contour(V, P_{opt})
$\forall_{ij} W_{ij} = \infty$
$\forall_j W_{1,j} = 1, j$ in the neighborhood of q_1
$W_{1,j} = 0$ if $q_{1j} = q_1$
$q_{n+1} = q_1$
for each $q_i \in \{q_2, q_3, \ldots, q_n, q_{n+1}\}$
 for each q_{ij} in the neighborhood of q_i
  for each $q_{i-1,k}$ in the neighborhood of q_{i-1}
   if $|q_{ij} - q_{i-1,k}| < \tau i \cdot r_i$
    if $q_{ij} = q_i$
     if $W_{ij} > W_{i-1,k}$
      $W_{ij} = W_{i-1,k}$
      $P_{ij} = append(P_{i-1,k}, \{q_{ij}\})$
     endif
    else
     if $W_{ij} > W_{i-1,k} + 1$
      $W_{ij} = W_{i-1,k} + 1$
      $P_{ij} = append(P_{i-1,k}, \{q_{ij}\})$
     endif
    endif
  endfor
 endfor
endfor
$P_{opt} = P_{n+1,j}$ if $W_{n+1,j} \geq W_{n+1,i}, \forall i \in$ neighborhood of q''_1

If there are two or more contours with same number of points, we pick the contour that bends the least. We compute the total $|r''|$ values of all the maximal contours and pick the contour that has the minimum sum.

6 Results

Our algorithm provides promising results on real data. We have run this algorithm on ultrasound images of the human heart to trace the boundaries of the heart chambers, and also on optical images of objects of varying smoothness. The displayed results are for constant smoothing constraint τ.

6.1 Contouring in Ultrasound images

In ultrasound images objects of interest are brighter than the background, and the objects are with smooth contours. So we set τ small so that contour traced is smooth. In addition, we need to trace either increasing or decreasing image gradients. A contour can not trace both decreasing and increasing contours. In Figure 7 we show contouring of left ventricular chamber of a long-axis two dimensional echocardiogram. The left image shows the original image and the contoured image is shown in the right. The middle image shows the contour generated by simply connecting the locally optimal edge points.

In Figure 8 we show input output pair of a short-axis two dimensional echocardiogram displaying a different cross section of the left ventricular chamber. The algorithm traces smooth chamber walls even through the valves and papillary muscles which occlude the wall contours. Further more, in the places where no gradient is found, and in the places where smoothness criterion is not met, the contour interpolates the region with smooth circular arcs, because our algorithm traces the contour

Figure 7: Input contour on the original image (left), The contour generated by connecting locally optimal edge points (middle), and the trace of the optimal contour generated by setting a constant smoothness constraint.

without radial changes at rough regions.

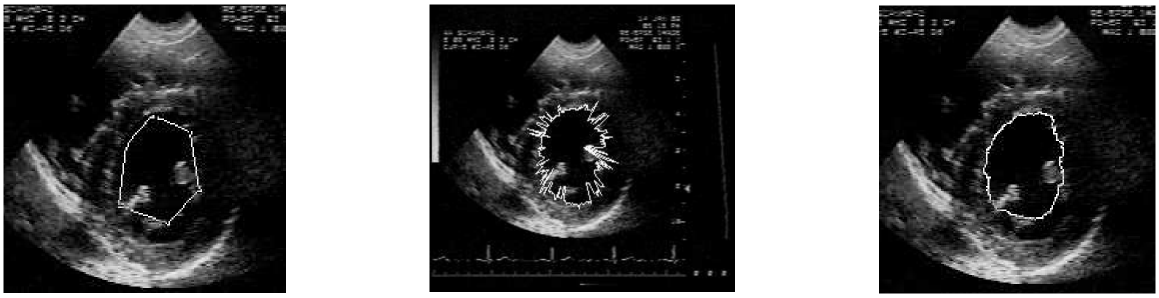


Figure 8: Long-axis two-dimensional echocardiograms with an input contour (left) the traced contour.

Our algorithm can be used to trace the contours of a sequence of images of a dynamic scene. In Figure 9 we show the contours generated by tracing the contours of a sequence of short axis view, two dimensional echocardiograms. A rough contour is input in one of the frame. The optimal contour of that frame is used as the rough initial contour for the neighboring frames. The procedure is repeated for generating the contours in all the frames.

6.2 Contouring in Optical Images

In Figure 10 we show the results of our contouring algorithm on two optical images. Both images have various textures and grey level gradients. The left most images in each row are the original images. The second image in each row shows the rough

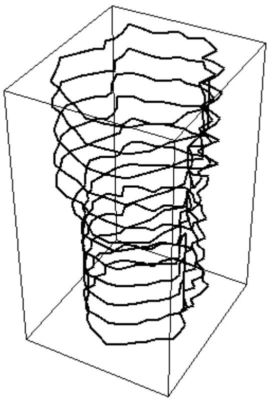
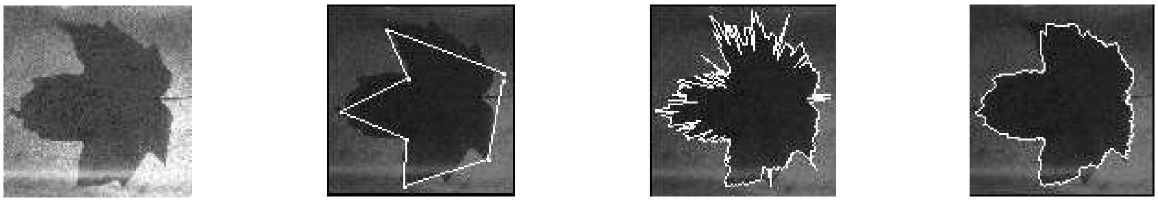


Figure 9: Contours of a sequence of short axis views of the left ventricle in two dimensional echocardiograms.

input contour and the third images are the contour joining the locally optimal contour points without smoothness constraints. The final images are generated by tracing the optimal contour that satisfy a constant smoothing constraint.



a. Stages in contouring an optical image of a leaf.

b. Stages in contouring the optical image of a shoe.

Figure 10: The left images show the original images and the images in the second column show the rough initial contour. The images in the third column show the trace of locally optimal contour points, and the right most images show the optimal contour traced by setting a constant smoothing constraint.

7 Conclusions

Tracing salient contours of regions of interest in two dimensional images is important for analysis of shape, size and nonrigid motion of the region. Simple edge detectors cannot satisfactorily extract the contours, in the absence of global knowledge of the shape and related contextual knowledge. Researchers have proposed several energy minimization and edge linking algorithms to detect salient contours. In this paper we have described a contouring algorithm that facilitates the use of several higher level constraints to govern edge selection and linking.

In addition to providing an interesting alternative method to trace the contour, we also study bending and shearing of contours in polar coordinates. The polar coordinate transformation has some inherent advantages, such as providing quadratic interpolation of missing traces, relating the roughness with the size, and avoiding unexpected behaviors such as shrinking. This process can be viewed as being equivalent to deforming a circle, and it is natural for tracing closed contours.

References

[1] Leonard Zusne. *Visual Perception of Form*, chapter Contemporay theory of visula form perception:I. The psychology of contour perception, pages 16–20. Academic Press, 1970.

[2] G.T.Toussaint. The use of context in pattern recognition. *Pattern Recognition*, 10:189–204, 1978.

[3] A. Rosenfeld and A.C. Kak. *Digital Picture Processing*. New York: Academic Press, 1976.

[4] A. Martelli. Edge detection using heuristic search methods. *Computer Graphics and Image Processing*, 1:169–182, 1972.

[5] A.K. Jain. *Fundamentals of Digital Image Processing.* Prentice-Hall, 1989.

[6] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. In *Proceedings: IEEE 1st International Conference on Computer Vision, London England, June 8-11*, pages 259–267, 1987.

[7] Amir A. Amini, Terry E. Weymouth, and Ramesh C. Jain. Using dynamic programming for solving variational problems in vision. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 12(9):855–867, September 1990.

[8] Donna J. Williams and Mubarak Shah. A fast algorithms for active contours and curvature estimation. *CVGIP, Image Understanding*, 55(1):14–26, January 1992.

[9] Alan Yuille, David Cohen, and Peter Hallinan. Facial feature extraction by deformable templates. Technical Report 88-2, Harvard Robotics Laboratory, 1988.

[10] Terzopoulos D. and Fleischer K. Deformable models. *The Visual Computer*, 4(6):306–331, 1988.

[11] Shriram Revankar and David Sher. Supervised image segmentation. Technical report, Department of Computer Science, State University of New York at Buffalo, January 1992.

[12] Cathryn J. Downing. Expectancy and visual-spatial attention: Effects on perceptual quality. *Journal of Experimental Psychology: Human Perception and performance*, 14(2):188–202, 1988.

[13] G.R.R.Mangun and A. Hillyard. The spatial allocation of visual attention as indexed by event related brain potentials. *Human Factors*, 29(2):195–211, 1987.