Deformable Model Segmentation

June 29, 1992

David L. Wilson
Lindseth Assistant Professor

Biomedical Image Processing Laboratory
Department of Biomedical Engineering
Case Western Reserve University
Cleveland, OH 44106

Outline

- Introduction
- Segmentation with locally deformable models
- Method of calculation
- SCR approach
- Global models and segmentation
- Potential modifications

Characteristics of Ventricular Segmentation Methods

Score segmentation methods on a scale of 1-5.

Humans: 35

- 5 **knowledge of a single closed contour**
- 5 **edge tracking**
- 5 **regional uniformity tracking**
- 5 **smoothness of edges**
- 5 **knowledge of size**
- 5 **knowledge of orientation**
- 5 **knowledge of shape**

Conventional region growing: 9

- 4 **knowledge of a single closed contour**
- 0 **edge tracking**
- 5 **regional uniformity tracking**
- 0 **smoothness of edges**
- 0 **knowledge of size**
- 0 **knowledge of orientation**
- 0 **knowledge of shape**

Characteristics of Ventricular Segmentation Methods

Score segmentation methods on a scale of 1-5.

Bill Connor's modified watershed: 16

- 5 **knowledge of a single closed contour**
- 2 **edge tracking**
- 5 **regional uniformity tracking**
- 2 **smoothness of edges**
- 0 **knowledge of size**
- 0 **knowledge of orientation**
- 2 **knowledge of shape**

Locally deformable models: 21

- 5 **knowledge of a single closed contour**
- 5 **edge tracking**
- 3 **regional uniformity tracking**
- 5 **smoothness of edges**
- 0 **knowledge of size**
- 0 **knowledge of orientation**
- 3 **knowledge of shape**

Characteristics of Ventricular Segmentation Methods

Score segmentation methods on a scale of 1-5.

Globally and locally deformable model: 31

- 5 **knowledge of a single closed contour**
- 5 **edge tracking**
- 4 **regional uniformity tracking**
- 5 **smoothness of edges**
- 4 **knowledge of size**
- 4 **knowledge of orientation**
- 4 **knowledge of shape**

Deformable Models

Locally deformable model:

- local elasticity
- no constraint on global shape
- algorithms in 2D: snakes and balloons

Globally deformable model:

- 3D volume mathematical models such as superquadrics
- useful for graphic representations
- may introduce knowledge of shape, size, and orientation to segmentation
- does not allow local deformation

Spatially varying material properties:

- also allows a priori shape information
- allow corners

References:

1. Kass, Michael; Andrew Witkin; and Demetri Terzopoulos; "Snakes: active contour models;" Int. Journ. of Comp. Vision; 1988; 321-331
2. Cohen, Laurent D.; "Note on active contour models and balloons;" CVGIP; 1991; 53:211-218.
3. L. H. Staib and J. S. Duncan, Boundary finding with parametrically deformable models. in Advances in Image Analysis. Y. Mahdavieh and R. C. Gonzalez, eds. SPIE Press. 1992.

Locally Deformable Model: Rationale

- Includes a smoothness constraint.
- Can require a single closed curve (a contour). This constraint is useful for making biological *measurements*.
- Traditional segmentation uses a bottom-up approach: edge filtering -> edge linking -> edge smoothing, etc. (Only sometimes create a top-down path that incorporates knowledge of the applications.)
- Locally deformable models using an energy minimization approach use more *global* information in the early segmentation process.
- Provides a frame-work for combining higher-level knowledge. For example, use anchor point forces or change material properties (smoothness parameter), etc.

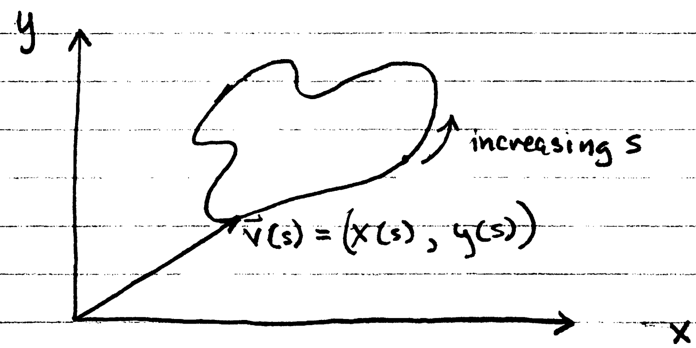
The deformable model approach is very similar to dynamic programming (DP) described earlier. In fact, some solve "deformable model" segmentations using DP.

II. Theory of Locally Deformable Model Segmentation (Snakes + Balloons)

A. Energy formulation

1. position in 2D image is obtained parametrically

$\vec{v}(s) = (x(s), y(s))$ where $\vec{v}(s)$ is the position along the curve given as a function of s. In the continuous case, we will parameterize s between 0 and 1.



2. Energy function

$$E^*_{snake} = \int_0^1 E_{snake}(v(s))\, ds \tag{1}$$

$$= \int_0^1 E_{int}(v(s)) + E_{image}(v(s)) + E_{constraint}(v(s))\, ds$$

a.) E_{int} includes the local elasticity of the curve - internal spline energy

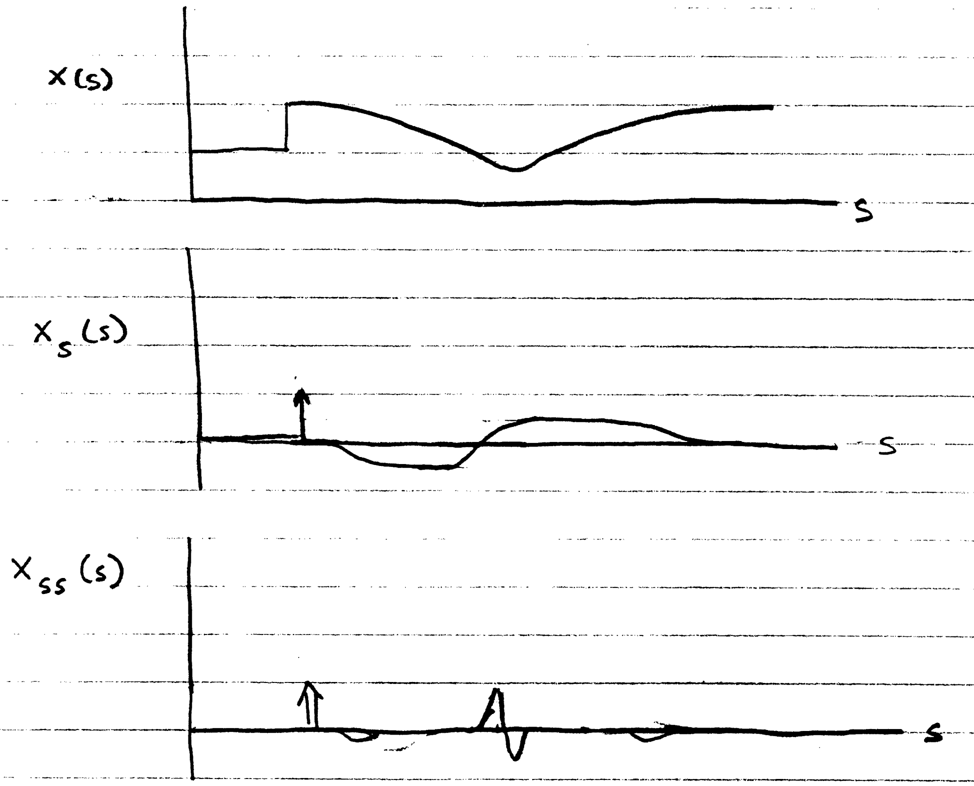
$$E_{int} = \underbrace{\alpha(s)\left|V_s(s)\right|^2}_{\text{membrane}} + \underbrace{\beta(s)\left|V_{ss}(s)\right|^2}_{\text{thin plate}} \tag{2}$$

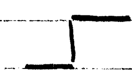
where $V_s(s)$ is the first derivative

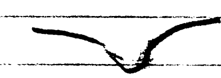
$V_{ss}(s)$ " " second "

$\alpha(s) + \beta(s)$ determine the weighting and can, in general, depend upon s

Example curve:



high $\alpha(s)$ $\Rightarrow$ no 

high $\beta(s)$ $\Rightarrow$ no  and reduced bending 

Note that if we simply make $x_s(s)$ high then we will preferentially track lines closer to the horizontal.

If $\alpha(s)$ and $\beta(s)$ are made zero at a position s, then a corner can be made.

b) E_{image} is the image force. It includes attraction forces of desirable features in the image.

$$E_{image} = \begin{cases} I & (\text{low-valued edges}) \\ -(\nabla I)^2 & (\text{edges} - 1^{st}\text{ derivative}) \quad \text{Sobel} \\ (\nabla^2 I)^2 & (\quad " \quad - 2^{nd}\text{ derivative}) \quad \text{LOG, DOG} \end{cases}$$

Image forces can be very domain-dependent.

Low values of E_{image} are "desireable."

c) $E_{constraint}$ are externally-imposed constraint forces.

Kass et al use:

- repulsive forces (volcanoes)
- attraction forces (springs)

These are given for user interaction (snake pit). Forces can be $1/r^2$ repulsion or attraction, or kx attraction.

Cohen introduces a balloon inflation force.

III. Method of Calculation

$$E^*_{snake} = \int_0^1 E_{int}(v(s)) + \underbrace{E_{image}(v(s)) + E_{constraint}(v(s))}_{E_{ext}} \, ds$$

$$E^*_{snake} = \int_0^1 E_{int}(V(s)) + E_{ext}(v(s)) \, ds \qquad (5)$$

Recall that $\vec{V}(s) = (x(s), y(s))$.

We wish to minimize E^*_{snake}. Minimizing an integral requires "calculus of variations". Hildebrand, "Methods of Applied Mathematics" is a good reference.

Assuming $\alpha(s)$ and $\beta(s)$ to be constants, we obtain the Euler equations:

$$\alpha x_{ss} + \beta x_{ssss} + \frac{\partial E_{ext}}{\partial x} = 0 \qquad (6)$$

$$\alpha y_{ss} + \beta y_{ssss} + \frac{\partial E_{ext}}{\partial y} = 0$$

In discrete form, Equation 5 is writen below where i indexes around the curve.

$$E^*_{snake} = \sum_{i=0}^{n} E_{int}(i) + E_{ext}(i)$$

Using finite differences, Equation 2 becomes

$$E_{int}(i) = \alpha_i |v_i - v_{i-1}|^2 / 2h^2 + \beta_i |v_{i-1} - 2v_i + v_{i+1}|^2 / 2h^4.$$

A closed curve is obtained by requiring $V(0) = V(n)$.

(Equation 6)

The corresponding Euler equations are:

$$
\begin{aligned}
&\alpha_i (v_i - v_{i-1}) - \alpha_{i+1} (v_{i+1} - v_i) && \text{a second derivative approximation} \\
&\quad + \beta_{i-1} [v_{i-2} - 2v_{i-1} + v_i] \\
&\quad - 2\beta_i [v_{i-1} - 2v_i + v_{i+1}] && \text{a 4th derivative approximation} \\
&\quad + \beta_{i+1} [v_i - 2v_{i+1} + v_{i+2}] \\
&\quad + (f_x(i), f_y(i)) = 0
\end{aligned}
$$

where f_x and f_y are $\partial E/\partial x$ and $\partial E/\partial y$, respectively.
Note that the second derivative is obtained by taking the first derivative of a first derivative. The 4th is obtained from the 2nd of a 2nd. Step sizes are unity.

Splitting into x and y components and writing in matrix form, we obtain

$$\underline{A}\,\underline{x} + \underline{f_x}(x,y) = 0 \tag{15}$$

$$\underline{A}\,\underline{y} + \underline{f_y}(x,y) = 0 \tag{16}$$

where

$$
\underline{A}\,\underline{x} = \begin{bmatrix}
K_0 & K_1 & K_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
K_{-1} & K_0 & K_1 & K_2 & 0 & 0 & 0 & 0 & 0 & 0 \\
K_{-2} & K_{-1} & K_0 & K_1 & K_2 & 0 & 0 & 0 & 0 & 0 \\
0 & K_{-2} & K_{-1} & K_0 & K_1 & K_2 & 0 & 0 & 0 & 0 \\
0 & 0 & K_{-2} & K_{-1} & K_0 & K_1 & K_2 & 0 & 0 & 0 \\
0 & 0 & 0 & K_{-2} & K_{-1} & K_0 & K_1 & K_2 & 0 & 0 \\
0 & 0 & 0 & 0 & K_{-2} & K_{-1} & K_0 & K_1 & K_2 & 0 \\
0 & 0 & 0 & 0 & 0 & K_{-2} & K_{-1} & K_0 & K_1 & K_2 \\
0 & 0 & 0 & 0 & 0 & & K_{-2} & K_{-1} & K_0 & K_1 \\
0 & 0 & 0 & 0 & 0 & & & K_{-2} & K_{-1} & K_0
\end{bmatrix}
\begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ \vdots \\ \\ \\ \\ \\ x(8) \\ x(9) \end{bmatrix}
$$

for the case of $n=9$, and where $\underline{f_x} = \dfrac{\partial E_{ext}}{\partial x}$ = force and

$$
\left.\begin{aligned}
K_{-2} &= \beta_{i-1} \\
K_{-1} &= -[\alpha_{i-1} + 2\beta_i + 2\beta_{i-1}] \\
K_0 &= \alpha_i + \alpha_{i+1} + \beta_{i-1} + 4\beta_i + \beta_{i+1} \\
K_1 &= -[\alpha_{i+1} + 2\beta_i + 2\beta_{i+1}] \\
K_2 &= \beta_{i+1}
\end{aligned}\right\} \text{should be checked}
$$

If $\alpha_j = \alpha$ and $\beta_j = \beta$, then

$$
\begin{aligned}
K_{-2} &= \beta \\
K_{-1} &= -[\alpha + 4\beta] \\
K_0 &= 2\alpha + 6\beta \\
K_1 &= -[\alpha + 4\beta] \\
K_2 &= \beta
\end{aligned}
$$

Equations 15 and 16 are obtained when the energy is minimized. To find the minimum, we use an iterative method. For an iteration from $t-1$ to t, we get

$$\underline{A}\,\underline{x}_t + \underline{f}_x(\underline{x}_{t-1}, \underline{y}_{t-1}) = \gamma(\underline{x}_t - \underline{x}_{t-1}) \qquad (17)$$

$$\underline{A}\,\underline{y}_t + \underline{f}_y(\underline{x}_{t-1}, \underline{y}_{t-1}) = \gamma(\underline{y}_t - \underline{y}_{t-1}) \qquad (18)$$

Equations 17 and 18 are solved by matrix inversion.

$$\underline{x}_t = (\underline{A} + \gamma \underline{I})^{-1}\left[-\gamma \underline{x}_{t-1} - \underline{f}_x(\underline{x}_{t-1}, \underline{y}_{t-1})\right]$$

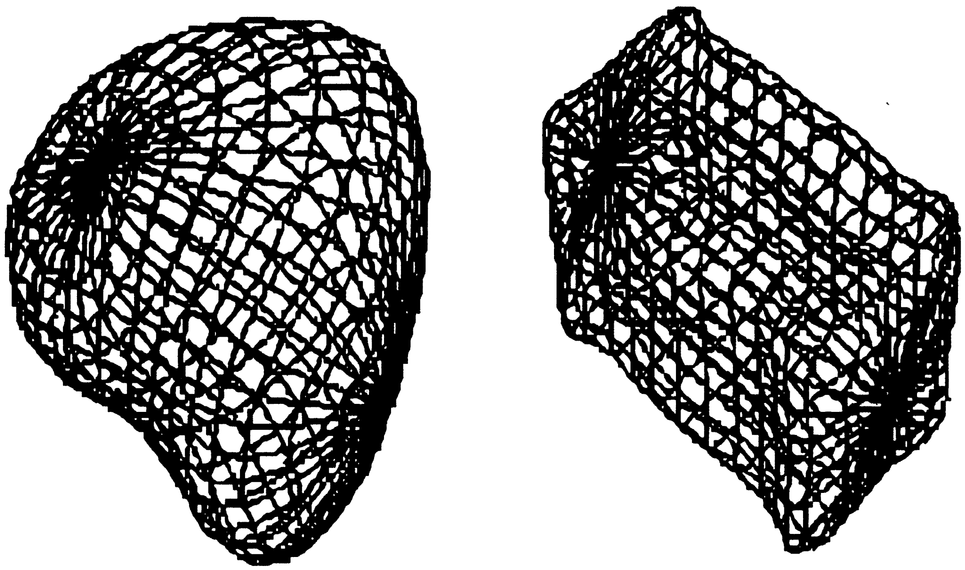
$$\underline{y}_t = (\underline{A} + \gamma I)^{-1}\left[-\gamma \underline{y}_{t-1} - \underline{f}_y(\underline{x}_{t-1}, y_{t-1})\right]$$

The matrix $\underline{A} + \gamma \underline{I}$ is a pentadiagonal, $n \times n$ matrix. It's inverse can be calculated using LU decompositions.

Note:

1. To create a closed curve, $v(n) = v(0)$.
2. For simplicity, we will probably use $n+1$ points along the curve. As the curve perimeter increases, the curve is more sparsely sampled.
3. $\underline{f}_x$ and $\underline{f}_y$ include constraint and image forces. To obtain $\underline{f}_x$ from an image energy, we must take a numerical gradient. From a spring force, we might compute it analytically.

Global Surface Models



These are example 3D global models displayed with wire frame rendering. The particular model is a Fourier surface model. These surfaces are closed. Open tube models are also possible.

The structure on the left is made with a 4th order model whereas the one on the right is made with an 8th order model.

Global Models

An arbitrary curve can be represented explicitly by two functions of one parameter, x(s), y(s). A surface can be represented as three functions in two parameters x(u,v), y(u,v), and z(u,v). Potential models are:

- Polynomials. Conics are 2nd degree curves such as ellipses, parabolas, etc. Quadrics are 2nd degree surfaces such as spheres, ellipsoids, cones, etc.
- Superquadrics. This model includes parameters that allow it to vary from an ellipsoid to a rectangular parallelpiped. They are more "expressive" than the quadric.
- Generalized cylinders. They represent elongated objects. There is a curve representing the spine of the object and a 2D cross-sectional area. For the case of a straight, homogeneous generalized cylinder, we have
  - » $x(u,v) = r(u)\, x(v) + pz(u)$
  - » $y(u,v) = r(u)\, y(v) + pz(u)$
  - » $z(u,v) = z(u)$
- Spherical harmonics. Surfaces are a radial deformation of a sphere. All surface points may be "seen" from a point inside the surface.

Global Models

- Fourier surfaces
- Open surfaces such as tubes
- Closed surfaces

A surface model can be used to:

- Calculate curvature directly from the model.
- Create 3D displays.
- Interpolate
- Compute volumes
- Compute motion of a deformable object

3D surface models require many parameters in order to fit complex surfaces. Simpler models are less "expressive."

Segmentation with Global Models

We wish to segment a 3D, gray-scale data set to a global surface model.

- Requires a match of the model to "boundary strength" in the 3D data set.
- A natural candidate is the 3D gradient strength.
- The gradient strength can be computed with 3D discrete operators such as a 3x3x3 finite difference operator.
- The gradient strength on the surface is integrated and maximized.

An objective function for a curve is

$$M(b,P)=\int_0^S | b\{x(P,s),y(P,s),z(P,s) | ds$$

An objective function for a surface is

$$M(b,P)=\int ? \int_A | b\{x(P,u,v),y(P,u,v),z(P,u,v) | dA$$

Segmentation with Global Models

- We wish to optimize the objective functions with respect to the model parameters, P.
- Optimization is done with a gradient ascent method.
- The gradient ascent method requires the gradient of the objective function with respect to each of the parameters in the model, P.
- As in the case of locally deformable models, one likes to smooth the gradient image so as to make a larger area for attraction. This helps the optimization method "find" the surface.

Potential Modifications

- Use a priori shapes as starting value for locally deformable model. This will tend to create a desired shape. This has been used in the segmentation of lungs.
- Make the material properties depend upon position in the curve ($\alpha(s)$, $\beta(s)$). This might be another way to incorporate shape memory.
- Use one surface to help anchor another. An example is to use the epicardial surface of the heart to anchor the endocardial surface. Attractive (spring) forces can be placed between the two surfaces.
- The properties of the image gradients can be used to modify the material properties ($\alpha(s)$, $\beta(s)$). That is, if the edge information is appropriate, then a corner is placed!
- An attractive possibility is to use a global model and then allow a locally deformable model to deform from it.

"Cardiac MR Imaging Using Deformable Models" Singh et al

I. Energy formulation

$$E = E_{int} + E_{ext} + E_{geo}$$

$$E_{int} = \alpha_1 \int_S |V_s(s)|^2 ds + \alpha_2 \int_S |V_{ss}(s)|^2 ds$$

$$E_{ext} = \beta_1 \underbrace{\int_S \left| I(x(s), y(s)) - I_v \right| - K \, ds}_{\text{tracks constant intensity}} + \beta_2 \underbrace{\int_S [\nabla I]^2 ds}_{\text{use Sobel}}$$

$$E_{geo} = \sum_S [r(s) - r_i] \left[\left(\frac{r_i}{r(s)} \right)^{\gamma} - 1 \right]$$

$\uparrow \gamma \longrightarrow E_{geo} \uparrow \rightarrow$ circular

For outer wall, use E_{geo} such that a constant distance from inner wall is desireable.

II Parameters

$\uparrow \alpha_2 \longrightarrow$ smooth contour

$\alpha_2 = 0$ required for corner

α_1 gives the "stretchability" of the contour

$\uparrow \alpha_1 \longrightarrow$ doesn't stretch

$\uparrow \gamma \longrightarrow \uparrow E_{geo} \longrightarrow$ circular

$\downarrow \gamma$ used for inner boundary

$$\alpha_1(x,y) = C_1 + \frac{\nabla I(x,y)}{Q_1}$$

$$\alpha_2(x,y) = C_2 + \frac{\nabla I(x,y)}{Q_2}$$

spatially varying α's

$\nabla I \uparrow \rightarrow \alpha_1, \alpha_2 \uparrow$

$\nabla I \downarrow \rightarrow \alpha_1, \alpha_2 \downarrow$

This matches the internal and external energies. If use constant α's, then when $\nabla I \uparrow$, E_{ext} dominates and the curve is very flexible.
Whe $\nabla I \downarrow$, the curve is very rigid.

III Energy minimization

$$E = \alpha_1 \sum_s |v(s) - v(s-1)|^2 + \alpha_2 \sum_s |v(s+1) - 2v(s) + v(s-1)|^2$$
$$+ \beta_1 \sum_s \left[|I(x(s), y(s)) - I_v| - K \right]$$
$$+ \beta_2 \sum_s \left\{ [S_x * I(x(s), y(s))]^2 + [S_y * I(x(s), y(s))]^2 \right\}$$
$$+ \sum_s (r(s) - r_i)\left[\left(\frac{r_i}{r(s)} \right)^{\gamma} - 1 \right]$$

Algorithm

1 iteration
- E evaluated over 5 pixels in contour
- central pixel is moved to each of 8 neighbors and E is evaluated for each of 9 possible positions.
- the central pixel is left at the position giving the min E
- this process is repeated for each pixel on the contour

- Stoping criterion is that the contour doesn't change

IV. Questions

1. Does this give an optimum for $E_{contour}$?

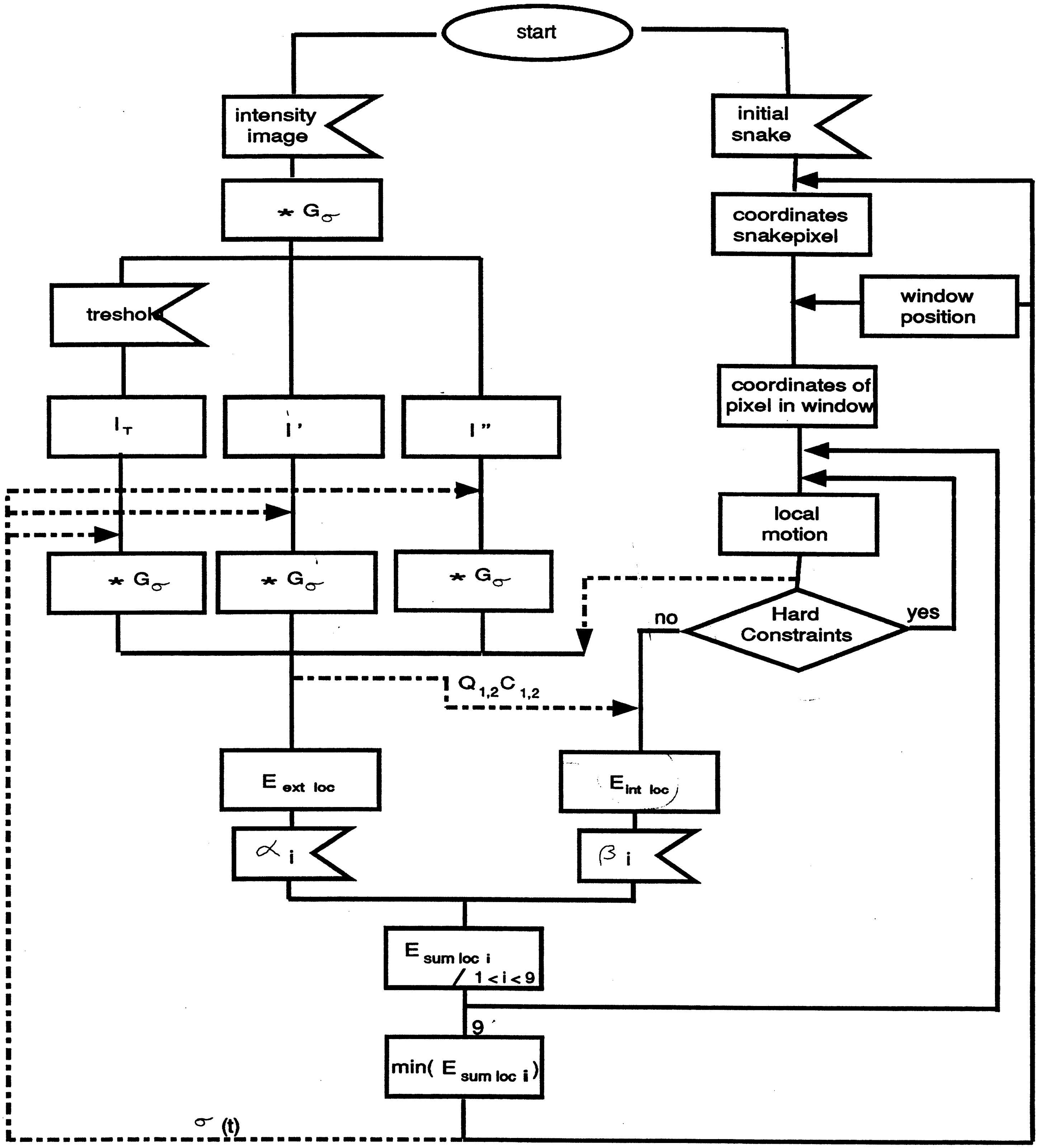
2. Does this give a global optimum?

3. Why does E_{int_1} relate to the length (stretchiness) of the contour?

4. Is the sign of E_{geo} correct?

5. Blurring the gradient image with a 3×3 Gaussian results in better convergence. Explain.

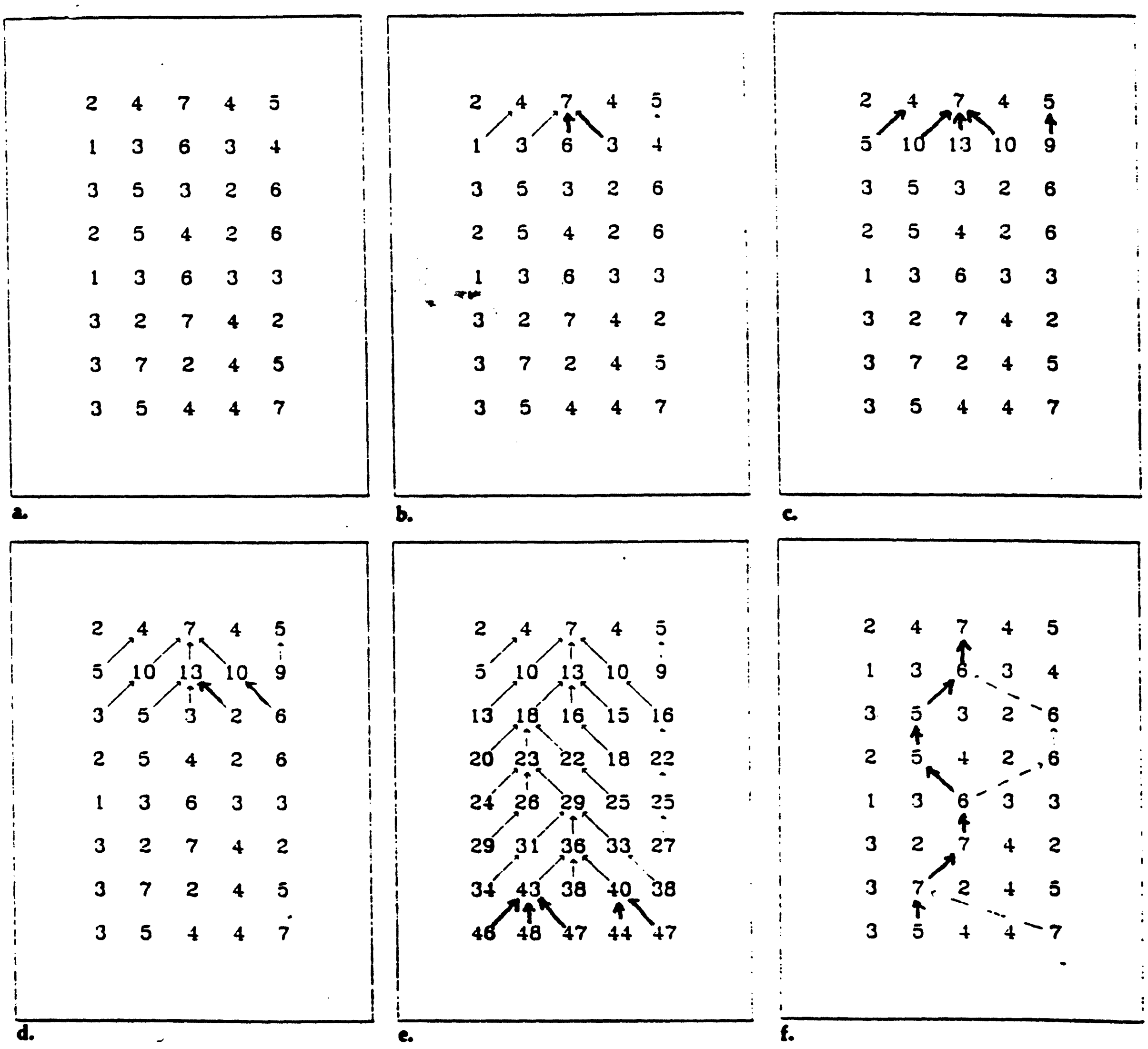
6. Compare this method to DP.


start
intensity image
initial snake
* G_σ
coordinates snakepixel
treshold
window position
I_T
I'
I''
coordinates of pixel in window
local motion
* G_σ
* G_σ
* G_σ
no
Hard Constraints
yes
$Q_{1,2} C_{1,2}$
$E_{ext\ loc}$
$E_{int\ loc}$
α_i
β_i
$E_{sum\ loc\ i}$
$1 < i < 9$
9
$\min(E_{sum\ loc\ i})$
$\sigma(t)$

B) Dynamic programing (Ref: Pope et al)

1) Generate a liklihood matrix. Often this is from a matched edge filter.

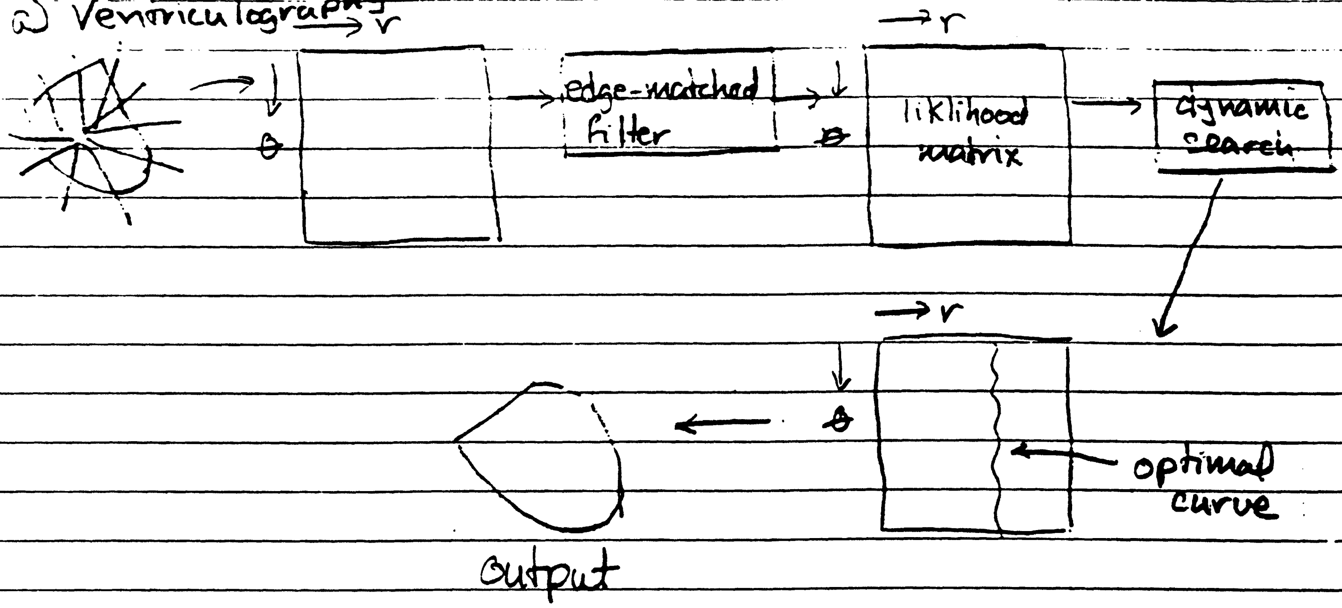
2.) Task is to find the best "path" through the liklihood matrix. Examine the method by example.



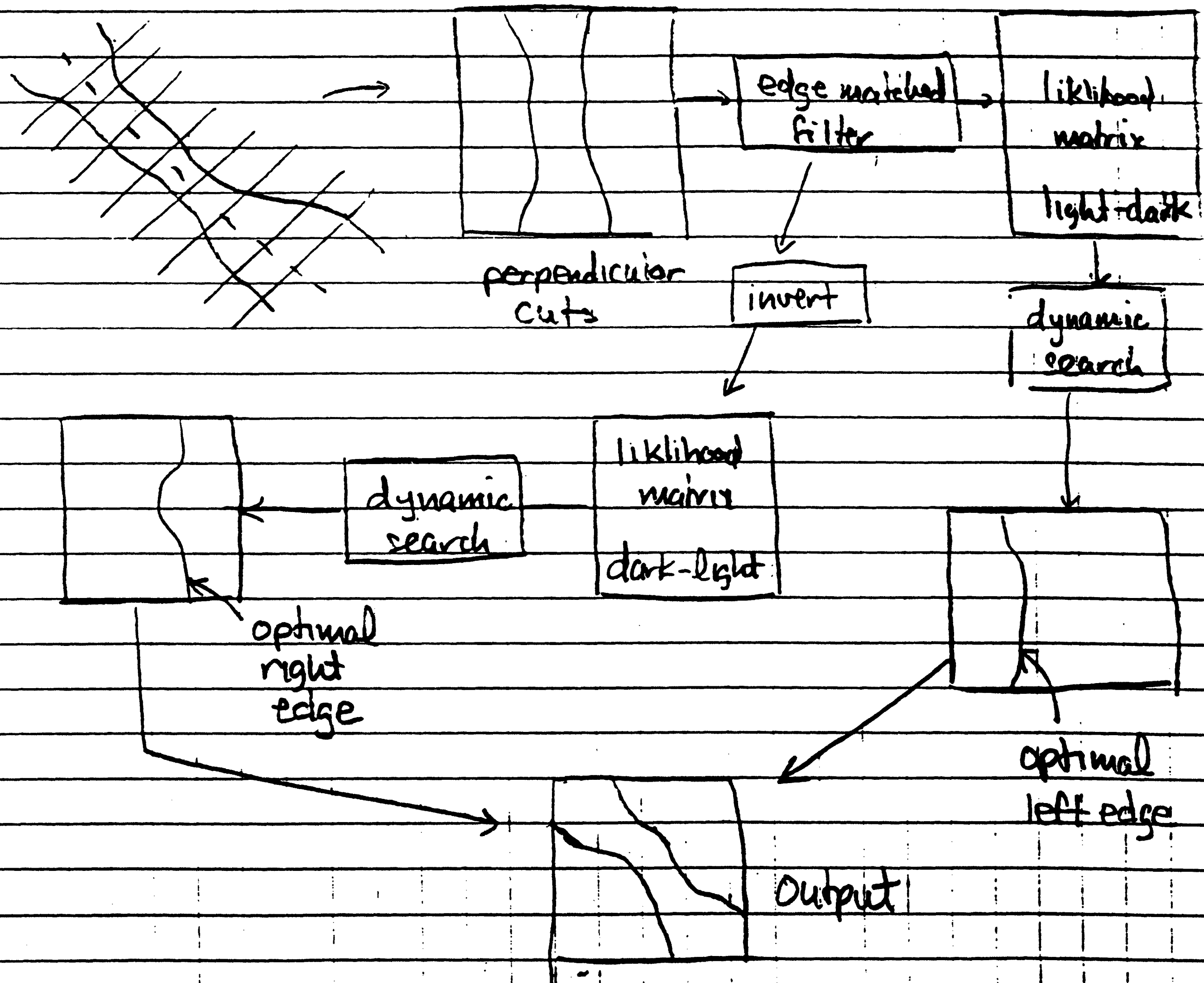
An example of the dynamic search algorithm applied to a hypothetical likelihood matrix.

3) Applications

a) Ventriculography



b) artery tracking



4) Observations

a) Dynamic programming tracks "global" edges. Edges are guaranteed to be continuous in an 8-neighborhood sense.

b) The cost function is now simply the cumulative edge strength. Modifications to the cost function could include a smoothness criterion, etc.

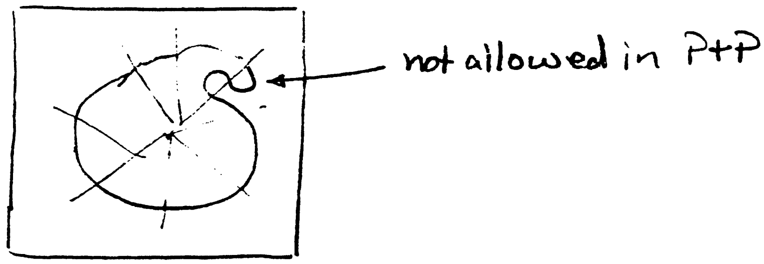
c) In the artery edge-tracking example, one can insure that the left and right edges don't cross. How?

d) One way to add a manual override is to allow one to add high-valued points in the liklihood matrix. These become "attractors".

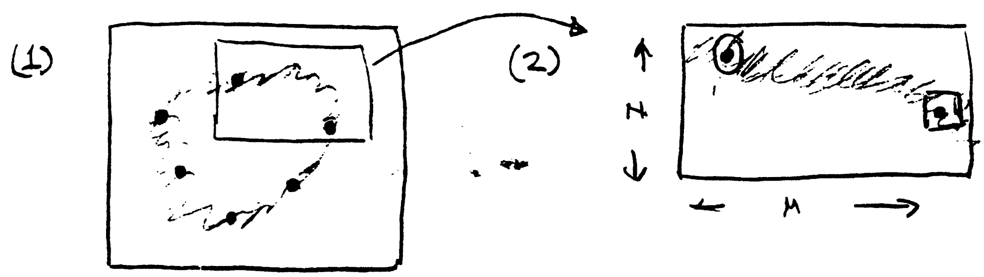
e) Can one track both artery edges simultaneously?

2D Dynamic Programming

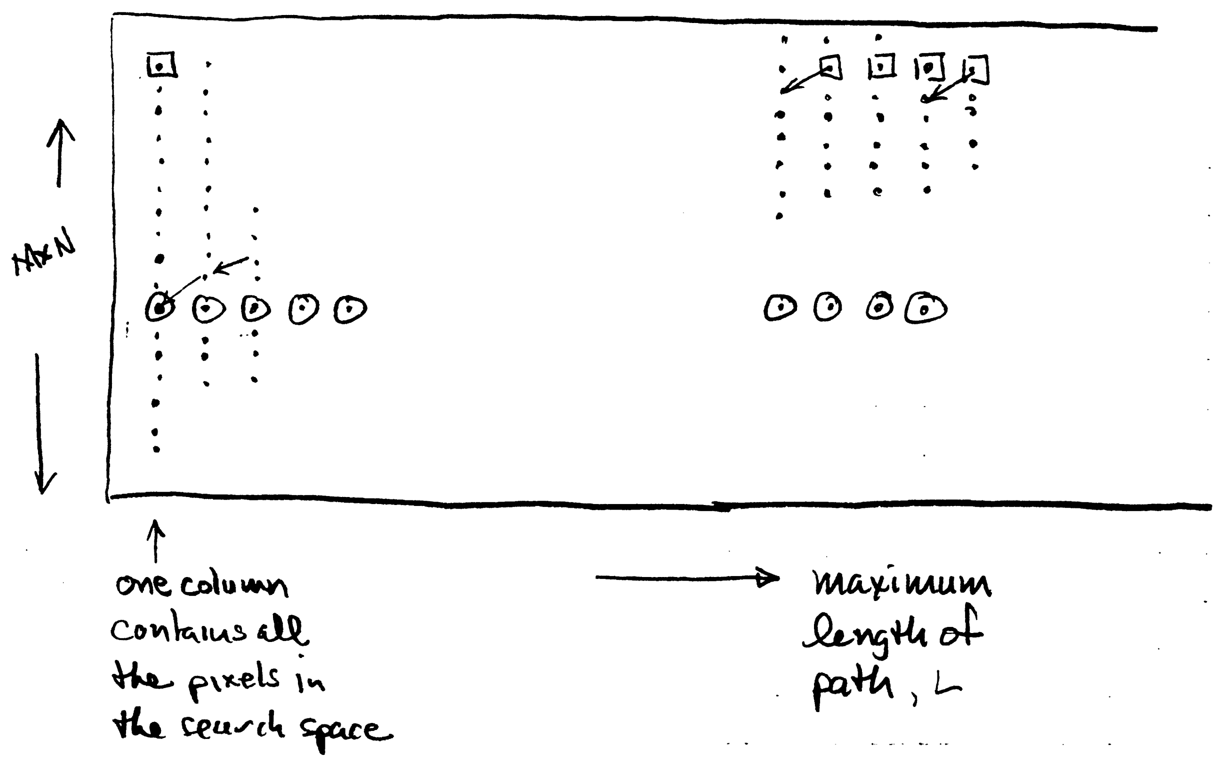
A downside to the Pope + Parker algorithm is that it requires an almost circular boundary.



Geiger and Vlontzos (submitted for publication) use a true 2D approach.



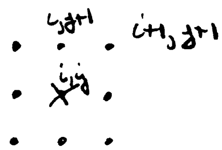
(3) Create a DP array



The DP algorithm runs from left to right above. Starting at ⊙ we will finally get a path to ⊡. But, this may not be the best path. The DP algorithm runs for a length L. One then evaluates all the possible paths to L and picks the one with least cost.

At each step, a cost function is evaluated.

Hard constraints are imposed corresponding to 8-neighbor connectivity. As in P+P, only neighbors are checked.



Recently Mortensen, Morse, + Udupa (Comp. Cardiol, 1992) created a "live-wire" approach to 2D DP. They effectively compute the 2D DP matrix above and leave the forward cost and the backtracking information stored. Then, if the user changes ⊡, a new path can be tracked back! This can happen at near interactive speeds. Hence, the name live-wire.

QLW

"Dynamic Programming for Detecting, Tracking, and Matching Deformable Contours" Geiger et al PAMI-17: 294-302.

A. Optimization

- hard constraints, 8 neighbors

$$\text{cost} = \frac{1}{l_{ij}^2 + \rho} + \eta\left[\phi_{ij} I - \phi_{i_p, j_p} I\right]$$

$l_{i,j}$ = normalized gradient magnitude $= \frac{|\nabla I|}{\max}$

$\phi_{i,j} I$ = angle of gradient $= \arctan\left(\frac{\nabla_y I}{\nabla_x I}\right)$

ρ = constant to eliminate huge penalty for zero gradient

η = weighting constant

B. Interactive
- user selects points
- algorithm determines lines ~~& search space~~
- curve ends on lines

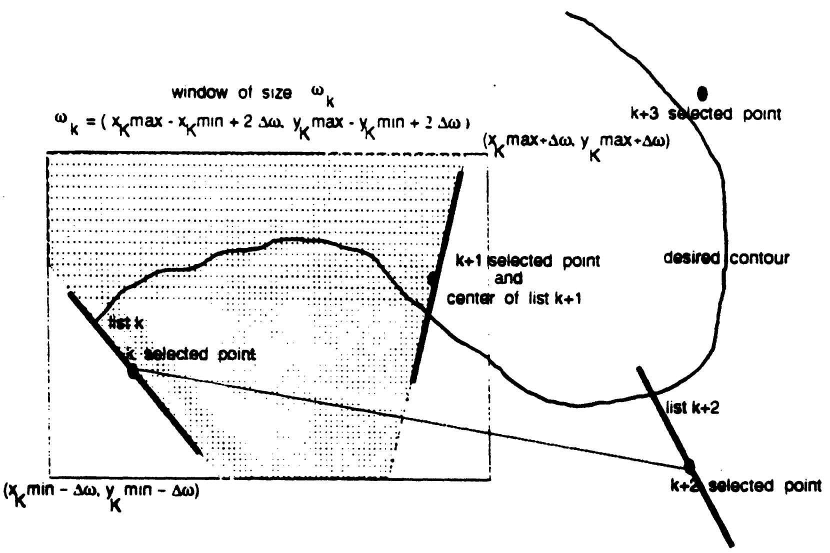


Fig. 1. Four selected points, from k to $k+3$, three lists of uncertainty, from $list_k$ to $list_{k+2}$, and a window for DP search, $\omega = (\omega_k^x, \omega_k^y)$. The desired contour is also shown. The search space is denoted by the shaded region. Note that $list_{k+1}$ is perpendicular to the straight line connecting k to $k+2$ and analogously for $list_{k+2}$.

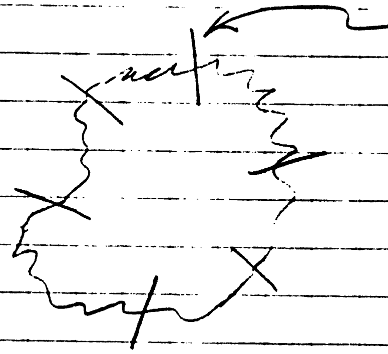
C. Open contours

- 2D DP from <u>all</u> points on list k to list k+1
- At list k+1, get the costs corresponding to various paths back through matrix. At each point keep the path to the best starting point on the list k.
- Keep costs and paths
- Use costs on list k+1 as starting values for 2D DP to k+2.
- At the end list, find the min total cost and back track the entire curve.

D. Closed contours

1. Use the approach above. The only constraint is that the starting point must equal the last point. This constraint is imposed by making them very high values.

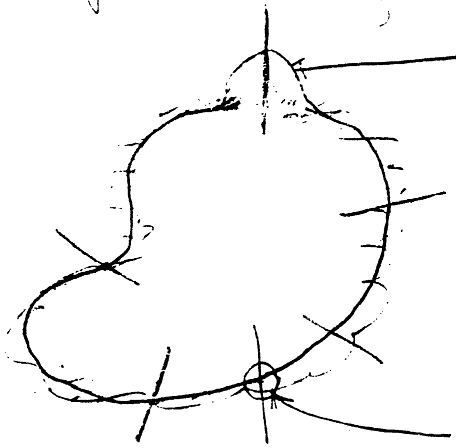
   One approach is to run the entire DP process (2M+1) times. Each time pick the start (and end) pixel.



2M+1 pixels on start and end line.

~~Run 2D DP (2M+1) times.~~

2. 2 pass technique



On first pass, start and end at center of line. This is a fixed endpoint, 2D DP.

On second pass, use the pixel from the first pass as the start and stop pixel.

This gets a result in 2 passes rather than (2M+1) passes!

E. Curve matching

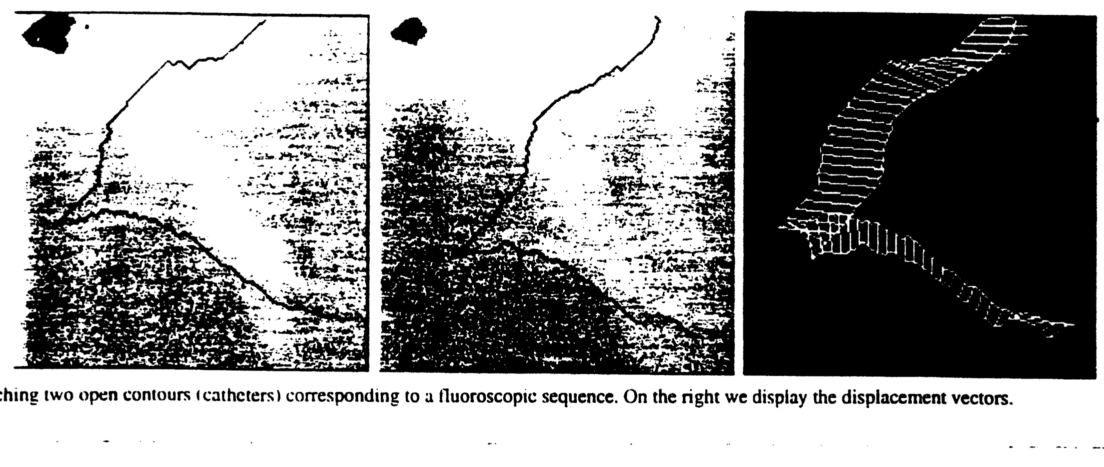


Fig. 8. Matching two open contours (catheters) corresponding to a fluoroscopic sequence. On the right we display the displacement vectors.