قسم الهندسة الحيوية الطبية والمنظومات

جامعة القاهرة
كلية الهندسة

Computer Vision
404 B
Tutorial 3
25/03/20

Eng. Eman Marzban

It aims at fitting a curve of arbitrary shape to a set of object edge points

$$\sum_{i=1}^{N} (\alpha_i E_{cont} + \beta_i E_{curv} + \gamma_i E_{image})$$

Internal Energy

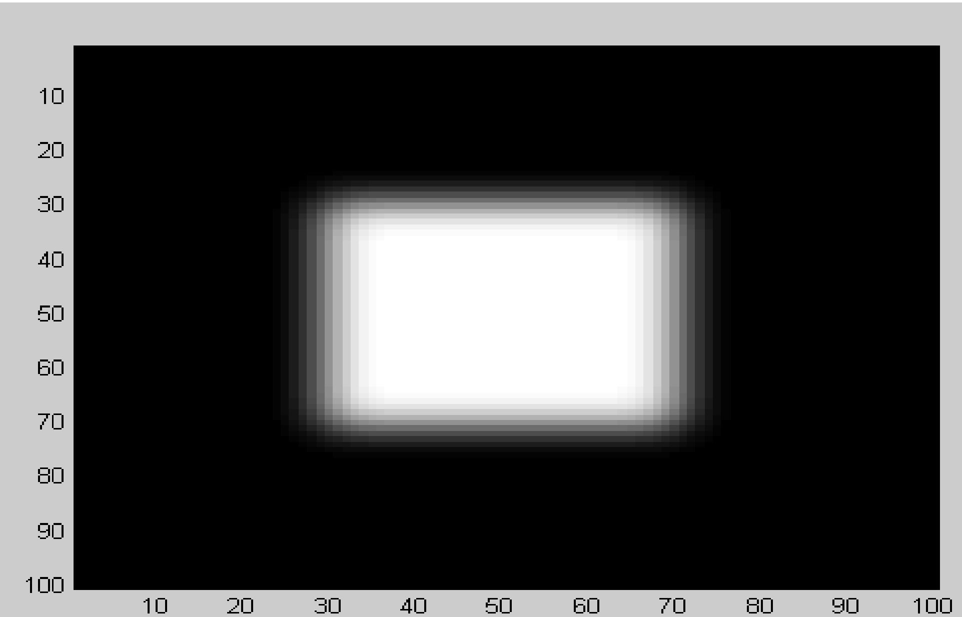
Continuity Energy Term

$$E_{cont} = \|\mathbf{p}_i - \mathbf{p}_{i-1}\|^2 \Rightarrow E_{cont} = \left(\bar{d} - \|\mathbf{p}_i - \mathbf{p}_{i-1}\|\right)^2$$

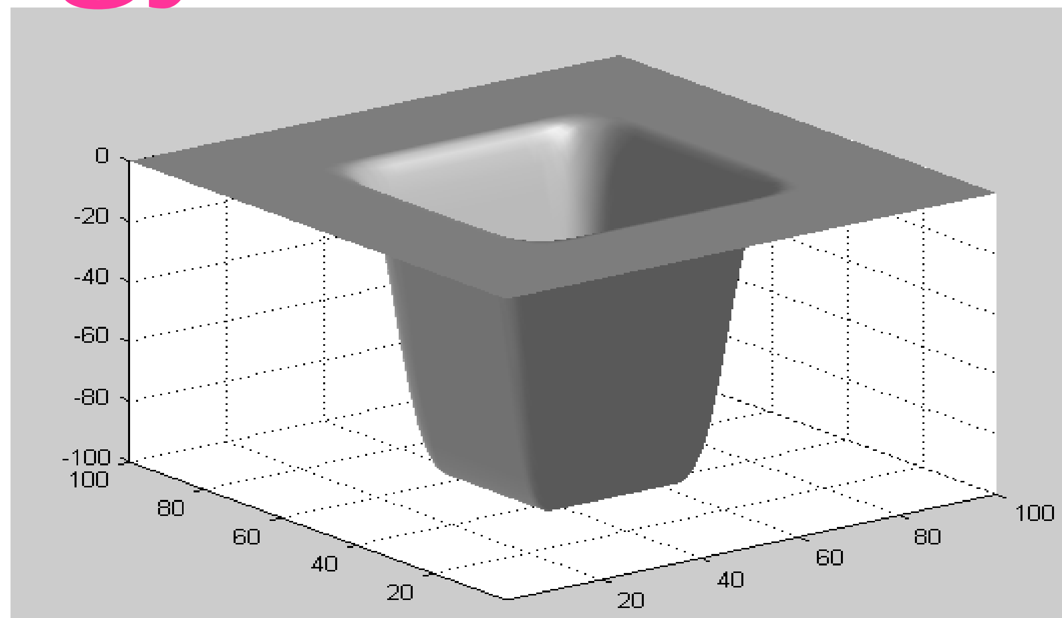
Curvature Energy Term

$$E_{curv} = \|\mathbf{p}_{i-1} - 2\mathbf{p}_i + \mathbf{p}_{i+1}\|^2$$

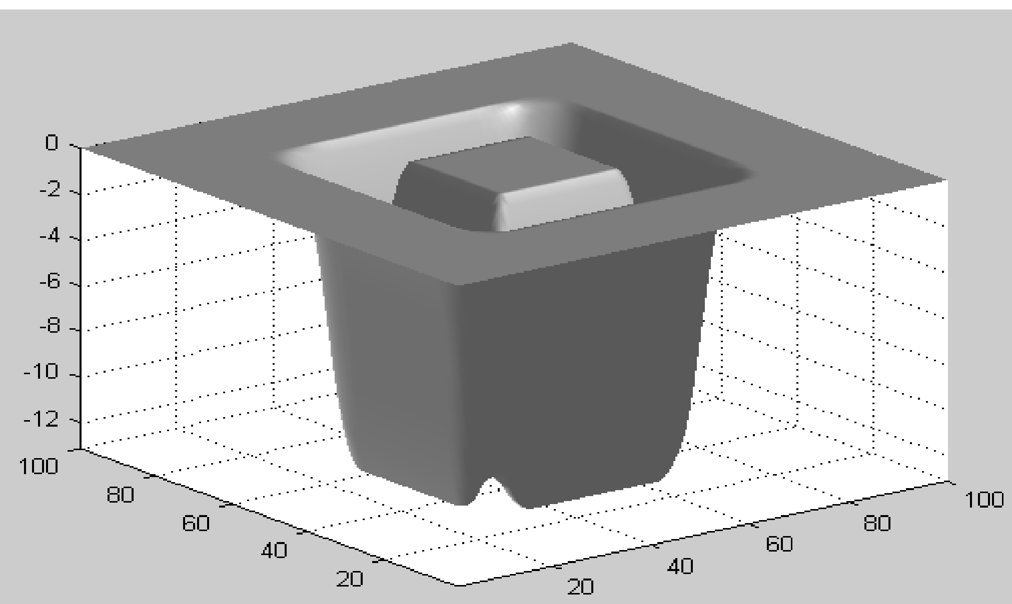
External Energy: Image Energy



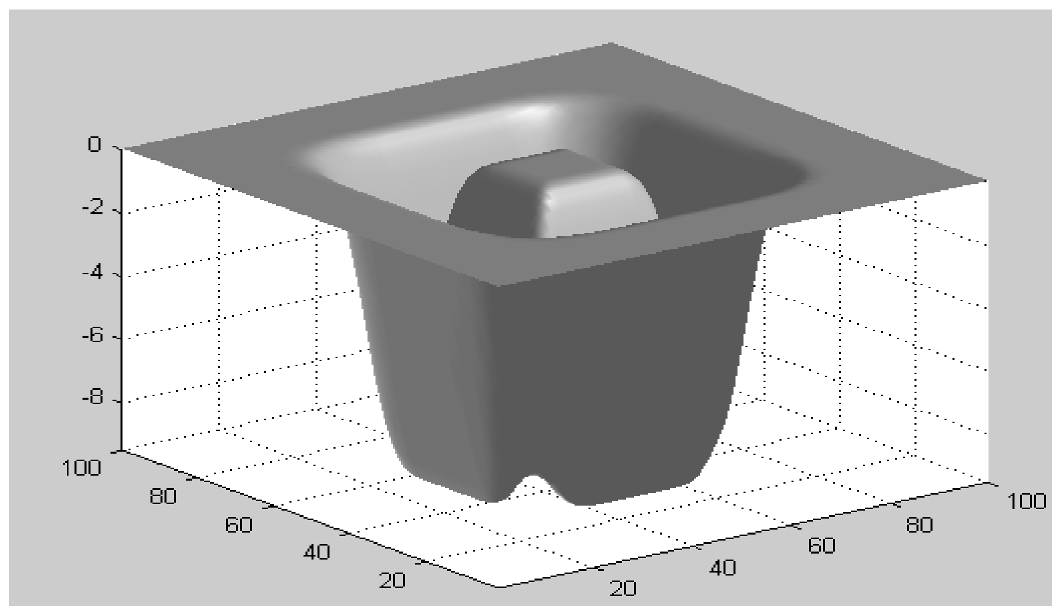
I



-I



$-|\nabla I(x,y)|$



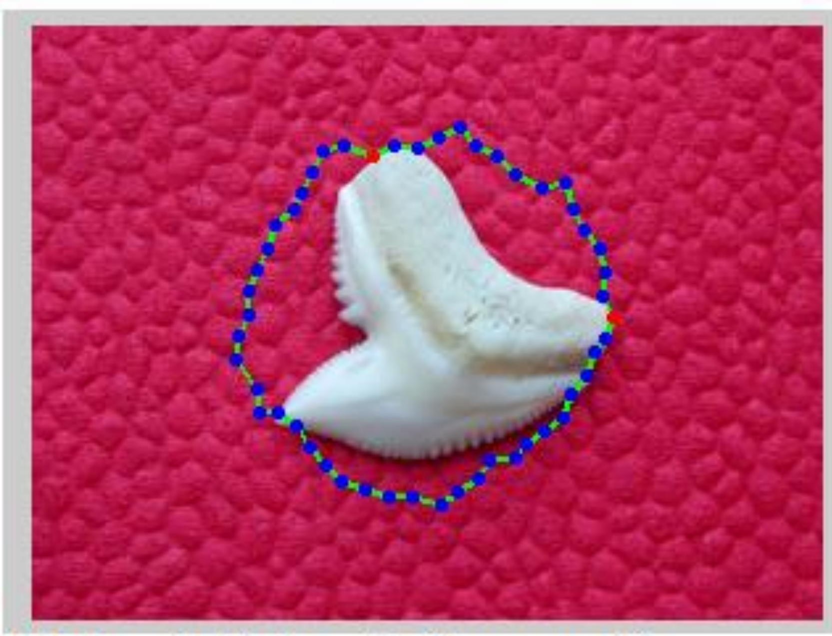
$-||\nabla[G_\sigma(x,y) * \mathbf{I}(x,y)]||$

Weights: α, β, and γ

E	α, β, γ	Etot	Min P
1, 1, -1	1, 1, 1	1	
1, 1, -2	1, 1, 1	0	✔
1, 1, -1	1, 2, 1	2	
1, 1, -2	1, 2, 1	1	✔
1, 1, -1	1, 2, 1	2	✔
2, 1, -2	1, 2, 1	4	

alpha=1 , Beta =1, Gamma =0.5

alpha=1, Beta=1, Gamma =1



Alpha=1, Beta =1, Gamma=2

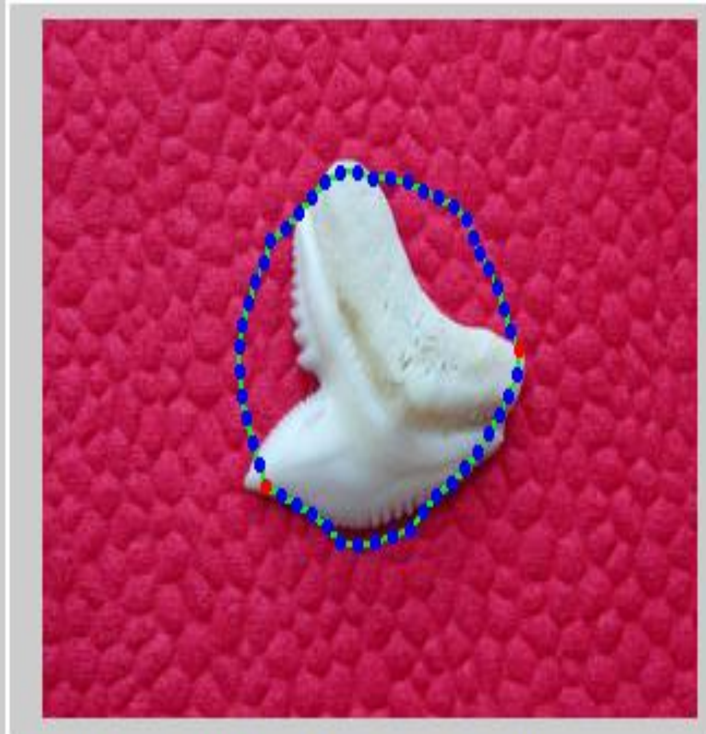
Alpha=1, Beta=1, Gamma =5

Alpha=0.5, Beta=1, Gamma=1

Alpha=2, Beta=1, Gamma=1

Alpha=4, Beta=1, Gamma=1

Alpha=1, Beta=.5, gamma=1



Alpha=1, Beta=2, gamma=1

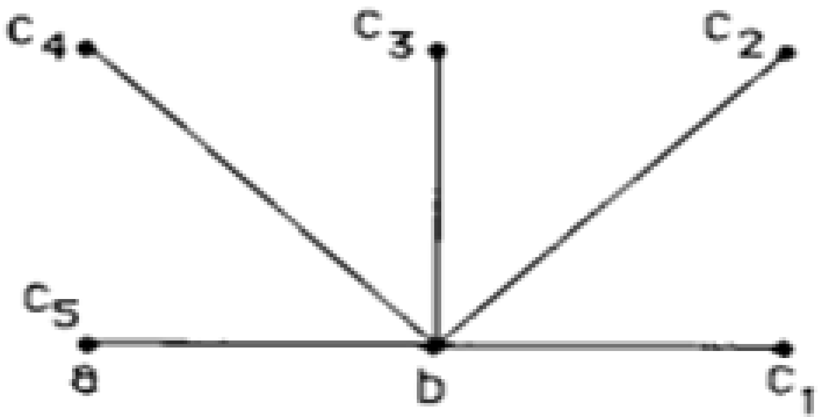
Alpha=1, Beta=4, Gamma=1

alpha = 5; beta = 2; gamma = 1.2

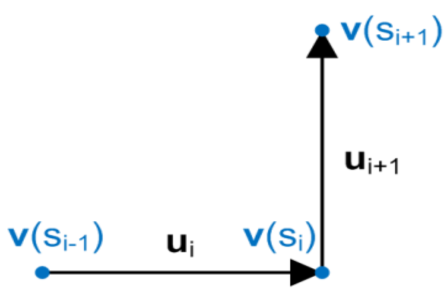
alpha = 5; beta = 1.5; gamma = 1.2

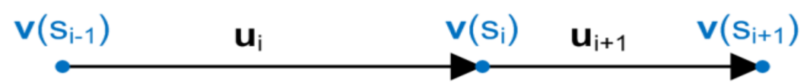
alpha = 5; beta = 1.5; gamma = 1

Curvature: a-b-c



c	$\|\vec{\mathbf{u}}_i - \vec{\mathbf{u}}_{i+1}\|^2$
1	0.0
2	1.0
3	2.0
4	5.0



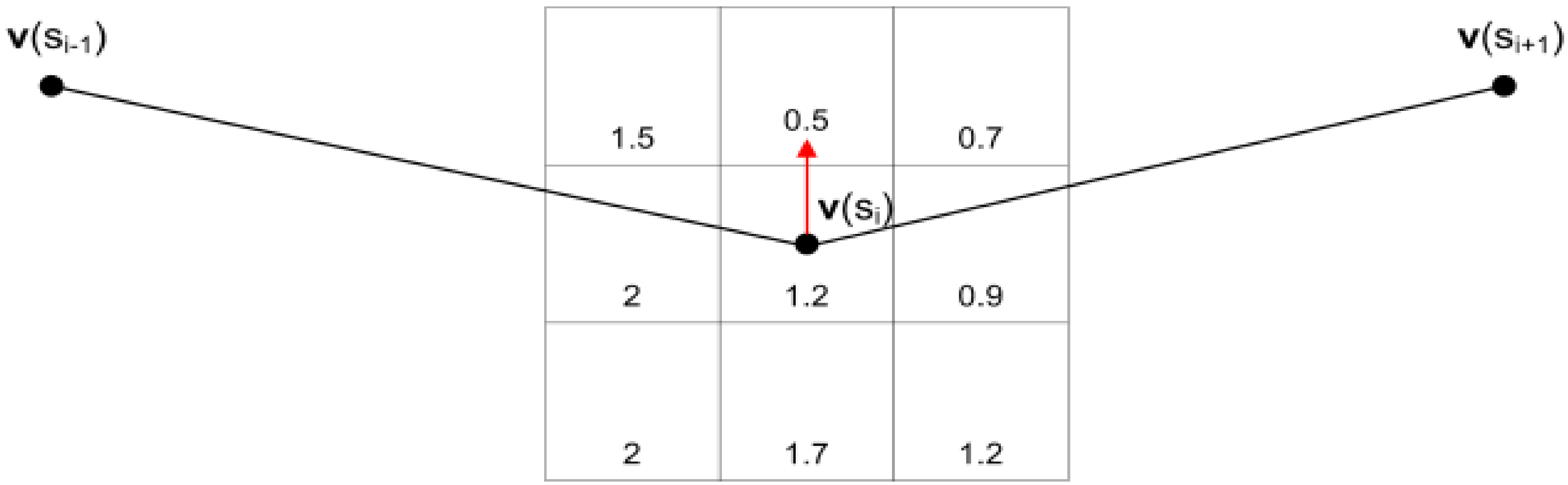


$$\left\| \frac{u_i}{\|u_i\|} - \frac{u_i+1}{\|u_i+1\|} \right\|^2$$

where $ui = [x(s_i) - x(s_{i-1}), y(s_i) - y(s_{i-1})]$ *and* $ui + 1 = [x(s_{i+1}) - x(s_i), y(s_{i+1}) - y(s_i)]$

Curvature: The two measures explained

Vi-1	Vi	Vi+1	Ecurv	Ui	Ui+1	C	Cos(θ)
1,1	3,1	3, 3	8	2,0	0, 2	2	0
1,1	6,1	6, 3	29	5,0	0, 2	2	0
1,1	3,1	5, 1	0	2, 0	2, 0	0	-1
1,1	6,1	7, 1	4	5, 0	1, 0	0	-1
1,1	7,1	3, 5	116	6, 0	-4, 4	3.38	0.71
3,1	7,1	3, 5	80	4, 0	-4, 4	3.38	0.71
1,1	7,1	2, 2	125	6, 0	-5, 1	3.96	0.98
3,1	7,1	2, 2	85	4, 0	-5, 1	3.96	0.98



Values shown in the cells correspond to E_{total}

Algorithm 1 A GREEDY SNAKE

Input: : determine ROI, parameters α, β and γ

Output: Stop if only few points have moved

```
% n is the total number of snake control points
Index arithmetic for the snake control points is modulo n
Initialize the parameters  α, β and  γ
Do    % Main loop that moves the snake points to new locations
  for i = 1 to n  % The first and last point are the same in snake
                    Emin = infinity
         for j = 1 to m     % m is the neighborhood size
                    E(j) =  α Eela(j) + β Ecurv(j) + γ Eimg(j)
           if E(j) < Emin then     % Find location with min energy
                    Emin = E(j)
                    jmin = j
                    Move point v(i) to location jmin
             if jmin is not the current location then ptsmoved ++
   % The process below determines where to relax ?
        for i = 1 to n % Calculate exact curvature
           c(i) = || u(i) / || u(i) || - u(i+1) / || u(i+1)|| ||^2
        for i = 1 to n % Find where to relax ?
            if (c(i) > c(i-1) and c(i) > c(i+1)
               and c(i) > TH
               and mag(v(i)) > TH-mag
        then β(i) = 0      % Relax ? if all conditions true
           while ptsmoved > TH-moved
   % Stop if only few points have moved
```

Figure 3.4:Pseudo-code for the greedy snake algorithm[39].

Snakes: Beta effect

The input is formed by an intensity image, I, which contains a closed contour of interest, and by a chain of image locations, $\mathbf{p}_1, \ldots, \mathbf{p}_N$, defining the initial position and shape of the snake.

Let f be the minimum fraction of snake points that must move in each iteration before convergence, and $U(\mathbf{p})$ a small neighborhood of point $\mathbf{p}$. In the beginning, $\mathbf{p}_i = \bar{\mathbf{p}}_i$ and $d = \bar{d}$ (used in E_{cont}).

While a fraction greater than f of the snake points move in an iteration:

1. for each $i = 1, \ldots, N$, find the location of $U(\mathbf{p}_i)$ for which the functional $\mathcal{E}$ is minimum, and move the snake point $\mathbf{p}_i$ to that location;
2. for each $i = 1, \ldots, N$, estimate the curvature k of the snake at $\mathbf{p}_i$ as

$$k = |\mathbf{p}_{i-1} - 2\mathbf{p}_i + \mathbf{p}_{i+1}|,$$

   and look for local maxima. Set $\beta_j = 0$ for all $\mathbf{p}_j$ at which the curvature has a local maximum and exceeds a user-defined minimum value;
3. update the value of the average distance, $\bar{d}$.

On output this algorithm returns a chain of points $\mathbf{p}_i$ that represent a deformable contour.

- It is important to normalize the contribution of each term for correct implementation:

1) For E_{cont} and E_{curv}, it is sufficient to divide by the largest value in the neighborhood in which the point can move.
2) Normalize E_{image} using this formula

$$new\_v = \frac{v - min_X}{max_X - min_X} \cdot (new\_max_X - new\_min_X) + new\_min_X$$

Which is reduced to

$$new\_v = \frac{v - min_X}{max_X - min_X}$$

When the array is normalized between 0 and 1

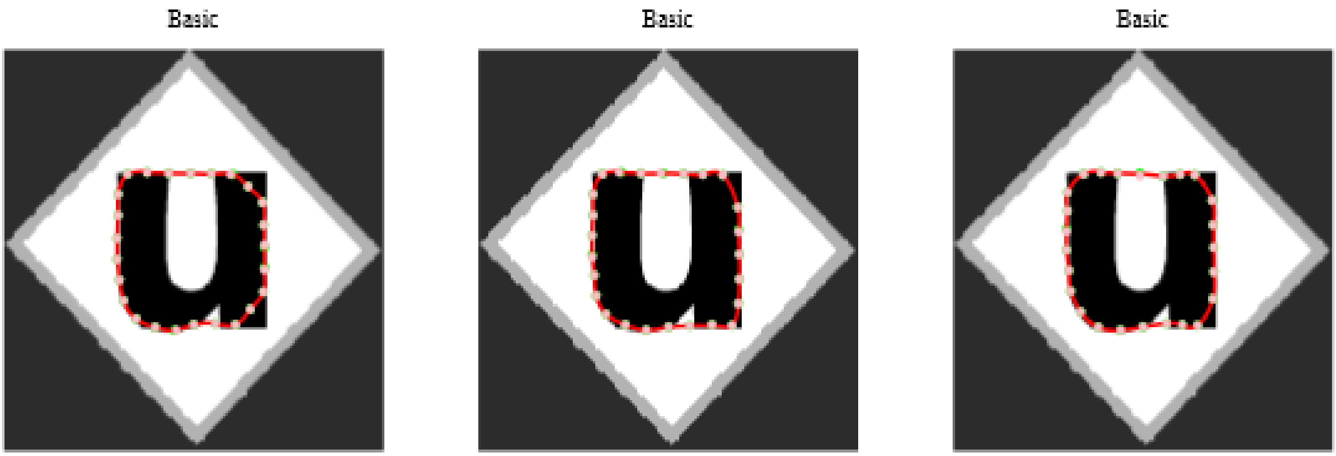


Figure 3.6: Result of the basic snake on a test image with $\alpha = 1$ (left), $\alpha = 0.5$ (middle) and $\alpha = 0$ (right).

As can be seen in figure 3.6, by decreasing α the distance between the contour points is less and less equal. When α is equal to zero the contour points leave big gaps between each other.

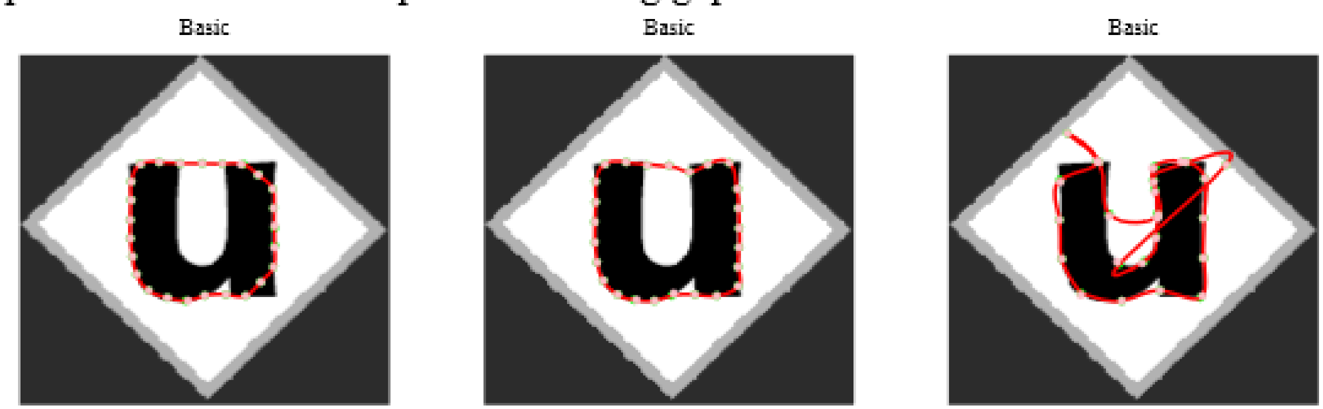


Figure 3.7: Result of the basic snake on a test image with $\beta = 1$ (left), $\beta = 0.5$ (middle) and $\beta = 0$ (right).

As can be seen in figure 3.7, by decreasing β the contour becomes less rigid and sharper angles can be formed between contour points. This can give dramatic results as in the right image. When β is too big though as in the left image, the contour points can't handle corners well. This can especially be seen by comparing the left image with the middle image, where the corners are taken much better.

References

1) P. Tiilikainen, "A Comparative Study of Active Contour Snakes". Copenhagen University, Denmark, 2007
https://www.scribd.com/document/324882550/nikolas-070901
3) https://www.cse.unr.edu/~lzhang/snake/snake.doc
4)https://pdfs.semanticscholar.org/562b/786ef6921c4770582905cc6ee1c3fa0fcee4.pdf?_ga=2.263374796.1921450585.1585126563-1685233747.1572623401