EECS332 Digital Image Analysis

Contours and Curve Fitting

Ying Wu

Electrical Engineering & Computer Science
Northwestern University
Evanston, IL 60208

http://www.ece.northwestern.edu/~yingwu
yingwu@ece.northwestern.edu

Motivations

- Low-level → higher-level?
  – we've got edges, but …
  – how can we represent an object?
  – can we model its shape?
- Rough → more accurate?
  – we've got segments, but …
  – they are rough
  – can we find more accurate information, such as the location of a fingertip?

2

Outline

- Motivation
- Curve representations
- Ployline fitting
- Arc fitting
- A better solution: LS fitting
  - The principle
  - Example 1: line fitting
  - Example 2: conic fitting

3

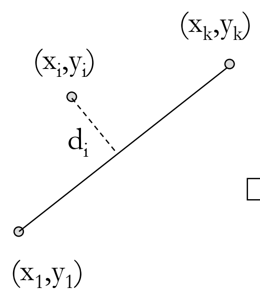
Curve Representations

- Non-parametric
  - A set of edge points
- Parametric
  - Line
  - Conic
  - Polyline
  - Spline

4

Simple Line Fitting

- Use only two end points



$$\frac{y-y_1}{x-x_1}=\frac{y_k-y_1}{x_k-x_1}$$

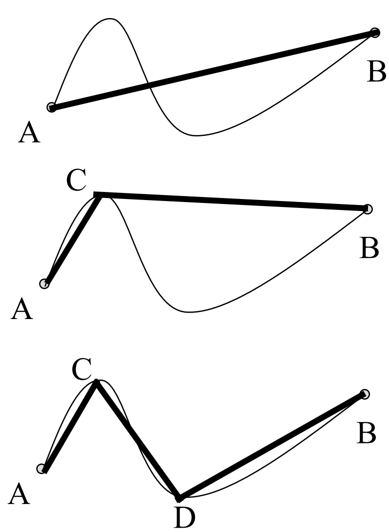
$$\Rightarrow \quad x(y_1-y_k)+y(x_k-x_1)+y_kx_1-y_1x_k=0$$

$$d_i=\frac{x_i(y_1-y_k)+y_i(x_k-x_1)+y_kx_1-y_1x_k}{\sqrt{(x_k-x_1)^2+(y_k-y_1)^2}}$$

5

Ployline Splitting

- Note: needs an ordered edge list

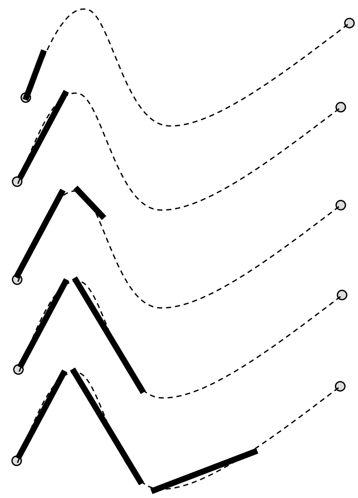


1. find the two end points A, B of the curve;
2. fit a straight line connecting AB;
3. find the furthest point C to the line AB;
4. if the error is above a threshold, insert a vertex at C;
5. else, do recursion.

6

Ployline Merging

- Note: needs an ordered edge list

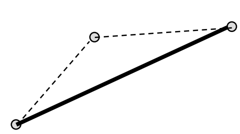


1. traverse the edge list:
2. if the nearby edge point is "good", (i.e., small error),
    1) then include it into the current line segment,
    2) and update the parameters of the line;
3. else, start a new line.

7

Compare with region seg.

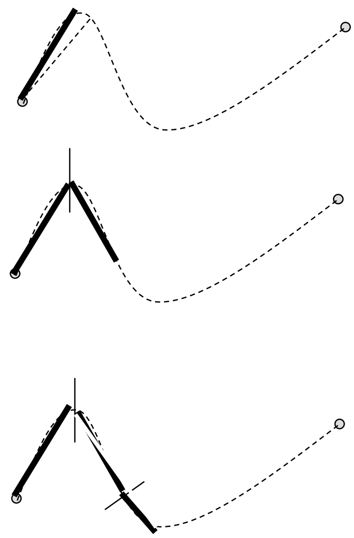
- The above merging scheme may end up with too many line segments
- → over-segmentation
- Why don't we merge two line segments?



Check two adjacent two line segments to see if we can merge them.

8

Hop-Along Algorithm



1. Start with the first k edge points
2. Fit a line between the first and the last edge points in the sublist;
3. If the error is large, shorten the sublist to the point of max error;
4. Compare the orientation of the current and previous line segments. If good, replace the two with a new line;
5. Check next k edge points

9

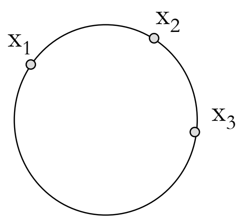
Is it good?

- Shortcomings:
  - Only use two end points
    - ✓ not accurate
    - ✓ not robust
    - ✓ may end up with too many line segments
- Solutions?
  - Why don't use more edge points in fitting?
  - We'll see this shortly.

10

Simple Arc Fitting

- Three points determine a circle



$$(x-x_o)^2+(y-y_o)^2=r^2$$

$$let \begin{cases} x'=x-x_1 \\ y'=y-y_1 \end{cases} \quad \text{we have } (x'-x_o')^2+(y'-y_o')^2=r^2$$

$$\text{then} \begin{cases} x_o'^2+y_o'^2-r^2=0 \\ x_2'^2-2x_2'x_o'+x_o'^2+y_2'^2-2y_2'y_o'+y_o'^2-r^2=0 \\ x_3'^2-2x_3'x_o'+x_o'^2+y_3'^2-2y_3'y_o'+y_o'^2-r^2=0 \end{cases}$$

$$\Rightarrow \quad (x_o', y_o', r^2)$$

11

Simple Conic Fitting

- Conics:
  - Hyperbolas
  - Parabolas
  - Ellipses
- We can use three points
  - The explicit solution is a bit complicated
  - And many other problems

12

Are these methods good?

- These explicit solutions use only the minimum number of edge points
- They are not robust to noise

LS fitting: the basic idea

- The principle of the Least Squares Fitting
  - There are a set of samples $\{(x_k,y_k), k=1,\ldots,N\}$
  - To find a mapping $y=f(x)$, such that

$$\min \; E = \frac{1}{N}\sum_{k=1}^{N}(y_k - f(x_k))^2$$

Example 1: line fitting

$$(a^*, b^*) = \arg\min_{a,b} D(a,b) = \arg\min_{a,b} \sum_{i=1}^{N} (ax_i + b - y_i)^2$$

Let's solve it! To minimize $D(a,b)$, we need to:

$$\frac{\partial D(a,b)}{\partial a} = \sum_i (ax_i + b - y_i)x_i = a\sum_i x_i^2 + b\sum_i x_i - \sum_i x_i y_i = 0$$

$$\frac{\partial D(a,b)}{\partial b} = \sum_i (ax_i + b - y_i) = a\sum_i x_i + bN - \sum_i y_i = 0$$

Then we can write them into a matrix form:

$$\begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & N \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}$$

Then, we have

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & N \end{bmatrix}^{-1} \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}$$

$$= \frac{\begin{bmatrix} N & -\sum x_i \\ -\sum x_i & \sum x_i^2 \end{bmatrix} \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}}{N\sum x_i^2 - (\sum x_i)^2} = \frac{\begin{bmatrix} N\sum x_i y_i - \sum x_i \sum y_i \\ -\sum x_i \sum x_i y_i + \sum x_i^2 \sum y_i \end{bmatrix}}{N\sum x_i^2 - (\sum x_i)^2}$$

Pseudo-inverse

$$\begin{cases} ax_1 + b = y_1 \\ ax_2 + b = y_2 \\ \vdots \\ ax_N + b = y_N \end{cases}$$

And we can write it in a matrix form:

$$\begin{bmatrix} x_1 & 1 \\ x_2 & 2 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

And we let

$$\mathbf{A} = \begin{bmatrix} x_1 & 1 \\ x_2 & 2 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix}; \qquad \mathbf{x} = \begin{bmatrix} a \\ b \end{bmatrix}; \qquad \mathbf{t} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

Then we have

$$\mathbf{Ax} = \mathbf{t}$$

$$\mathbf{x} = \mathbf{A}^\dagger \mathbf{t}$$

$$\mathbf{A}^\dagger = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$$

16

Why is P-inverse cute?

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} x_1 & 1 \\ x_2 & 2 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix}^{\dagger} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

You can easily verify that:

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum x_i^2 & \sum x_i \\ \sum x_i & N \end{bmatrix}^{-1} \begin{bmatrix} \sum x_i y_i \\ \sum y_i \end{bmatrix}$$

17

Example 2: plane fitting

$$\begin{cases} a_1 u_1 + a_2 v_1 + a_3 = I(u_1, v_1) \\ a_1 u_2 + a_2 v_2 + a_3 = I(u_2, v_2) \\ \vdots \\ a_1 u_N + a_2 v_N + a_3 = I(u_N, v_N) \end{cases}$$

$$\begin{bmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots \\ u_N & v_N & 1 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} I(u_1, v_1) \\ I(u_2, v_2) \\ \vdots \\ I(u_N, v_N) \end{bmatrix}$$

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} u_1 & v_1 & 1 \\ u_2 & v_2 & 1 \\ \vdots & \vdots & \vdots \\ u_N & v_N & 1 \end{bmatrix}^{\dagger} \begin{bmatrix} I(u_1, v_1) \\ I(u_2, v_2) \\ \vdots \\ I(u_N, v_N) \end{bmatrix}$$

18

Example 3: conic fitting

$$ax^2 + 2bxy + cy^2 + 2dx + 2ey + f = 0$$

You can easily figure it out!

$$\begin{bmatrix} x_1^2 & 2x_1y_1 & y_1^2 & 2x_1 & 2y_1 \\ x_2^2 & 2x_2y_2 & y_2^2 & 2x_2 & 2y_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_N^2 & 2x_Ny_N & y_N^2 & 2x_N & 2y_N \end{bmatrix} \begin{bmatrix} a' \\ b' \\ c' \\ d' \\ e' \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix}$$

19