ME6406 Machine Vision

Professor Kok-Meng Lee
Georgia Institute of Technology
George W. Woodruff School of Mechanical Engineering
Atlanta, GA 30332-0405
Email: kokmeng.lee@me.gatech.edu

Part 2C: LS Parameter Estimation, K-means Clustering, Curvature and Scale-space filtering

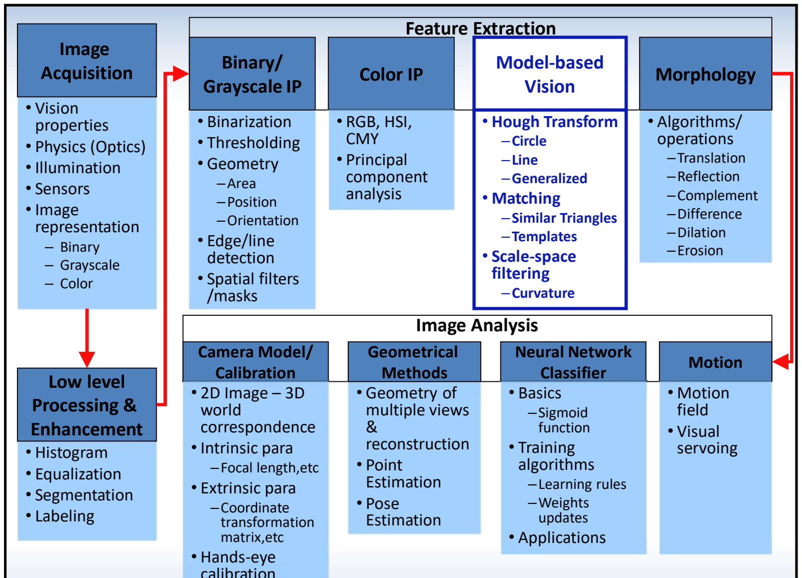
http://kmlee.gatech.edu/me6406

1

Course Outline

- Introduction and low-level processing
  - Physics of digital images, histogram equalization, segmentation, edge detection, linear filters.
- **Model-based Vision**
  - **Hough transform, pattern representation, matching**
- Geometric methods
  - Camera model, calibration, pose estimation
- Neural network for machine vision
  - Basics, training algorithms, and applications
- Color images and selected topics
  - Physics, perception, processing and applications

2



3

Contents

- ❑ **Template Matching**
  - Feature point representation
  - Use of similar triangles plane object matching
  - Transformation parameters
- ❑ **Linear Least-Square (LS) Parameter Estimation**
- ❑ ***K*-means Clustering**
- ❑ **Curvature (feature representation) and Scale-space filtering method**

4

Template matching (http://kmlee.gatech.edu/me6406/)

4. Lee, K-M. and S. Janakiraman, "A Model-based Vision Algorithm for Real-Time Flexible Part-feeding and Assembly," Paper number: MS 92-211. *SME Applied Machine Vision Conf.,* June 1-4, 1992, Atlanta, GA.

In Canvas (“Reading materials” folder)

A. Rattarangsi and T. R. Chin, “Scale-based Detection of corners and planar curves,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 14, no 4m April 1992.

Linear Least-Square (LS) Parameter Estimation

Consider

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1q}x_q &= y_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2q}x_q &= y_2 \\ \cdots a_{ij}x_j \cdots &= y_i \\ \underbrace{a_{p1}x_1 + a_{p2}x_2 + \cdots + a_{pq}x_q}_{[\mathbf{A}]\mathbf{x}} &= \underbrace{y_p}_{\mathbf{y}} \end{aligned}$$

$\mathbf{Ax} = \mathbf{y}$ where

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1q} \\ a_{21} & a_{22} & \cdots & a_{2q} \\ \vdots & \vdots & \vdots & \vdots \\ a_{p2} & a_{p2} & \cdots & a_{pq} \end{bmatrix}; \ \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_q \end{bmatrix}; \ \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{bmatrix}$$

$p < q$ Undercontrained ← More q unknowns than p equations (optimization)

$p = q$ Unique solutions

$p > q$ Overcontrained **← More equations than unknowns (curve-fit and Calibration)**

Minimizing the error measure,

$$E \overset{\text{def}}{=} \sum_{i=1}^{p} (a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{iq}x_q - y_i)^2 = \|\mathbf{Ax} - \mathbf{y}\|^2$$

In matrix form

or $E = \mathbf{e} \bullet \mathbf{e} = \mathbf{e}^{\mathrm{T}}\mathbf{e}$ where $\mathbf{e} = \mathbf{Ax} - \mathbf{y}$. $\boxed{\dfrac{\partial E}{\partial x_i} = 2\dfrac{\partial \mathbf{e}}{\partial x_i} \bullet \mathbf{e} = 0}$ $\boxed{\left[\dfrac{\partial \mathbf{e}}{\partial x_i}\right]^{\mathrm{T}} \mathbf{e} = 0}$

Let the columns of $\mathbf{A}$ are the vectors

$\mathbf{c}_j = \begin{bmatrix} a_{1j} & a_{2j} & \cdots & a_{pj} \end{bmatrix}^{\mathrm{T}}$ where $j = 1, \cdots, q$

then $$\mathbf{e} = \underbrace{\begin{bmatrix} \mathbf{c}_1 & \mathbf{c}_2 & \cdots & \mathbf{c}_q \end{bmatrix}}_{[\mathbf{A}]} \begin{bmatrix} x_1 \\ \vdots \\ x_q \end{bmatrix} - \mathbf{y} = \underbrace{x_1\mathbf{c}_1 + \cdots + x_i\mathbf{c}_i + \cdots + x_q\mathbf{c}_q - \mathbf{y}}$$

$$\Rightarrow \frac{\partial \mathbf{e}}{\partial x_i} = \mathbf{c}_i$$

$$\frac{\partial E}{\partial x_i} = 2\frac{\partial \mathbf{e}}{\partial x_i} \bullet \mathbf{e} = \underbrace{\begin{bmatrix} \mathbf{c}_1^{\mathrm{T}} \\ \vdots \\ \mathbf{c}_q^{\mathrm{T}} \end{bmatrix}}_{\mathbf{A}^{\mathrm{T}}} \begin{bmatrix} \mathbf{Ax} - \mathbf{y} \end{bmatrix} = 0$$

$\Rightarrow \mathbf{A}^{\mathrm{T}}\mathbf{Ax} - \mathbf{A}^{\mathrm{T}}\mathbf{y} = 0$

or $\mathbf{A}^{\mathrm{T}}\mathbf{Ax} = \mathbf{A}^{\mathrm{T}}\mathbf{y}$

$$\mathbf{x} = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A}\right)^{-1}\mathbf{A}^{\mathrm{T}}\mathbf{y}$$

MATLAB command: *x* = pinv(A) *b

K-means Clustering

Introduction to K-means Clustering

- **Clustering**
  - a process of grouping a set of objects into classes of similar objects.
  - A typical clustering approach is iteratively partitioning training data set to learn a partition of the given data space.
- **K-means algorithm** (Unsupervised learning from raw data).
  - A heuristic method for global optimum, exhaustively search all partitions.
  - In principle, optimal partition achieved via minimizing the sum of squared distance to its "representative object" in each cluster.

Training set — Classes of similar object

Euclidean distance $d^2(\underbrace{\mathbf{x}}_{\text{Training set}}, \underbrace{\mathbf{m}_k}_{\text{Classes of similar object}}) = \sum_{n=1}^{N} (x_n - m_{kn})^2$

Minimize $E = \Sigma_{k=1}^{K} \Sigma_{\mathbf{x} \in C_k} d^2(\mathbf{x}, \mathbf{m}_k)$

Cluster k

Introduction to K-means Clustering

- Given the cluster number K, find a partition of K *clusters* to optimize the chosen partitioning criterion (cost function):
  - each cluster is represented by the center of the cluster and the algorithm converges to stable centriods of clusters.
  - The simplest partitioning method for clustering analysis and widely used in data mining applications.
- Basic steps:

Initialization: set seed points (randomly)

1) Assign each object to the cluster of the nearest seed point measured with a specific distance metric
2) Compute new seed points as the centroids of the clusters of the current partition (the centroid is the centre, i.e., **mean point**, of the cluster)
3) Go back to Step 1), stop when no more new assignment (i.e., membership in each cluster no longer changes)

K-means algorithm

- Given a training set, $\{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(m)}\}$, m number, n dim., $\mathbf{x}^{(i)} \in R^n$

1. Initialize cluster centroids ($\mu_1, \mu_2, \ldots \mu_k$), $\mu_j \in R^n$ randomly.
2. Repeat until convergence:

{

(Assign step) For every i, set $\quad c(i) := \arg\min_j \left\| \mathbf{x}^{(i)} - \boldsymbol{\mu}_\mathbf{j} \right\|^2$

(Update step) For each j, set $\quad \boldsymbol{\mu}_\mathbf{j} := \dfrac{\sum_{i=1}^{m} \mathbf{1}\{c^{(i)} = j\} \mathbf{x}^{(i)}}{\sum_{i=1}^{m} \mathbf{1}\{c^{(i)} = j\}}$

}

- Matlab command:

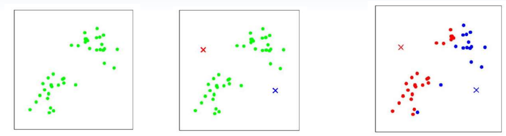
```
[cluster_idx, cluster_center,sumd] = kmeans(XY,cnum,'distance','sqEuclidean',
'Replicates',20);
```

Sec. 16.4

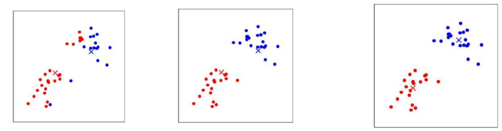
- **Termination conditions** (several possibilities):
  - A fixed number of iterations.
  - Partition unchanged.
  - Centroid positions don't change.
- **Convergence** (A state in which clusters don't change)
  - K-means is a special case of a general procedure known as the **Expectation Maximization** (EM) algorithm, which is known to converge.
  - Number of iterations could be large but in practice usually isn't.

K-means method

K=2



(a) 2D data (b) Random generate mean (c) Assign cluster



(d) Update mean (e) Assign cluster (f) Assign cluster (converge)

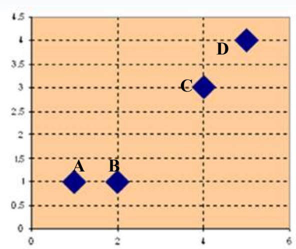
Expectation-maximization(EM) algorithm:

(E-step) Expectation (assign) step: (c), (e), (f): assign cluster

(M-step) Maximum (update) step: (b), (d): update mean

Example: Suppose we have 4 features (A, B, C D) and each has two attributes (a, b). The task is to group these objects into $K=2$ group of features.

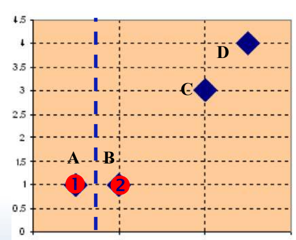
Features	Attribute 1 (x): a	Attribute 1 (y): b
A	1	1
B	2	1
C	4	3
D	5	4



Step 0: Initialization: set seed points (randomly)

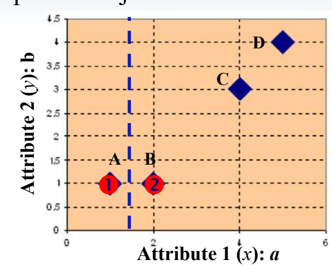
1. $c_1(1, 1)$: Group 1
2. $c_2(2, 1)$: Group 2

$$\begin{array}{cccccc} i & 1 & 2 & 3 & 4 & \text{Centroid} \\ & A & B & C & D & \mathbf{c}_1 \quad \mathbf{c}_2 \\ \begin{matrix} x \\ y \end{matrix} & \multicolumn{4}{c}{\begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix}} & \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \end{array}$$



Example: Suppose we have 4 features (A, B, C D) and each has two attributes (a, b). The task is to group these objects into $K=2$ group of features.

$$\begin{array}{cc} i & 1 \ 2 \ 3 \ 4 \\ & A \ B \ C \ D \\ \begin{matrix} x \\ y \end{matrix} & \begin{bmatrix} 1 & 2 & 4 & 5 \\ 1 & 1 & 3 & 4 \end{bmatrix} \end{array} \quad \begin{array}{cc} \text{Group} & \text{Centroid} \\ \mathbf{c}_1 & (c_{1x}=1,\ c_{1y}=1) \\ \mathbf{c}_2 & (c_{2x}=2,\ c_{2y}=1) \end{array}$$



Step 1: Use initial seed points for partitioning. Assign each object to the cluster with the nearest seed point.

$$\mathbf{D}^0 = \begin{bmatrix} d_x(i,c_1) \\ d_x(i,c_2) \end{bmatrix} = \begin{bmatrix} \sqrt{(x_i - c_{1x})^2 + (y_i - c_{1y})^2} \\ \sqrt{(x_i - c_{2x})^2 + (y_i - c_{2y})^2} \end{bmatrix}$$

For example, features C and D

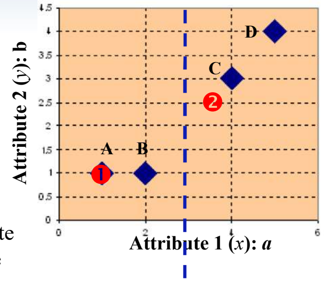
$$d(\mathbf{C}, c_1) = \sqrt{(4-1)^2 + (3-1)^2} = 3.61$$
$$d(\mathbf{C}, c_2) = \sqrt{(4-2)^2 + (3-1)^2} = 2.83$$
$$d(\mathbf{D}, c_1) = \sqrt{(5-1)^2 + (4-1)^2} = 5$$
$$d(\mathbf{D}, c_2) = \sqrt{(5-2)^2 + (4-1)^2} = 4.24$$

$$\begin{array}{cccccc} i & A & B & C & D & \text{Cluster} \quad \text{Centroid} \\ \mathbf{D}^0 = & \multicolumn{4}{c}{\begin{bmatrix} 0 & 1 & 3.61 & 5 \\ 1 & 0 & 2.83 & 4.24 \end{bmatrix}} & \begin{matrix} \mathbf{c}_1 & (1,1) \\ \mathbf{c}_2 & (2,1) \end{matrix} \end{array}$$

Example: Suppose we have 4 features (A, B, C D) and each has two attributes (a, b). The task is to group these objects into $K=2$ group of features.

i	A	B	C	D	Group	Centroid
$\mathbf{D}^0 =$	0	1	3.61	5	$\mathbf{c}_1$	(1, 1)
	1	0	2.83	4.24	$\mathbf{c}_2$	(2, 1)

❶ ❷ Assign the membership to objects



New Partition

	A	B	C	D
x	1	2	4	5
y	1	1	3	4

Step 2: Compute new centroids of the current partition.

a. Knowing the members of each cluster, compute the new centroid of each group based on these new memberships.

$$c_1 = (1,1); \quad c_2 = \left(\frac{2+4+5}{3}, \frac{1+3+4}{3}\right) = \left(\frac{11}{3}, \frac{8}{3}\right)$$

b. Compute the distance of all objects to the new centroids

i	A	B	C	D	Group	Centroid
$\mathbf{D}^1 =$	0	1	3.61	5	$\mathbf{c}_1$	(1, 1)
	3.14	2.36	2.83	4.24	$\mathbf{c}_2$	(11 / 3, 8 / 3)

❶ ❷

c. Renew membership based on new centroids.

17

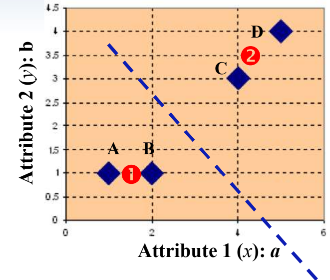
Example: Suppose we have 4 features (A, B, C D) and each has two attributes (a, b). The task is to group these objects into $K=2$ group of features.

Step 3: Repeat the first two steps until its convergence .

a. Knowing the members of each cluster, compute the new centroid of each group based on these new memberships.

$$c_1 = \left(\frac{1+2}{2}, \frac{1+1}{2}\right) = \left(1\frac{1}{2}, 1\right)$$

$$c_2 = \left(\frac{4+5}{2}, \frac{3+4}{2}\right) = \left(4\frac{1}{2}, 3\frac{1}{2}\right)$$



New Partition

	A	B	C	D
x	1	2	4	5
y	1	1	3	4

b. Compute the distance of all objects to the new centroids

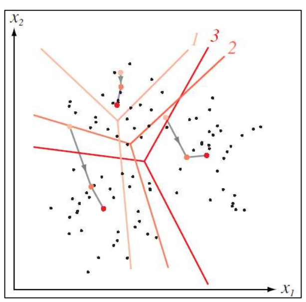
	A	B	C	D	Group	Centroid
$\mathbf{D}^2 =$	0.5	0.5	3.20	4.61	$\mathbf{c}_1$	(1, 1)
	4.30	3.54	0.71	0.71	$\mathbf{c}_2$	(9 / 2, 7 / 2)

❶ ❷

c. Stop due to no new assignment Membership in each cluster no longer change.

18

How K-means partitions?



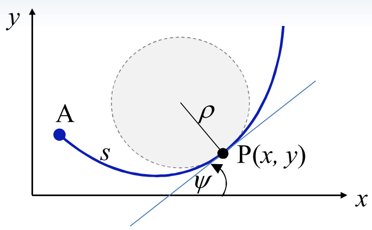
- When K centroids are set/fixed, they partition the whole data space into K mutually exclusive subspaces to form a partition.
- A partition amounts to a Voronoi Diagram
- Changing positions of centroids leads to a new partitioning.

19

Curvature (Feature Representation) and Scale-space filtering method

20

Curvature

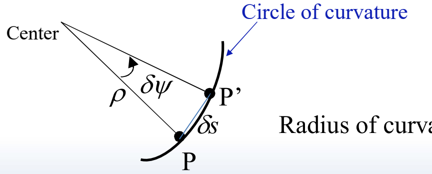


- A is a fixed point.
- P(*x*, *y*) is a variable point.
- Let *s* is the arc-length AP, and ψ is the angle between the *x*-axis and the tangent at P.

$$\tan\psi = \frac{dy}{dx}$$

Curvature is the rate of turning of the tangent with respect to the arc:

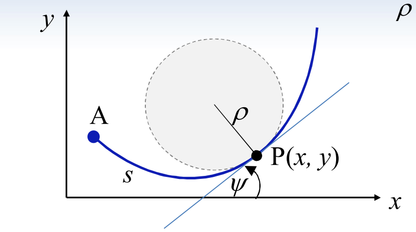
$$\text{Curvature } K = \frac{d\psi}{ds}$$



Radius of curvature, $\rho = \lim_{s\to 0}\frac{\delta s}{\delta\psi} = \frac{ds}{d\psi} = \frac{1}{K}$

Curvature

$$\rho = \lim_{s\to 0}\frac{\delta s}{\delta\psi} = \frac{ds}{d\psi} = \frac{1}{K}$$



$$\sec^2\psi\,\frac{1}{\rho} = \frac{d^2y}{dx^2}\cos\psi$$

$$\Rightarrow \rho = \frac{\sec^3\psi}{\dfrac{d^2y}{dx^2}} = \frac{\left(1+\tan^2\psi\right)^{3/2}}{\dfrac{d^2y}{dx^2}}$$

$$\tan\psi = \frac{dy}{dx}$$

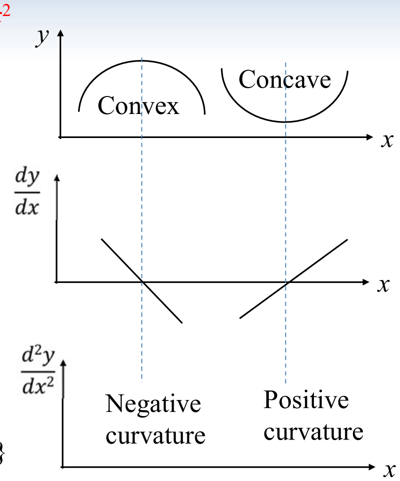
Differentiating with respect to *s*,

$$\underbrace{\sec^2\psi\,\frac{d\psi}{ds}}_{1/\rho} = \frac{d}{ds}\left(\frac{dy}{dx}\right) = \underbrace{\frac{d}{dx}\left(\frac{dy}{dx}\right)\frac{dx}{ds}}_{\frac{d^2y}{dx^2}\cos\psi}$$

$$K = \frac{1}{\rho} = \frac{\dfrac{d^2y}{dx^2}}{\left[1+\left(\dfrac{dy}{dx}\right)^2\right]^{3/2}}$$

The sign of K depends on d^2y/dx^2

$$K = \frac{1}{\rho} = \frac{\dfrac{d^2y}{dx^2}}{\underbrace{\left[1+\left(\dfrac{dy}{dx}\right)^2\right]^{3/2}}_{\text{Always positive}}}$$



In 2D images, $K = \{x(s), y(s)\}$

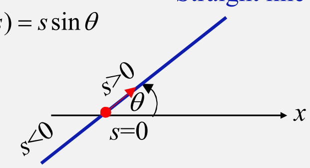
In parametric form, $x(s)$ and $y(s)$

Examples

$x(s) = s\cos\theta$

$y(s) = s\sin\theta$

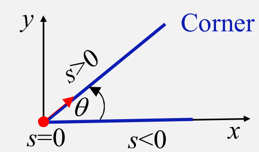
Straight line



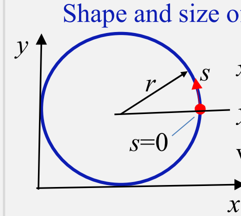
$$x(s) = \begin{cases} -s & s<0 \\ s\cos\theta & s\geq 0\end{cases}$$

$$y(s) = \begin{cases} 0 & s<0 \\ s\sin\theta & s\geq 0\end{cases}$$

Corner



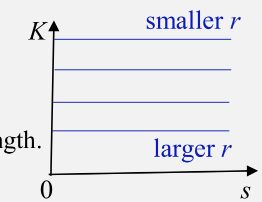
Shape and size of circle.



$x(s) = r\cos(2\pi s/N)$

$y(s) = r\sin(2\pi s/N)$

where N is the path length.



In parametric form,

$$x(s);\ \dot{x} = \frac{dx}{ds} \text{ and } \ddot{x} = \frac{d^2x}{ds^2}$$

Similarly,

$$y(s);\ \dot{y} = \frac{dy}{ds} \text{ and } \ddot{y} = \frac{d^2y}{ds^2}$$

Thus,

$$y' = \frac{\dot{y}}{\dot{x}} \text{ where } y' = \frac{dy(s)}{dx(s)}$$

$$y'' = \frac{d^2y}{dx^2} = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^3}$$

Recall

$$K(s) = \frac{y''}{\left[1+(y')^2\right]^{3/2}} = \frac{\frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\dot{x}^3}}{\left[1+\left(\frac{\dot{y}}{\dot{x}}\right)^2\right]^{3/2}}$$

$$K(s) = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\left(\dot{x}^2+\dot{y}^2\right)^{3/2}}$$

Scale-space filtering method

Let $g_\sigma(s) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(\frac{-s^2}{2\sigma^2}\right)$ be the Gausspan filter.

Smoothing the curve with $g_\sigma(s)$is the same as convolving with $x(s)$ and $y(s)$. Mathematically,

$$X(s,\sigma) = x * g_\sigma = \int_{-\infty}^{\infty} x(s-u)g_\sigma(u)du$$

$$Y(s,\sigma) = y * g_\sigma = \int_{-\infty}^{\infty} y(s-u)g_\sigma(u)du$$

X(s, σ) and Y(s, σ) are the x and y coordinates of the smoothed curve, respectively; and * is the convolution operator.

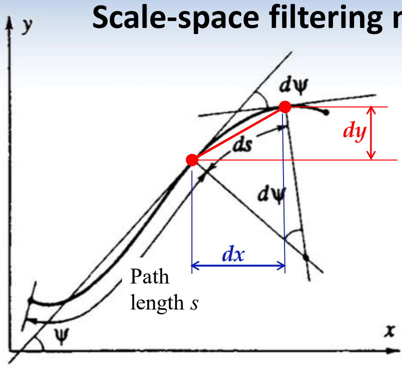
Recall curvature,

$$K\left[x(s), y(s)\right] = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\left(\dot{x}^2+\dot{y}^2\right)^{3/2}}$$

$$K_\sigma\left[X(s,\sigma), Y(s,\sigma)\right] = \frac{\dot{X}\ddot{Y} - \dot{Y}\ddot{X}}{\left(\dot{X}^2-\dot{Y}^2\right)^{3/2}}$$

where $\dot{X} = \frac{\partial X}{\partial s}, \dot{Y} = \frac{\partial Y}{\partial s}, \ddot{X} = \frac{\partial^2 X}{\partial s^2}, \ddot{Y} = \frac{\partial^2 Y}{\partial s^2}$

Scale-space filtering method (cont.)



$$\frac{dx}{ds} = \cos\psi;\quad \frac{dy}{ds} = \sin\psi$$

$$\dot{x}^2 + \dot{y}^2 = \cos^2\psi + \sin^2\psi = 1$$

$$K\left[x(s), y(s)\right] = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{\left(\dot{x}^2+\dot{y}^2\right)^{3/2}} = \dot{x}\ddot{y} - \dot{y}\ddot{x}$$

$$K(s,\sigma) = \dot{X}(s,\sigma)\ddot{Y}(s,\sigma) - \dot{Y}(s,\sigma)\ddot{X}(s,\sigma)$$

Detection of dominant points and corners

Scale-space curvature optimization

$$\partial K(X,Y)/\partial s = 0$$

$$\dot{X}(s,\sigma)\ddot{Y}(s,\sigma) - \dot{Y}(s,\sigma)\ddot{X}(s,\sigma) = 0$$

Scale-space filtering method (cont.)

Let $F(s,\ \sigma)$ represents the function, $X(s,\ \sigma)$ or $Y(s,\ \sigma)$; and

$F(s, 0)$ is the unsmooth coordinates, $x(s)$ and $y(s)$, respectively.

$$F(s,\sigma) = \int_{-\infty}^{\infty} f(s-u)g_\sigma(u)du = \int_{-\infty}^{\infty} g_\sigma(s-u)f(u)du$$

$$f * g_\sigma = g_\sigma * f$$

$$\frac{dF(s,\sigma)}{ds} = \frac{d}{ds}(f * g_\sigma) = \int_{-\infty}^{\infty}\frac{df(s-u)}{ds}g_\sigma(u)du = \int_{-\infty}^{\infty}\frac{dg_\sigma(s-u)}{ds}f(u)du$$

$$\dot{F}(s,\sigma) = \frac{d\left(F(s,0) * g_\sigma\right)}{ds} = \frac{d\left(F(s,0) * g_\sigma\right)}{ds} * g_\sigma$$

Discrete approximation:

$$F(s,\sigma) = f * g_\sigma = \sum_{u=-5\sigma}^{5\sigma} f(s-u)g_\sigma(u)$$

$$\frac{dF(s,\sigma)}{ds} \approx F(j+1,\sigma) - F(j-1,\sigma)$$

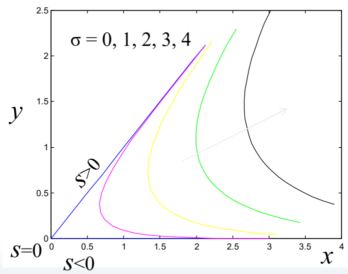
$$\frac{d^2F(s,\sigma)}{ds^2} \approx F(j+1,\sigma) - 2F(j) + F(j-1,\sigma)$$

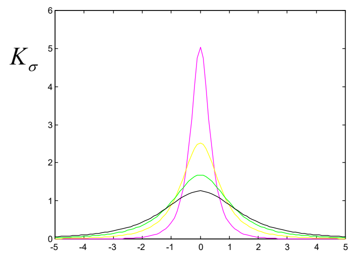
Scale-space filtering method (cont.)

$$x(s) = \begin{cases} -s & s < 0 \\ s\cos\theta & s \geq 0 \end{cases}$$

$$y(s) = \begin{cases} 0 & s < 0 \\ s\sin\theta & s \geq 0 \end{cases}$$

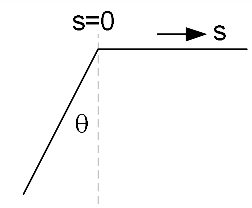
$$K_\sigma(s) = \dot{X}\ddot{Y} - \dot{Y}\ddot{X}$$





Detection of dominant points and corners

$$\boxed{K_\sigma(X,Y) = \dot{X}\ddot{Y} - \dot{Y}\ddot{X}}$$



Scale-space curvature optimization

$$\partial K(X,Y)/\partial s = 0$$

$$\dot{X}(s,\sigma)\ddot{Y}(s,\sigma) - \dot{Y}(s,\sigma)\ddot{X}(s,\sigma) = 0$$

$$-\dot{g}_\sigma \cos\theta = 0$$

At **0 = 90°, it becomes a straight line.**

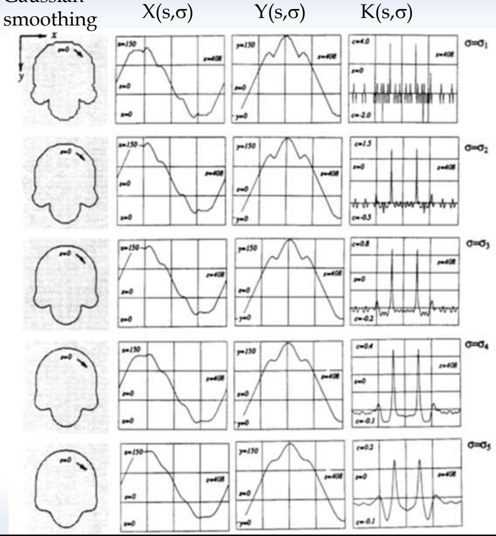
In the case of **0 > -90°** but not equal to *90°*, $\dot{g}_\sigma = 0$

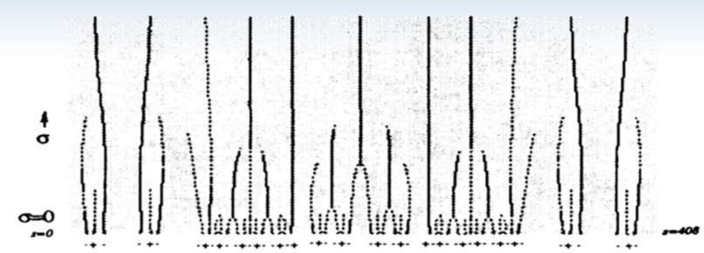
Therefore, the only solution is at s = 0 independent of the corner angle **0, and the scale parameter** ***σ. This produces a*** vertical line in scale space, that is, the absolute maxima occur at the same contour location independent of smoothing.

Example

Result sof the Gaussian smoothing

X(s,σ) Y(s,σ) K(s,σ)





Example

Scale space map of maxima of absolute curvature. The horizontal axis is the arc length of the curve at ***σ = 0. The vertical axis is the Gaussian*** **function parameter** ***D determining the degree of smoothing. The line pattern*** **in the map represents the locations of the local maxima of the curvature. A +sign indicates downward concavity, and a - sign indicates upward concavity.**

Example: curvature method

1. Find the outside boundary
   bwboundaries(~BW, 'noholes')
2. Gaussian smoothing
   conv.m
3. Calculate curvature
   $K(s) = \dot{X}(s)\ddot{Y}(s) - \ddot{Y}(s)\dot{X}(s)$
   $\dot{X}(s) \cong diff(X)$, $\dot{Y}(s) \cong diff(Y$
   $\ddot{X}(s) \cong diff(\dot{X})$, $\ddot{Y}(s) \cong diff(\dot{Y}$
4. Find peaks
   findpeaks.m

Example: Gaussian smoothing

```
x=[x0;x0];y=[y0;y0];
% x, y are the xy position of edges
n=length(x);
s=linspace(-n/2,n/2,n);
sigma=5;
g0=1/sqrt(2*pi*sigma^2)*exp(-s.^2/(2*sigma^2));
g=[g0 g0]';
%convolve x(s), y(s) with Gaussian
X=conv(x,g,'same');X=X(1:end/2);
% same: Returns the central part of the convolution
of the same size as x
Y=conv(y,g,'same');Y=Y(1:end/2);
```