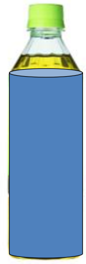
Computer vision class 2014

Object Representation I

Oct. 22. 2014
Bo Zheng
zheng@cvl.iis.u-tokyo.ac.jp

The first example:

Representing an object



Cylinder (shape)

Cap (color)

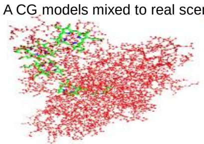
Plastic cover (texture)

Can we find some features that can be used to characterize the object?

Object representation in other fields

- **Computer Graphics (CG) or Augmented Reality (AR)**
  - how to realistically render a synthetic image
- **Computer Visualization**
  - how to make a visual form enabling the user to observe the invisible information

A CG models mixed to real scene



bacterial protein

Computer Vision

Tries to answer the question:

What is where?

Object Representation in Computer Vision

- How to make a digital form enabling computer to understand visual information.

$\Rightarrow (0.2, 0.7, \cdots)$

$\Rightarrow (0.5, 0.3, \cdots) \leftrightarrow (0.5, 0.2, \cdots)$

Preserved in computer

Computer vision system

	e.g.
Data acquisition	2D/3D Images
Object representation (Our interest)	Characterize objects with features (feature detection and description)
Internal processing	Object Recognition, Tracking, Retrieval an object from DB…
High-level processing	Scene understanding, Face Recognition, tracking a car, localizing a picture …

These objects are recognized by…

pixels? Colors? Textures? Shapes? …

Outline

- 2D representation (for RGB image)
  - basics
  - research in the state of arts
- Sparse representation
  - basics
  - research in the state of arts

Today's class

- 3D representation
  - basics
  - research in the state of arts

Next class

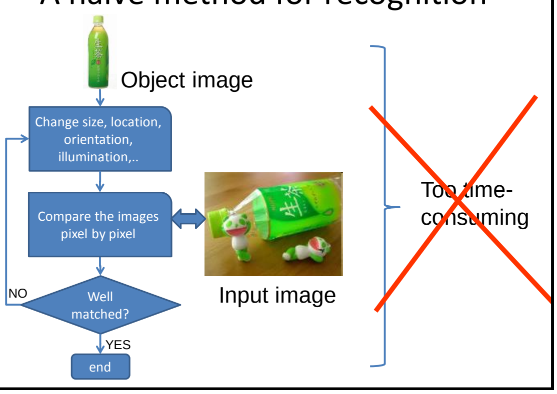
- Beyond "what is where": new trends in CV

Part I: Basics on 2D representation

Problems on 2D representation

- Appearance of objects change in different conditions
  - Location and orientation
  - scale or viewpoint
  - noises and occlusions
  - illumination conditions

A naive method for recognition



An efficient way:

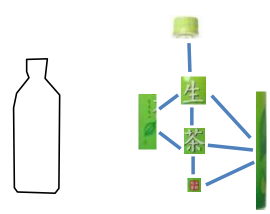
using the visual features

(sparse and special)

- Local representation

- Appearance at each local part

- Global representation



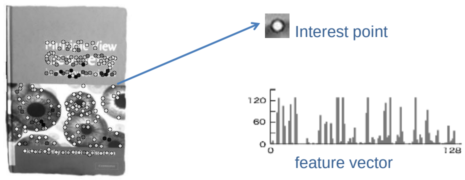
The shape of whole object

Structure (relations) of local features

Local feature representation

Step1: Feature detection

Step2: Feature description

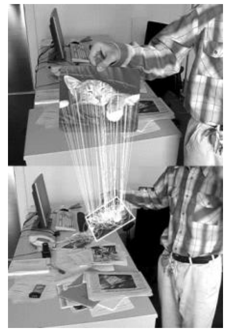


finding the special points or areas

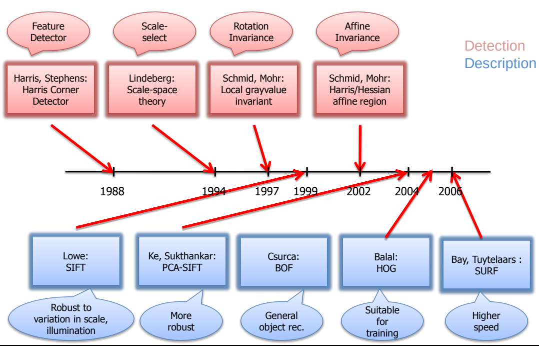
Characterizing each interest point with digital numbers (feature vector)

Example: template matching

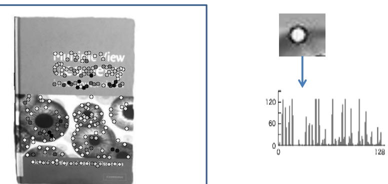
[M. Özuysal, PAMI'10]



The history of contribution



First step



Step1: feature detection

finding the special local areas

Step2: feature description

Characterizing each area with digital numbers

Image Feature Detection

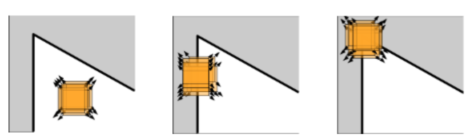
Goal: finding interest points or areas on an image invariant to image size, orientation, view point, illumination…

Approaches:

- corner detection
- Scale invariant detection
- Rotation/orientation invariant detection
- Affine invariant detection

Corner definition

At a corner, the image intensity will change largely in multiple directions.



"flat" region: no change in all directions

"edge": no change along the edge direction

"corner": significant change in all directions

Harris: how to determine a corner

Algorithm:

Autocorrelation matrix of the image I(x,y);

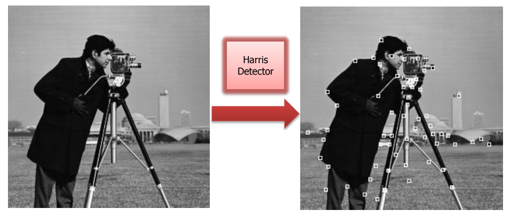
$$M = G(s) \otimes \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Two eigenvalues (λ_1, λ_2) of M : principal curvatures of the point.

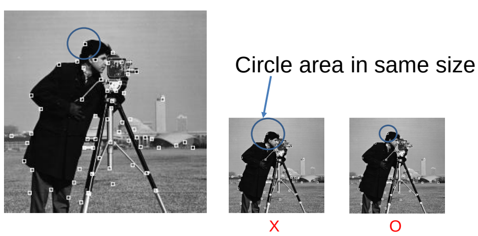
$\lambda_1 \approx 0$	and $\lambda_2 \approx 0$:	"Flat" (No feature)
$\lambda_1 \approx 0$	and $\lambda_2 >> 0$:	"Edge"
$\lambda_1 >> 0$	and $\lambda_2 >> 0$:	"Corner"

Harris Corner

(Harris, AVC1988)



Problem arise when image size changes

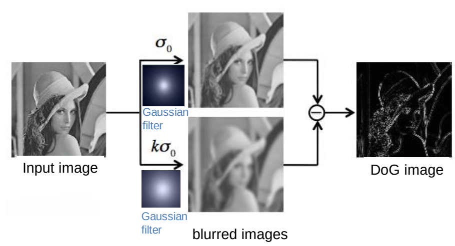


We want to know the relative scale (size) of the region around the interest point, when images are zoomed in/out?

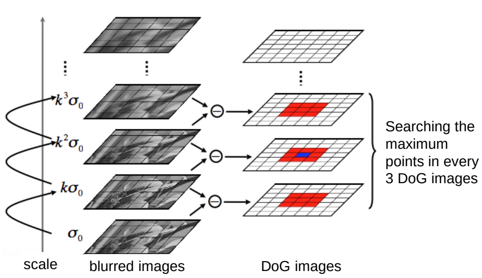
A solution: Detecting scale invariant interest points

[Lowe, IJCV'04]

Difference-of-Gaussian (DoG):

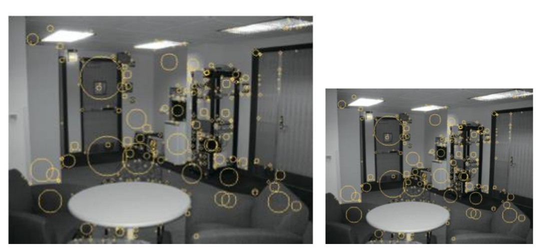


Constructing the scale (σ) space



The result:

finding the relative scale for each interest point

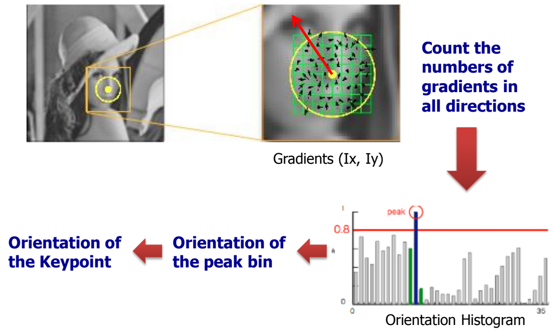


Problem arise when the images are captured in different orientation

We need to know the orientation of each interest point!

Orientation detection (in SIFT)

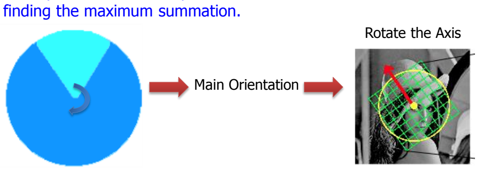
[Lowe, IJCV'04]



Orientation detection (in SURF)

[H. Bay et al. ECCV06]

1, sliding the sector around center of feature.
2, sum up the harr filter responses in each sector.
3, finding the maximum summation.



Problem arise if the pictures are taken in different view points?

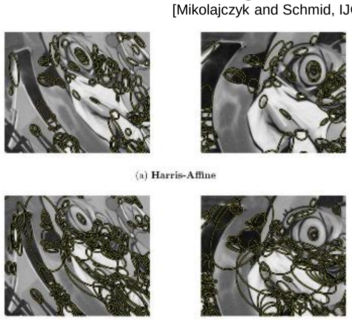
We need to know the affine-transformed information of each interest point.

Harris/Hessian Affine Region

[Mikolajczyk and Schmid, IJCV'04]

Harris Affine:

Hessian Affine:



(a) Harris-Affine

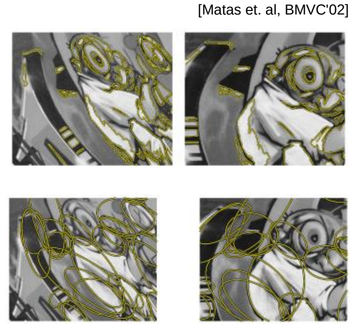
(b) Hessian-Affine

Others

[Matas et. al, BMVC'02]

MSER (Maximally stable extremal region):

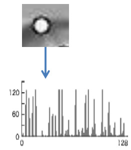
EBR (Edge-based region):



Second step

Step1: feature detection

finding the special local areas



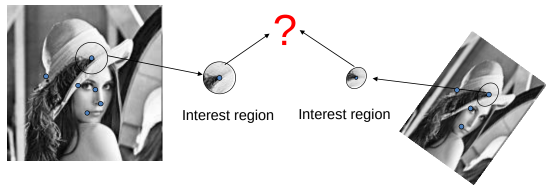
Step2: feature description

Characterizing each area with digital numbers

Feature-Based Descriptors

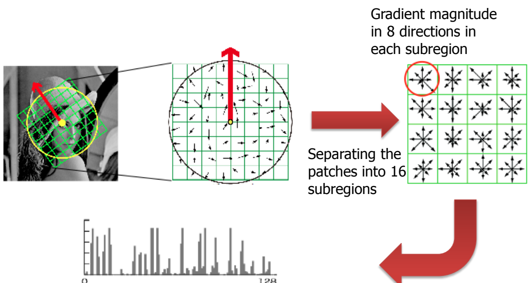
Goal: to characterize local regions with some digital numbers (feature vector) for the image matching.

Popular Approaches: SIFT, SURF…



Gradients Descriptor (SIFT)

[Lowe, IJCV'04]

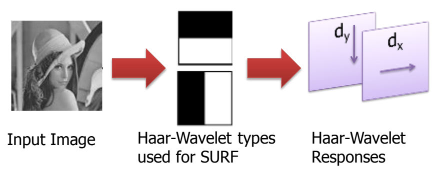


Feature Vector (8*16=128 Dimensions)

Speeded Up Robust Feature (SURF)

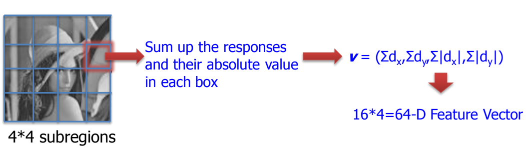
[Bay, ECCV2006]

- Approach: use Haar-Wavelet Responses to characterize the keypoints.



Speeded Up Robust Feature (SURF)

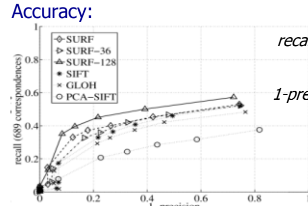
[Bay, ECCV2006]



Comparison of SIFT and SURF

(Bay, ECCV2006)

Accuracy:



$$recall = \frac{\text{number of } \textit{true-matched}}{\text{total number of } \textit{correspondences}}$$

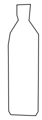
$$\textit{1-precision} = \frac{\text{number of } \textit{false-matched}}{\text{total number of } \textit{matched}}$$

Speed:

	U-SURF	SURF	SURF-128	SIFT
time (ms):	255	354	391	1036

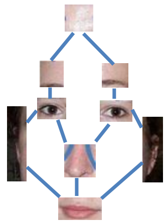
Computation Time on an example image (800*640)

Global representation

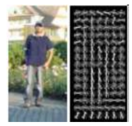


Shape descriptor

Bag of Features



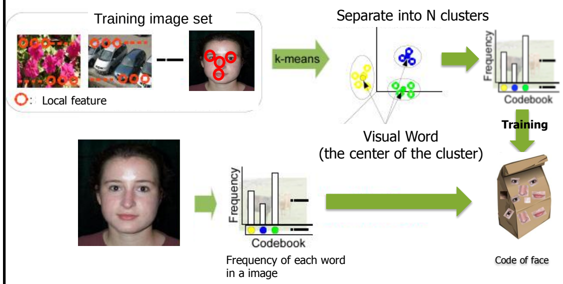
Constellation



HoG (Histogram of Oriented Gradients)

Example: Bag Of Features（BOF） for image understanding

(Csurka, 2004)



Research in the state of art

Critical Nets and Beta-Stable Features for Image Matching

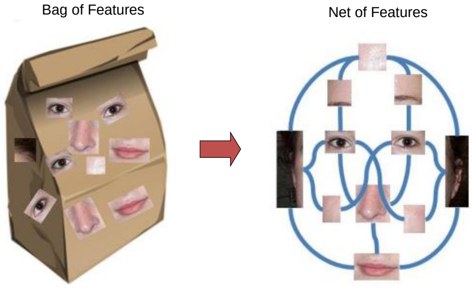
ECCV 2010, oral

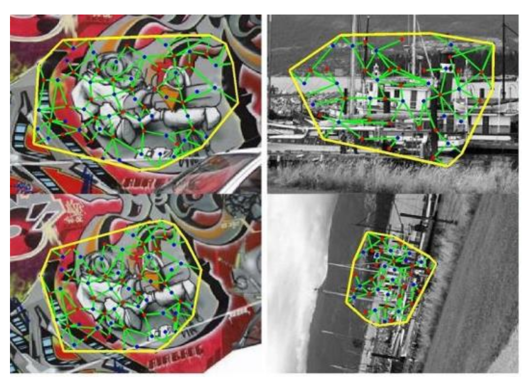
Steve Gu, Ying Zheng and Carlo Tomasi
Department of Computer Science
Duke University

idea

Bag of Features

Net of Features





Outline

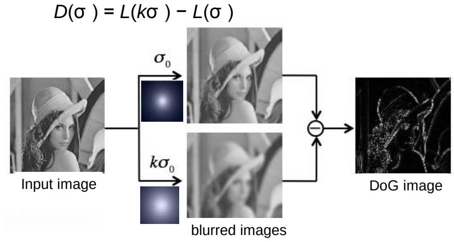
- Beta-stable features detection
- Critical nets construction
- Application to Image matching

Review of DoG

[Lowe, IJCV'04]

Difference-of-Gaussian (DoG):

$L(\sigma) = G(\sigma) * I(x, y)$
$D(\sigma) = L(k\sigma) - L(\sigma)$

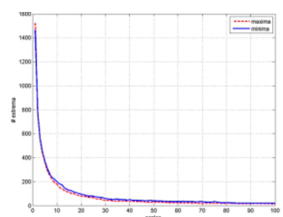


Scale space revisited

approximate Laplacian (DoG) images changed in different scales

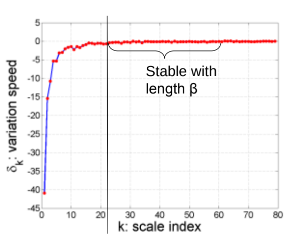
Scale space revisited

Compute the number of ``convex'' (bright) and ``concave'' (dark) regions



Scale space revisited

The variation speed of the DoG over scale



Beta-stable scale

- Scale k is called ``beta-stable'' if : the number of convex regions remains a constant within a scale interval of length beta.

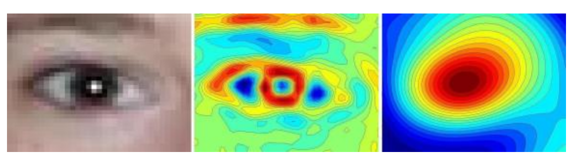


Fig. 3. From left to right: An image patch of a human eye and its DoG at scales 2 (middle) and 25 (right). Scale k = 25 is 10-stable.

Beta-stable features

- The extrema of the DoG function computed at the smallest beta-stable scale

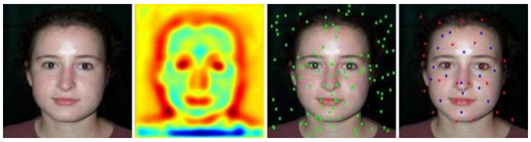
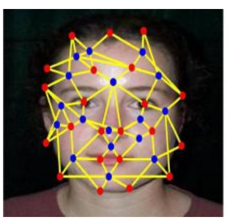


Fig. 4. From left to right: Original image; The 10-stable DoG image; SIFT features (green); 10-stable features. Red and blue dots are maxima and minima of L10.

Critical Nets

Definition of Critical Nets

- A minimum A is connected to a maximum B if an ascending path goes from A to B
- Such a graph is called a **critical net**



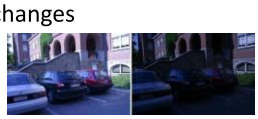
Ascending paths

- Define practically repeatable connections between beta-stable features
- Connection:

17	24	1	8	15
23	5	7	14	16
4	6	13	20	22
10	12	19	21	3
11	18	25	2	9

Critical Nets

- **Observation 1**: ascending paths are invariant under ``monotonic'' changes



- **Observation 2**:

  the higher the values of a point, the lower the probability this point can expand further

Image Matching

Example of dual SIFT descriptors

Note: using the edge information, no need to detect the scale and orientation

Matching results Comparison

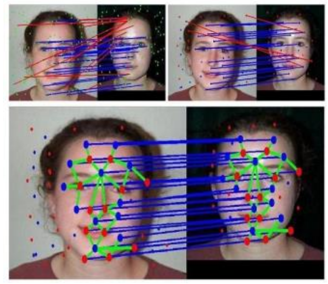
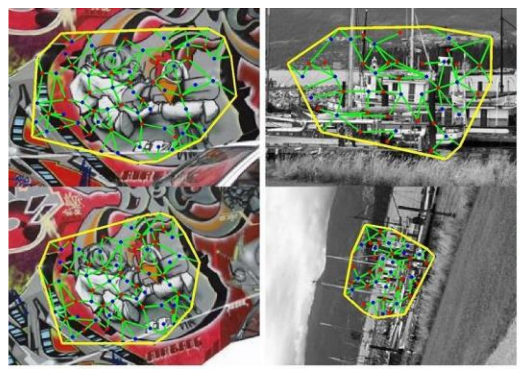
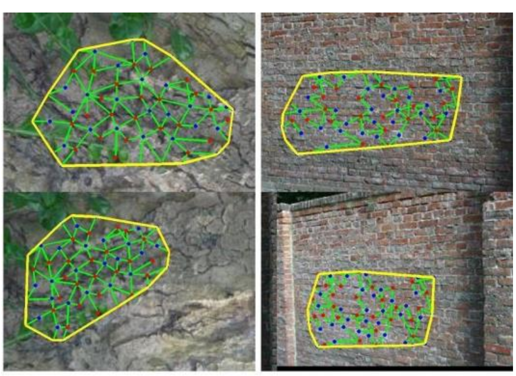


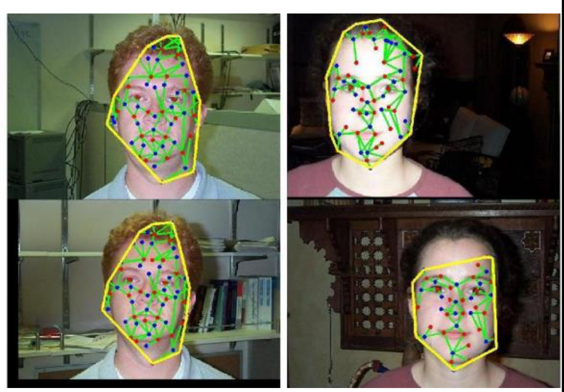
Fig. 7. Top left: SIFT; Top right: the 10-stable features and the matching result without using the critical net connections; Bottom: same 10-stable features, but with matching based on the critical net where dual SIFT descriptors are used.

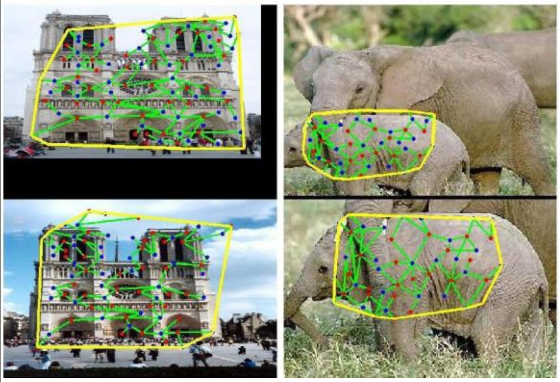
Compared to SIFT

- Reduce the number of parameters
  - –No need to determine the histogram peaks
  - –No need to assign multiple directions
- Richer local descriptor (dark-bright pattern)
- And better repeatability in matching









Other Contributions

- Critical nets are simple graphs that are invariant under affine and monotonic changes
- A more reliable geometrical reference: rotation and scale

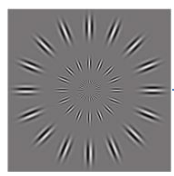
Homepage of this research:

http://www.cs.duke.edu/~steve/cnet.html

Download
1) Papers
2) Matlab code

Part II: Sparse representation

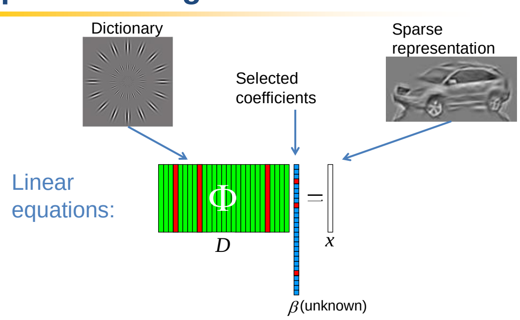
An Example: sparse representation



Selecting a small number of the elements to represent an object

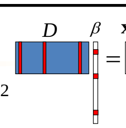
Elements in dictionary
[Wu, IJCV'09]

Sparse coding



$$\min_{\beta} \|\beta\|_0 \quad s.t. \quad \|D\beta - x\|_2^2 \le \varepsilon^2$$

How to solve the problem

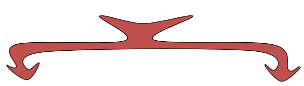


$$\min_{\beta} \|\beta\|_0 \quad s.t. \quad \|D\beta - x\|_2^2 \le \varepsilon^2$$

finding a representation with the smallest number of bases from an dictionary

↓

NP-hard !



Greedy methods

build up the approximation coefficients one by one

Relaxation methods

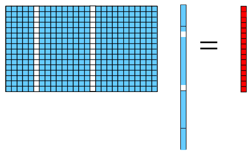
process all the coefficients simultaneously

Greedy methods

$$\min_{\beta} \|\beta\|_0 \quad s.t. \quad \|D\beta - x\|_2^2 \le \varepsilon^2$$

Solve Instead

$$\min_{\beta} \|D\beta - x\|_2$$



Greedy methods

- Matching Pursuit (MP) [Mallat et al, Signal Pro. 1993]
- OMP [Mallat et al, constructive approximation, 1997]
- OMP [Tropp Info. T. 2006]
- StOMP, SP, CoSaMP…

Relaxation methods

Ex. **Basis Pursuit**
[Chen, Donoho, Saunders ('95)]

$$\min_{\beta} \|\beta\|_0 \quad s.t. \quad \|D\beta - x\|_2^2 \le \varepsilon^2$$

Solve Instead

$$\min_{\beta} \|\beta\|_1 \quad s.t. \quad \|D\beta - x\|_2^2 \le \varepsilon^2$$

$$\min_{\beta} \|D\beta - x\|_2^2 + \lambda \|\beta\|_1$$

Linear programming or interior point methods

- BP [Donoho, Info. Theory 2006]
- l1 minimization [Candes et al, Info. Theory 2006]
- an interior method for l1-regularized least squares [Kim et al., Signal Pro.]

Why sparse is good?

Briefly review applications

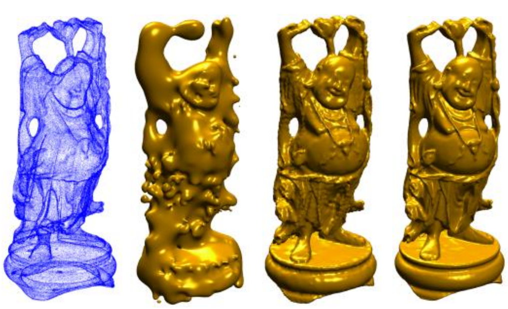
Image representation- Gabor wavelets

[Mendels et al. ('06)]

20 bases 200 bases

Shape representation- using RBF basis

[Carr et al. (SIGGRAPH 01)]



544,000 point cloud → 8000 control points

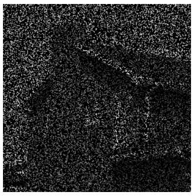
Denoising - DCT wavelets

[Mairal, Elad & Sapiro, ('06)]

Original 'Barbara' image | Separated texture | Recovered image

Image completion

[Mairal, Elad & Sapiro, ('08)]



30% pixels removed

recovered image

Inpainting – using learned dictionary

[Mairal, Bach, Ponce, Sapiro & Zisserman, ('09)]

Image with overlapped texts

Inpainting – using learned dictionary

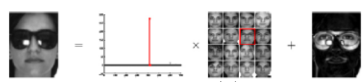
[Mairal, Bach, Ponce, Sapiro & Zisserman, ('09)]

Inpainting result

Robust Face Recognition

[J. Wright and Y.Ma, PAMI'09]

- Sparse representation

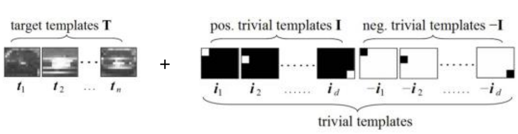


- Occlusion compensation Adding the redundant bases to the dictionary

$$\mathbf{y} = (A \mid I)\begin{pmatrix}\mathbf{x}\\ \mathbf{e}\end{pmatrix} = B\mathbf{w}$$

Object Tracking

[X. Mei and H. Ling, ICCV09]



Object Tracking

[X. Mei and H. Ling, ICCV09]

More recently

- Matrix factorization
  - Low rank minimization
  - Sparse PCA
  - Robust PCA

…

Research in the state of arts

Journal paper

Y. N. Wu, Z.Z. Si, H. Gong and S.-C. Zhu (UCLA)

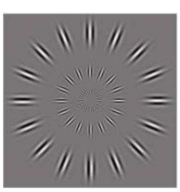
International Conference on Computer Vision (ICCV) 2007
International Journal of computer vision (IJCV) 2009

Learning Active Basis Model for Object Detection and Recognition

Motivation

- Design a deformable template to model a set of images of a certain object category.
- The template can be learned from example images.

Dictionary construction using Gabor wavelets



$\{B_{x,y,s,\alpha}\}$

location scale orientation

Each element can be formed by a Gabor wavelet:

$$G(x,y) \propto \exp\{-\frac{1}{2}[\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}]\}e^{ix}$$

Linear additive image model

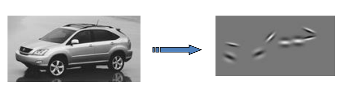
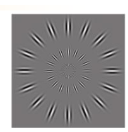


Image reconstruction by matching pursuit.



$$I = \sum_{i=1}^{n} c_i B_i + \varepsilon, \quad \{B_i, i=1,\ldots,n\}$$

selected from a dictionary of Gabor wavelet elements $\{B_{x,y,s,\alpha}\}$

location scale orientation

Two extensions:

1. Encoding a single image → Simultaneously encoding a set of images;
2. Allow each Gabor wavelet element B_i to locally perturb.

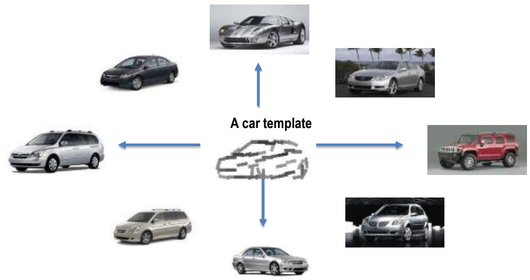
Deformable template using active basis

A car template

(Gabor elements represented by bar)

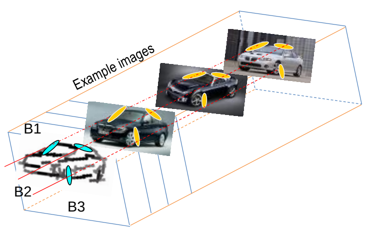
An incoming car image

Deformable template using active basis



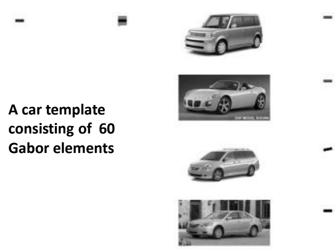
Deformed to fit many car instances

Learning the template: pursuing the active basis



Learning the template: pursuing the active basis

A car template consisting of 60 Gabor elements



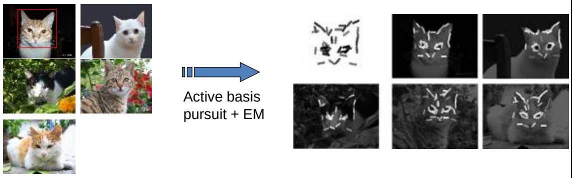
Car instances

Experimental results

Experiment 1: learning an active basis model of vehicle

- 37 training images, listed in the descending order of log-likelihood ratio
- 4.3 seconds (Core 2 Duo 2.4GHz) , after convolution

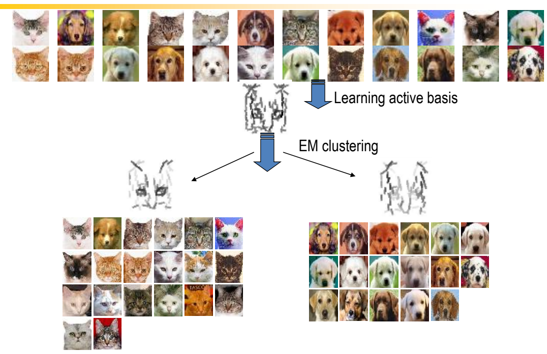
Experiment 2: learning without alignment



Given bounding box for the first example for initialization.

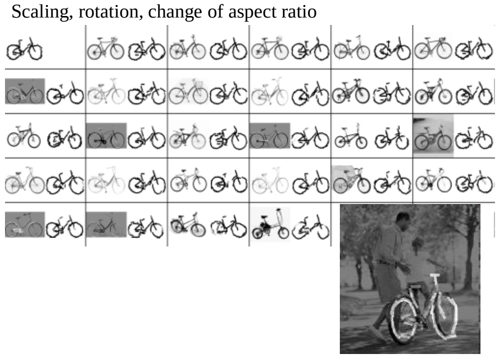
Iterate:
- Estimate the bounding boxes using current model.
- Re-learn the model from estimated bounding boxes.

Experiment 3: learning and clustering



Geometric transformation

Scaling, rotation, change of aspect ratio



Geometric transformation

Main contributions

1. An **active basis model** as deformable template.
2. An **active bases pursuit algorithm** for fast learning.
3. Robust for template matching

Homepage of this research:

http://www.stat.ucla.edu/~ywu/ActiveBasis.html

Download
1) Training and testing images
2) Matlab and mex-C source codes

RASL: Robust Alignment by Sparse and Low-rank Decomposition for Linearly Correlated Images

CVPR 2010, oral

Yigang Peng[1], Arvind Ganesh[2], John Wright[3], Wenli Xu[1] and Yi Ma [2,3]

1 TNLIST and Dept. of Automation, Tsinghua Univ.
2 Dept. of Electrical and Computer Engg., UIUC
3 Visual Computing Group, MSRA

Background

• Increasing data set from internet: Facebook, Flickr, You Tube…

Large amount

unconstraint

Difficult for object recognition or classification

Challenges for recognition algorithms

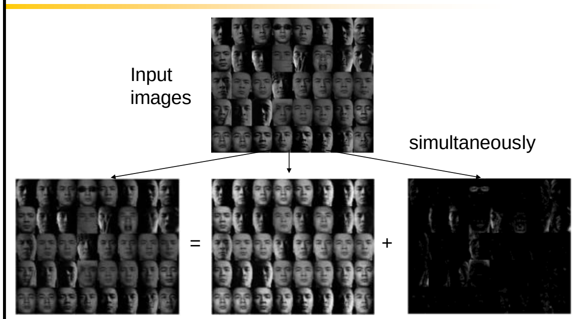
• illumination variation, partial occulusion, no alignment

Problem:

Can we recover the faces, despite corruptions or occlusions (shadow, sunglasses, hat, and scarves) ?

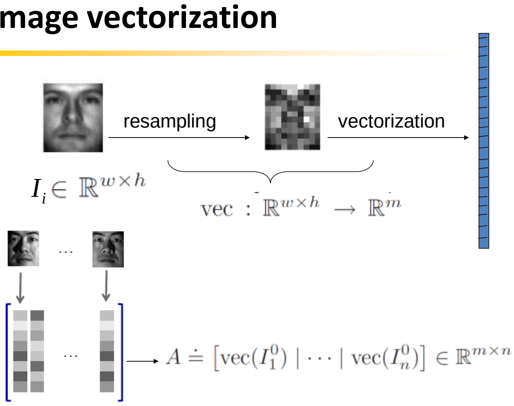
Difficult for measuring the image similarity

Idea



Method

Image vectorization

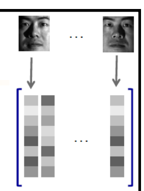


$I_i \in \mathbb{R}^{w\times h}$

$\mathrm{vec} : \mathbb{R}^{w\times h} \rightarrow \mathbb{R}^m$

$A \doteq [\mathrm{vec}(I_1^0) \mid \cdots \mid \mathrm{vec}(I_n^0)] \in \mathbb{R}^{m\times n}$

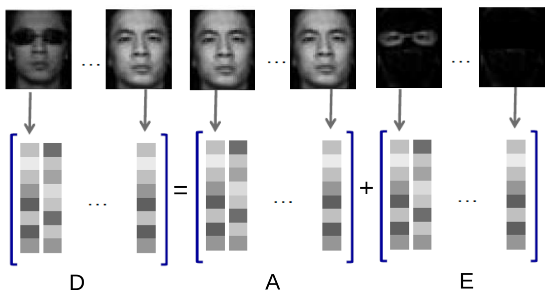
Assumption



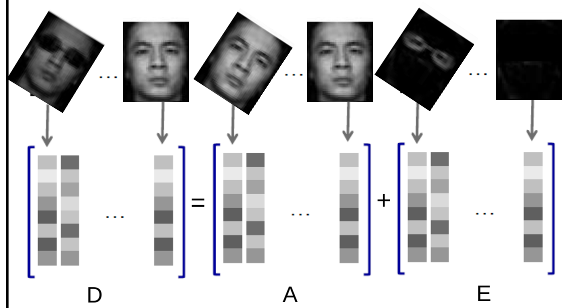
$A \doteq [\mathrm{vec}(I_1^0) \mid \cdots \mid \mathrm{vec}(I_n^0)] \in \mathbb{R}^{m\times n}$

If $I_1^0, \ldots, I_n^0$ are taken from same object, well-aligned and without corruption, they are linearly correlated. Then A should be low-rank.

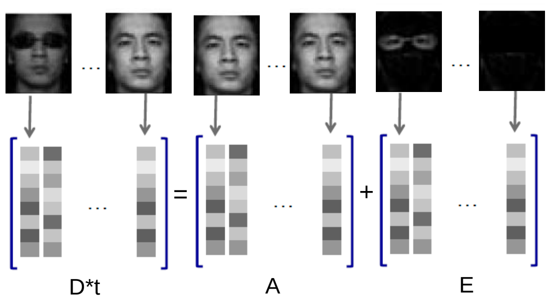
Challenge 1: corruption



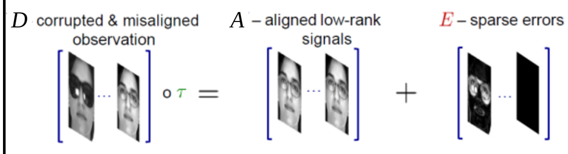
Challenge 2: misalignment



Challenge 2: misalignment



Problem definition

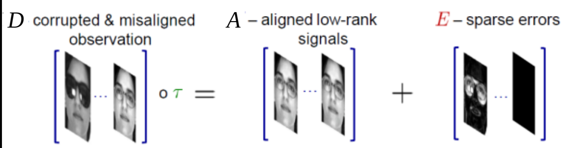


Problem: Given $D \circ \tau = A + E$, recover τ, A_0 and E_0.

Parametric deformations (rigid, affine, projective…) Low-rank component Sparse component

Solution: Robust Alignment via Low-rank and Sparse (**RASL**) Decomposition

Formulation



Problem: Given $D \circ \tau = A + E$, recover τ, A_0 and E_0.

Parametric deformations (rigid, affine, projective…) Low-rank component Sparse component

$$\min_{A,E,\tau} \ \mathrm{rank}(A) + \gamma \|E\|_0 \quad \text{s.t.} \quad D \circ \tau = A + E.$$

The number of nonzeros

The problem is N-P hard

$$\min_{A,E,\tau} \ \mathrm{rank}(A) + \gamma \|E\|_0 \quad \text{s.t.} \quad D \circ \tau = A + E.$$

$\gamma > 0$

The number of nonzeros

Optimization problem:

Both the low-rank and ℓ^0-norm minimizations are nonconvex and discontinuous!

N-P problem

Convex relaxation

$$\min_{A,E,\tau} \ \mathrm{rank}(A) + \gamma\|E\|_0 \quad \text{s.t.} \quad D \circ \tau = A + E.$$

Instead they optimize the problem as:

$$\min_{A,E,\tau} \ \|A\|_* + \lambda\|E\|_1 \quad \text{s.t.} \quad D \circ \tau = A + E$$

$$\|A\|_* \doteq \textstyle\sum_{i=1}^{m} \sigma_i(A) \qquad \sum_{ij} |E_{ij}|$$

Transformation linearization

The current problem:

$$\min_{A,E,\tau} \ \|A\|_* + \lambda\|E\|_1 \quad \text{s.t.} \quad D \circ \tau = A + E$$

The constraint is nonlinear, due to the dependence of the transformations.

Solution:

When the change in tau is small, we can approximate this constraint by linearizing about the current estimate of tau.

Transformation linearization

$$\min_{A,E,\tau} \ \|A\|_* + \lambda\|E\|_1 \quad \text{s.t.} \quad D \circ \tau = A + E$$

$$D \circ (\tau + \Delta\tau) \approx D \circ \tau + \textstyle\sum_{i=1}^{n} J_i \Delta\tau_i \epsilon_i^T$$

$$J_i \doteq \frac{\partial}{\partial \zeta} \mathrm{vec}(I_i \circ \zeta)|_{\zeta=\tau_i} \in \mathbb{R}^{m \times p}$$

Transformation linearization

$$J_i = \begin{bmatrix} \frac{\partial I_{i1}}{\partial \tau_i^1} & \frac{\partial I_{i1}}{\partial \tau_i^2} & \frac{\partial I_{i1}}{\partial \tau_i^p} \\ \frac{\partial I_{i2}}{\partial \tau_i^1} & \frac{\partial I_{i2}}{\partial \tau_i^2} & \frac{\partial I_{i2}}{\partial \tau_i^p} \\ & & \\ \frac{\partial I_{im}}{\partial \tau_i^1} & \frac{\partial I_{im}}{\partial \tau_i^2} & \frac{\partial I_{im}}{\partial \tau_i^p} \end{bmatrix} \in \mathbb{R}^{m \times p}$$

$\Delta\tau_i \in R^{p\times 1}$ → Difference of transformation parameter

$\varepsilon_i \in R^{n\times 1}$ → The *i*-th Standard basis

Transformation linearization

$$\min_{A,E,\tau} \ \|A\|_* + \lambda\|E\|_1 \quad \text{s.t.}$$

$$D \circ \tau + \textstyle\sum_{i=1}^{n} J_i \Delta\tau_i \epsilon_i^T = A + E$$

Solvable problem: by convex programming

Algorithm 1 (Outer loop of RASL)

INPUT: Images $I_1, \ldots, I_n \in \mathbb{R}^{w \times h}$, initial transformations $\tau_1, \ldots, \tau_n$ in certain parametric group $\mathbb{G}$, weight $\lambda > 0$.

WHILE not converged **DO**

step 1: compute Jacobian matrices w.r.t. transformation:

$$J_i \leftarrow \frac{\partial}{\partial \zeta} \left(\frac{\mathrm{vec}(I_i \circ \zeta)}{\|\mathrm{vec}(I_i \circ \zeta)\|_2} \right) \Big|_{\zeta=\tau_i}, \quad i = 1, \ldots, n;$$

step 2: warp and normalize the images: transform images

$$D \circ \tau \leftarrow \left[\frac{\mathrm{vec}(I_1 \circ \tau_1)}{\|\mathrm{vec}(I_1 \circ \tau_1)\|_2} \Big| \ldots \Big| \frac{\mathrm{vec}(I_n \circ \tau_n)}{\|\mathrm{vec}(I_n \circ \tau_n)\|_2} \right];$$

step 3: solve the linearized convex optimization:

$$(A^*, E^*, \Delta\tau^*) \leftarrow \arg\min_{A,E,\Delta\tau} \ \|A\|_* + \lambda\|E\|_1$$

$$\text{s.t. } D \circ \tau + \sum_{i=1}^{n} J_i \Delta\tau_i \epsilon_i^T = A + E;$$

step 4: update transformations: $\tau \leftarrow \tau + \Delta\tau^*$;

END WHILE

OUTPUT: solution A^*, E^*, τ to problem (5).

Next problem

$\tau_1, \ldots, \tau_n$ in certain parametric group $\mathbb{G}$, weight $\lambda > 0$.

WHILE not converged **DO**

step 1: compute Jacobian matrices w.r.t. transformation:

$$J_i \leftarrow \frac{\partial}{\partial \zeta}\left(\frac{\mathrm{vec}(I_i \circ \zeta)}{\|\mathrm{vec}(I_i \circ \zeta)\|_2}\right)\Big|_{\zeta=\tau_i}, \quad i = 1, \ldots, n;$$

step 2: warp and normalize the images:

$$D \circ \tau \leftarrow \left[\frac{\mathrm{vec}(I_1 \circ \tau_1)}{\|\mathrm{vec}(I_1 \circ \tau_1)\|_2} \Big| \cdots \Big| \frac{\mathrm{vec}(I_n \circ \tau_n)}{\|\mathrm{vec}(I_n \circ \tau_n)\|_2}\right];$$

step 3: solve the linearized convex optimization:

$$(A^*, E^*, \Delta\tau^*) \leftarrow \arg\min_{A,E,\Delta\tau} \|A\|_* + \lambda\|E\|_1$$

$$\text{s.t. } D \circ \tau + \sum_{i=1}^{n} J_i \Delta\tau_i \epsilon_i^T = A + E;$$

- **Semi-definite program**
- **Thousands or millions of variables**

step 4: update transformations: $\tau \leftarrow \tau + \Delta\tau^*$;

END WHILE

OUTPUT: solution A^*, E^*, τ to problem (5).

Using APG (accelerated proximal gradient) algorithm [2,22,17]

$$\min_{A,E,\Delta\tau} \|A\|_* + \lambda\|E\|_1 \ \text{ s.t. } \ D\circ\tau + \sum_{i=1}^{n} J_i \Delta\tau_i \epsilon_i^T = A + E$$

$$f(A, E, \Delta\tau) \doteq \frac{1}{2}\|A + E - D\circ\tau - \textstyle\sum_{i=1}^{n} J_i \Delta\tau_i \epsilon_i^T\|_F^2$$

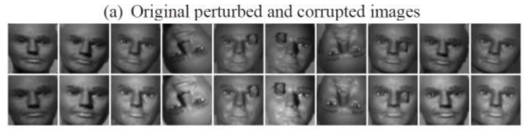
$$\min_{A,E,\Delta\tau} \|A\|_* + \lambda\|E\|_1 + \frac{1}{\mu} f(A, E, \Delta\tau)$$

Fast solved by Accelerated Proximal Gradient (APG) [1,22,17]

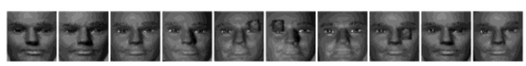
Experimental results

Result 1 synthesized image

(a) Original perturbed and corrupted images



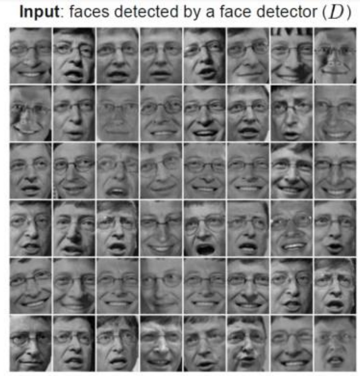
(b) Alignment results by [23] (Top: direct; Bottom: gradient)



(c) Alignment results by RASL

Nature image

Input: faces detected by a face detector (D)



Average

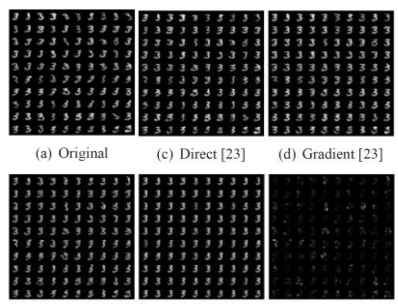
Output: aligned faces ($D \circ \tau$)

Output: clean low-rank faces (A)

Average image

(a) Average faces from face detector (b) Average faces after alignment

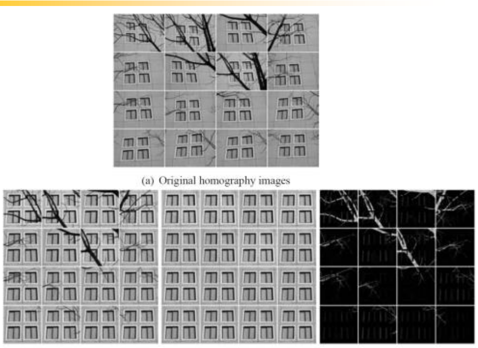
Aligning handwritten digits



(a) Original (c) Direct [23] (d) Gradient [23]

(e) $D \circ \tau$ (RASL) (f) A (RASL) (g) E (RASL)

Aligning planar surfaces



(a) Original homography images

(b) Aligned images $D \circ \tau$ (c) Reconstructed images A (d) Removed occlusions E

Contributions

- Robustness to corruption and occlusion
- Robustness to misalignment

Homepage of this research:

http://perception.csl.illinois.edu/matrix-rank/rasl.html

Download

1) Papers
2) Sample code in Matlab

Readings

1. David G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp. 91-110
2. H. Bay, T. Tuytelaars, L. V. Gool, "SURF: Speeded Up Robust Features," ECCV 2006
3. S. Gu, Y. Zheng and C. Tomasi, "Critical Nets and Beta-Stable Features for Image Matching," ECCV2010
4. Y. N. Wu, Z.Z. Si, H.f. Gong and S.-C. Zhu: Learning Active Basis Model for Object Detection and Recognition. International Journal of Computer Vision 90(2): 198-235 (2010)
5. Y.G. Peng, A. Ganesh, J. Wright, W. Xu and Y. Ma, "RASL: Robust Alignment via Sparse and Low-Rank Decomposition for Linearly Correlated Images," CVPR 2010.
6. B. Zheng, Y.-Q. Sun, J. Takamatsu, and K. Ikeuchi, “A Feature Descriptor by Difference of Polynomials”, IPSJ Trans. on Computer Vision and Applications (CVA), Vol.5, 80-84, 2013.