Shape Features

WangRuchen

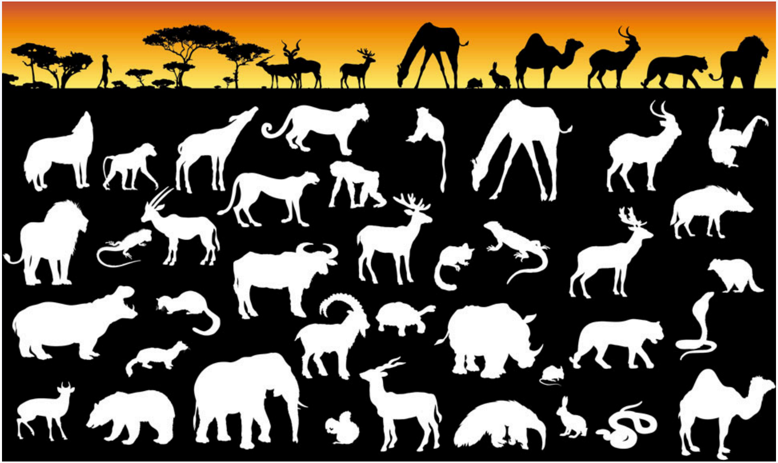
CVBIOUC
http://vision.ouc.edu.cn/~zhenghaiyong

June 11, 2015

Contents

1.1 Why study shape?

Humans can recognize many objects based on shape alone.



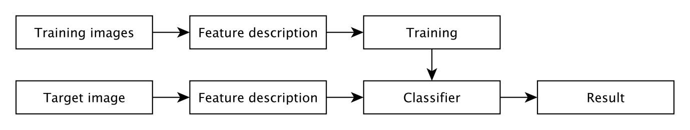
1.2 Application

- Image retrieval
- Character recognition
- Object detection
- ...

1.3 Matching & Recognition

1.3 Matching & Recognition

Shape recognition:

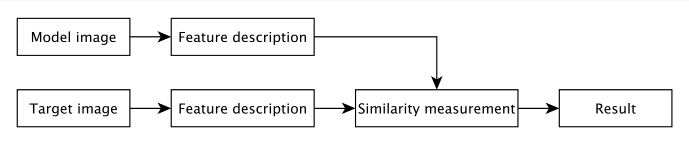


1.3 Matching & Recognition

1.3 Matching & Recognition

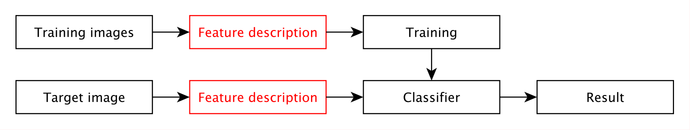
1.3 Matching & Recognition

Shape matching:

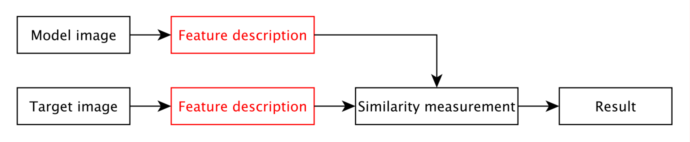


1.3 Matching & Recognition

Shape recognition:



Shape matching:



2 Shape description

Shape can be described in two different ways:

- Contour-based
  This method is connected to edge and line detection.
- Region-based
  This method is linked to the region segmentation.

2.1 Shape description techniques[1][2][3]

- Chain code
- Polygonal approximation
- B-spline
- Wavelet descriptor
- Fourier descriptor
- Curvature scale space
- Shape context

...

- Skeletons
- Convex hull
- Geometric moments
- Zernike moments
- Shape matrix
- Core

...

[1] Mingqiang Yang, Kidiyo Kpalma, Joseph Ronsin. "A Survey of Shape Feature Extraction Technique", PR, 2010.

[2] Dengsheng Zhang, Guojun Lu. "Review of shape representation and description techniques", PR, 2003.

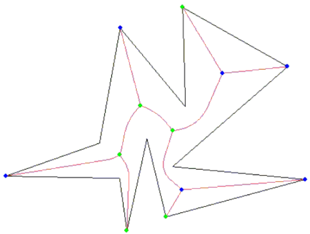
[3] Yu Zhou, Juntao Liu, Xiang Bai. "Research and Perspective on Shape Matching", Acta Automatica Sinica, 2012.

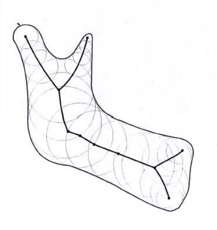
2.1 Shape description techniques

- Skeleton
- Convex hull
- Shape context

2.2 Skeleton (Medial axis)

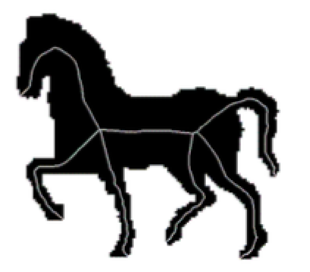
Skeleton is defined by Grassfire Model[4]. It is also defined as the locus of centers of maximal disks that fit within the shape.





[4]Harry Blum. "Biological shape and visual science", Theoretical Biology, 1973.

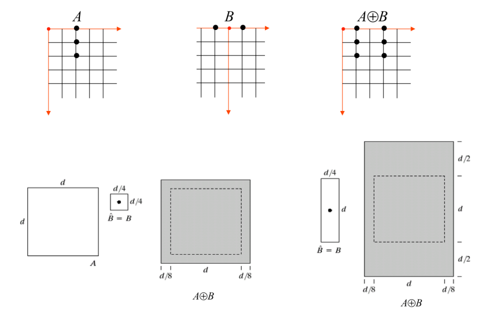
2.2.1 Skeleton extraction (Medial axis transform)



- Voronoi diagram
- Distance transform
- Mathematical morphology

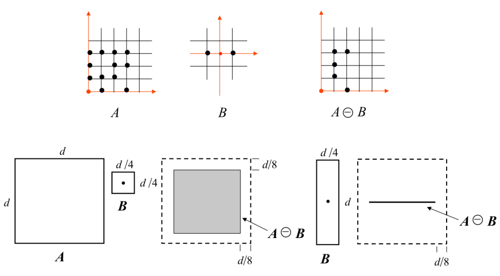
- Dilation

$$A \oplus B = \{z | (B)_z \cap A \neq \emptyset\}$$



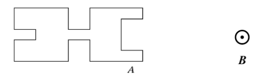
- Erosion

$$A \odot B = \{z | (B)_z \subseteq A\}$$



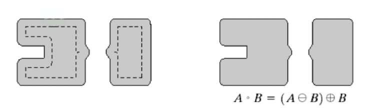
2.2.1 Skeleton extraction

Mathematical morphology



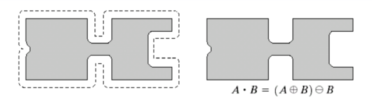
- Opening

$$A \circ B = (A \odot B) \oplus B$$



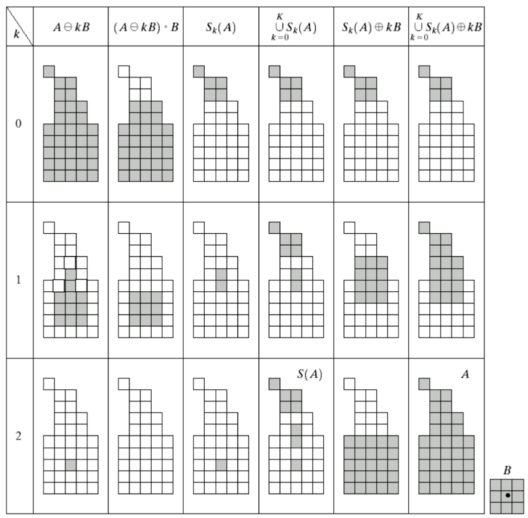
- Closing

$$A \bullet B = (A \oplus B) \odot B$$



2.2.1 Skeleton extraction

Skeletonization via Mathematical morphology



2.2.1 Skeleton extraction

Skeletonization via Mathematical morphology

The skeleton of a set can be expressed in terms of erosions and openings:

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B$$

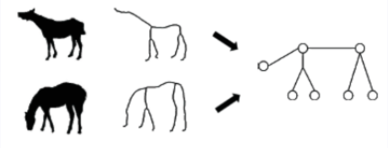
$$S(A) = \bigcup_{k=0}^{K} S_{k(A)}$$

- B - is a sturcturing element.
- K - is the last iterative step before A erodes to an empty set.

A can be reconstructed from its skeleton subsets $S_k(A)$ using the equation:

$$A = \bigcup_{k=0}^{K} (S_{k(A)} \oplus kB)$$

2.2.2 Attributed relation graphs (ARG)



- Shock graph[5]
- Skeleton tree[6]
- Bone graph[7]

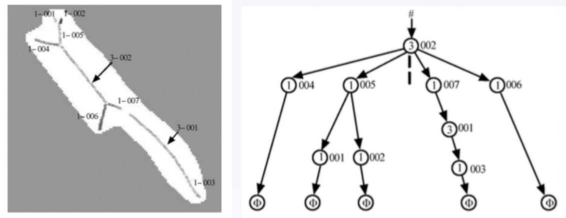
---

[5] Kaleem Siddiqi *et al.*. "Shock graph and shape matching", IJCV, 1999.

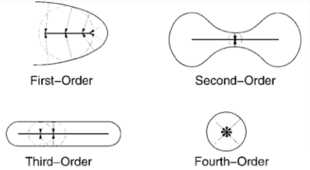
[6] Wenyu Liu, Juntao Liu. "Objects similarity measure based on skeleton tree descriptor matching", Journal Infrared Millimeter and Wave, 2005.

[7] Diego Macrini, Kaleem Siddiqi *et al.*. "From skeletons to bone graphs: medial abstraction for object recognition", CVPR, 2008.

What is shock graph?



Shock points:

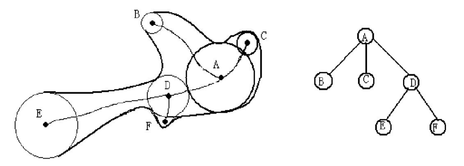


[8] K Siddiqi. “Shock graph and shape matching”, IJCV, 1999.

2.2.2 Attributed relation graphs (ARG)

Skeleton tree[9]

What is skeleton tree?



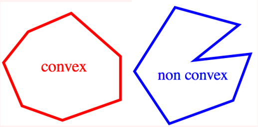
Tree descriptor: $(3, 0, 0, 2, 0, 0)$

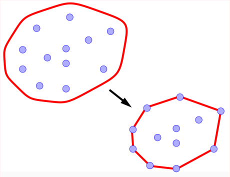
[9] Wenyu Liu, Juntao Liu. "Objects similarity measure based on skeleton tree descriptor matching", Journal Infrared Millimeter and Wave, 2005.

2.3 Convex hull

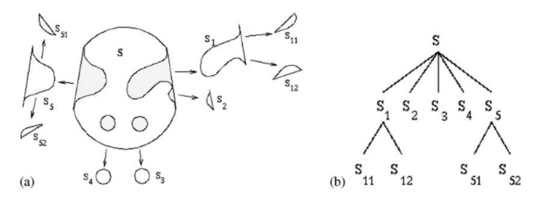
What?

The convex hull of a region is the smallest convex polygon that contains all the points of the region.



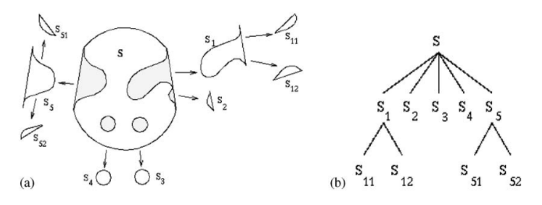


How to describe shape?



2.3 Convex hull

How to describe shape?

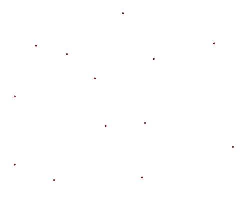


Convex hull algorithms:

- Graham Scan Algorithm
- Quickhull Algorithm

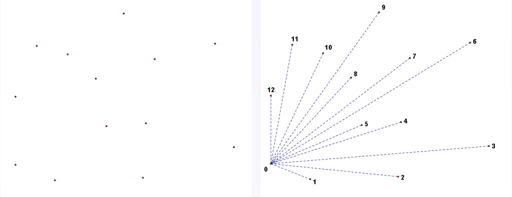
2.3 Convex hull

- Graham Scan Algorithm[10]



[10] Ronald L. Graham. "An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set", Information Processing Letters, 1972.

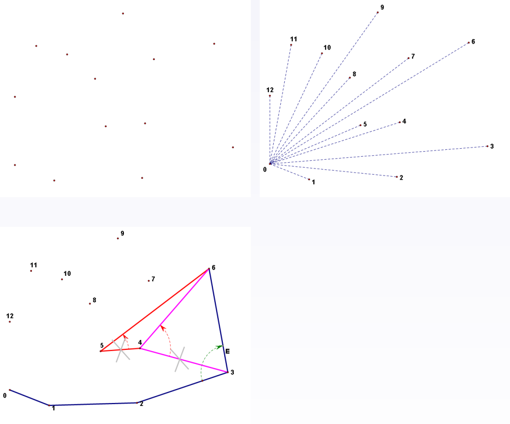
- Graham Scan Algorithm[10]



[10] Ronald L. Graham. "An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set", Information Processing Letters, 1972.

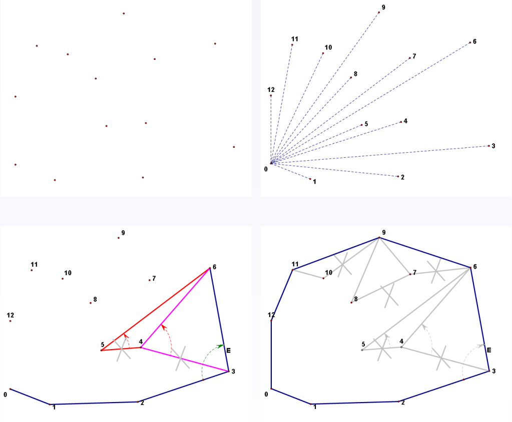
2.3 Convex hull

- Graham Scan Algorithm[10]



[10] Ronald L. Graham. “An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set”, Information Processing Letters, 1972.

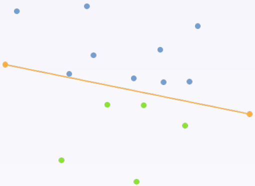
- Graham Scan Algorithm[10]



[10] Ronald L. Graham. "An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set", Information Processing Letters, 1972.

2.3 Convex hull

- Quickhull Algorithm[11]



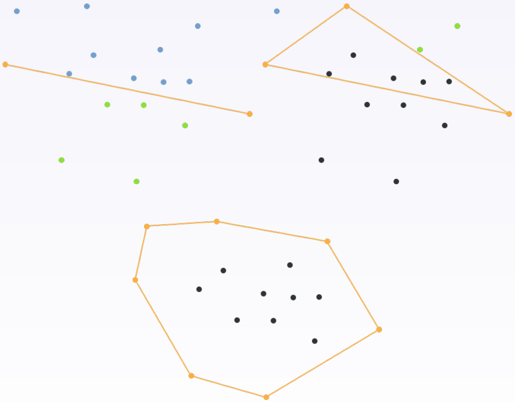
[11] Barber C. Bradford *et al.*. "The quickhull algorithm for convex hulls", ACM Transactions on Mathematical Software, 1996.

2.3 Convex hull

- Quickhull Algorithm[11]

[11] Barber C. Bradford *et al.*. "The quickhull algorithm for convex hulls", ACM Transactions on Mathematical Software, 1996.

- Quickhull Algorithm[11]



[11] Barber C. Bradford *et al.*. "The quickhull algorithm for convex hulls", ACM Transactions on Mathematical Software, 1996.

2.4 Shape context[12]

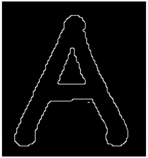
The shape context is intended to be a way of describing shapes that allows for measuring shape similarity and the recovering of point correspondences.

How to describe a shape?

[12] Serge Belongie *et al.*. "Shape Matching and Object Recognition Using Shape Contexts", PAMI, 2002.

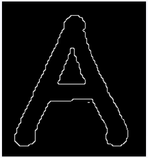
2.4.1 How to describe a shape?

1. Obtain contours using edge detector

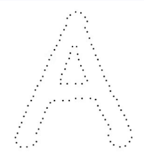


2.4.1 How to describe a shape?

1. Obtain contours using edge detector

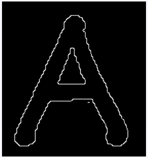


2. Pick n points on the contours of a shape

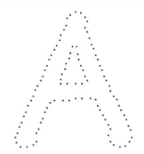


2.4.1 How to describe a shape?

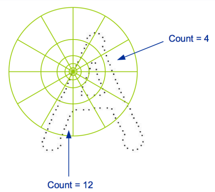
1. Obtain contours using edge detector

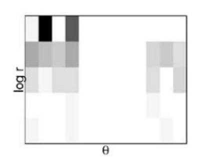


2. Pick n points on the contours of a shape



3. Compute the shape context of each point





2.4.2 Matching with shape contexts

- Match each point from the known shape to a point on an unknown shape

2.4.2 Matching with shape contexts

1. Computing the cost matrix

$$C_{ij} = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}$$

$C_{ij} = C(p_i, q_j)$ denote the cost of matching these two points.

Cost matrix:

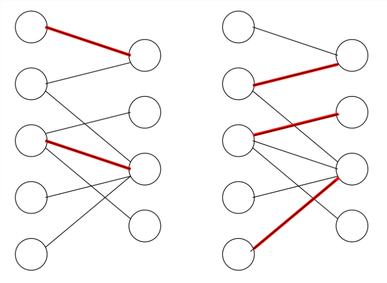
$$\begin{pmatrix} C_{11} & C_{12} & \dots & C_{1n} \\ C_{21} & C_{22} & \dots & C_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ C_{n1} & C_{n2} & \dots & C_{nn} \end{pmatrix}$$

2. Finding the matching that minimizes total cost:

$$H(\pi) = \sum_i C(p_i, q_{\pi(i)})$$

Hungarian Algorithm is used in optimizing the assignment problems.

- Bipartite graph



[13] Harold W. Kuhn. "The Hungarian Method for the assignment problem", Naval Research Logistics Quarterly, 1955.

2.4.2 Matching with shape contexts

Hungary algorithm

- Matrix interpretation

Theorem

If a number is added to or subtracted from all of the entries of any one row or column of a cost matrix, then on optimal assignment for the resulting cost matrix is also an optimal assignment for the original cost matrix.

Step 1. Subtract the smallest entry in each row from all the entries of its column.

$$\begin{bmatrix} 250 & 400 & 350 \\ 400 & 600 & 350 \\ 200 & 400 & 250 \end{bmatrix} \sim \begin{bmatrix} 0 & 150 & 100 \\ 50 & 250 & 0 \\ 0 & 200 & 50 \end{bmatrix}$$

Step 2. Subtract the smallest entry in each column from all the entries of its column.

$$\begin{bmatrix} 0 & 150 & 100 \\ 50 & 250 & 0 \\ 0 & 200 & 50 \end{bmatrix} \sim \begin{bmatrix} 0 & 0 & 100 \\ 50 & 100 & 0 \\ 0 & 50 & 50 \end{bmatrix}$$

Step 3. Draw lines through appropriate rows and columns so that all the zero entries of the cost matrix are covered and the minimum number of such lines is used.

$$\begin{bmatrix} 0 & 0 & 100 \\ 50 & 100 & 0 \\ 0 & 50 & 50 \end{bmatrix}$$

Step 4. (i)If the minimum number of covering lines is n, an optimal assignment of zeros is possible and we are finished. (ii) If the minimum number of covering lines is less than n, an optimal assignment of zeros is not yet possible. In that case, proceed to Step 5

Step 5. Determine the smallest entry not covered by any line. Subtract this entry from each uncovered row, and then add it to each covered column. Return to Step 3.

$$\begin{bmatrix} 0 & \boxed{0} & 100 \\ 50 & 100 & \boxed{0} \\ \boxed{0} & 50 & 50 \end{bmatrix} \quad \begin{bmatrix} 250 & \boxed{400} & 350 \\ 400 & 600 & \boxed{350} \\ \boxed{200} & 400 & 250 \end{bmatrix}$$

2.4.2 Matching with shape contexts

Finding the matching that minimizes total cost:

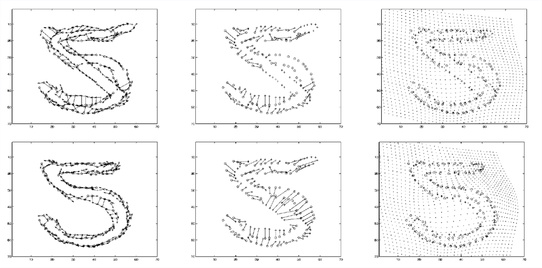
$$H(\pi) = \sum_i C(p_i, q_{\pi(i)})$$

2.4.2 Matching with shape contexts

S. Belongie et al.. "Shape Matching and Object Recognition Using Shape Contexts", PAMI, 2002

Thin Plate Spline (TPS)

- TPS are a spline-based technique for data interpolation and smoothing.
- TPS has been widely used as the non-rigid transformation model in shape matching.



2.4.2 Matching with shape contexts

The shape distance is going to be a weighted sum of three potential terms: shape context distance, image appearance distance, and bending energy.

- Shape context distance

$$D_{SC}(P, Q) = \frac{1}{n} \sum_{p \in P} arg \min_{q \in Q} C(p, T(q)) + \frac{1}{m} \sum_{q \in Q} arg \min_{p \in P} C(p, T(q))$$

- Appearance cost
- Transformation cost

2.4.2 Matching with shape contexts

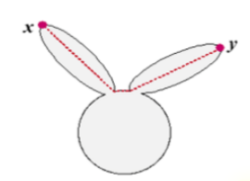
Algorithm

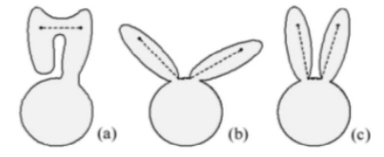
1. Finding a list of points on shape edges
2. Computing the shape context
3. Computing the cost matrix
4. Finding the matching that minimizes total cost
5. Modeling transformation
6. Computing the shape distance

Disadvantage: Can't address the deformation of the same object.

2.4.3 Inner-distance shape context[14](Demo)

Replace euclidean distance with the inner-distance.

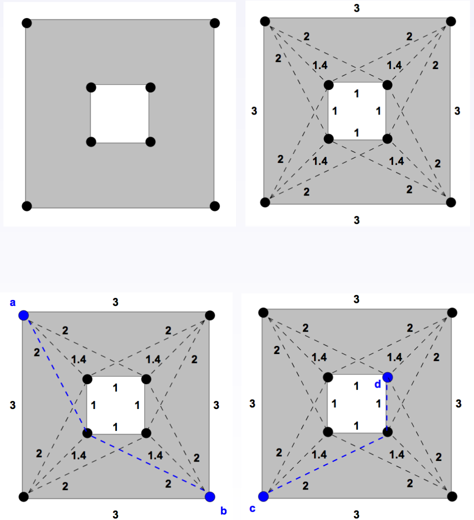




- Insensitive to shape articulations
- Often more discriminative for complex shapes

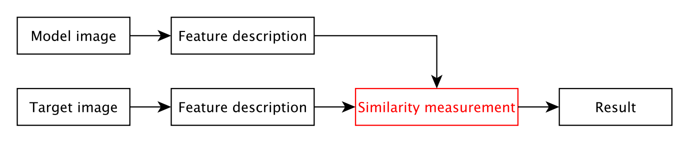
[14] Haibin Ling, David W. Jacobs. "Shape Classification Using the Inner-Distance", PAMI, 2007.

2.4.3 Inner-distance shape context



3 Similarity calculation

Shape matching:



3.1 Similarity calculation method

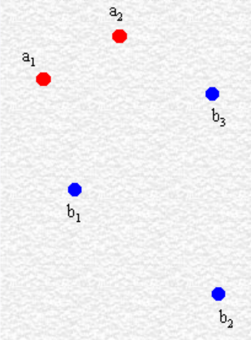
Similarity calculation is used to measure the difference (distance) or similarity between different objects.

- Euclidean Distance
- Manhattan Distance
- Mahalanobis Distance
- Minkowski Distance
- Hausdorff distance

- Cosine Similarity
- Tanimoto Coefficient
- Pearson correlation coefficient

3.2 Hausdorff distance[15]

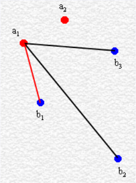
Hausdorff distance measures how far two subsets of a metric space are from each other.

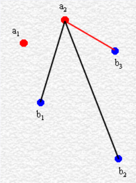


$$d_H(A, B) = \max[h(A, B), h(B, A)]$$

[15] Daniel P. Huttenlocher, Gregory A. Klanderman. "Comparing images using the Hausdorff", PAMI, 1993.

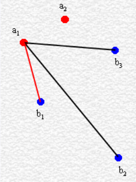
3.2 Hausdorff distance

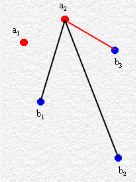




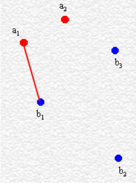
$$d_B(a) = \min_{b \in B} \| a - b \|$$

3.2 Hausdorff distance



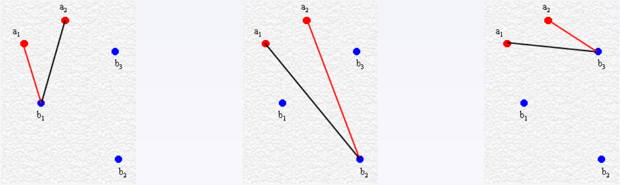


$$d_B(a) = \min_{b \in B} \| a - b \|$$

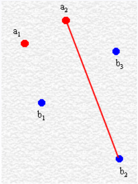


$$h(A, B) = \max_{a \in A} d_B(a)$$

3.2 Hausdorff distance

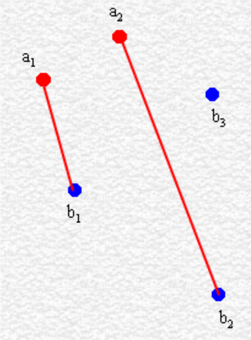


$$d_A(b) = \min_{a \in A} \| a - b \|$$



$$h(B, A) = \max_{b \in B} d_A(b)$$

3.2 Hausdorff distance



$$d_H(A, B) = \max[h(A, B), h(B, A)]$$

3.2 Hausdorff distance

Given two finite point sets A and B , the Hausdorff distance $d_H(A, B)$ is defined as:

$$d_H(A,B) = \max[h(A,B), h(B,A)]$$

$$h(A,B) = \max_{a \in A} d_B(a) \qquad d_B(a) = \min_{b \in B} \| a - b \|$$

- $h(A, B)$ - the directed Hausdorff distance from A to B.

Partial hausdorff distance:

$$H_K(A,B) = \max[h_K(A,B), h_K(B,A)]$$

$$h_K(A,B) = K^{th}_{a\in A} \min_{b\in B} \| a - b \|$$

- $K^{th}_{a\in A}$ - the Kth ranked value in the set of distance.

Partial hausdorff distance can overcome cover and external point exists.

[16] Daniel P. Huttenlocher, Gregory A. Klanderman. “Comparing images using the Hausdorff”, PAMI, 1993.

3.2 Hausdorff distance

Modified Hausdorff Distance (MHD)[17]

Modified Hausdorff Distance:

$$H_K(A, B) = \max[h_{MHD}(A, B), h_{MHD}(B, A)]$$

$$h_{MHD}(A, B) = \frac{1}{N_A} \sum_{a \in A} d_B(a)$$

[17] MP Dubuisson. “A Modified Hausdorff Distance for Object Matching”, PR, 1994.

Thanks!