Utrecht University

Shape Matching
Part 3: Direct Shape

Remco Veltkamp

Center for Geometry, Imaging, and Virtual Environments (GIVE)

Utrecht University

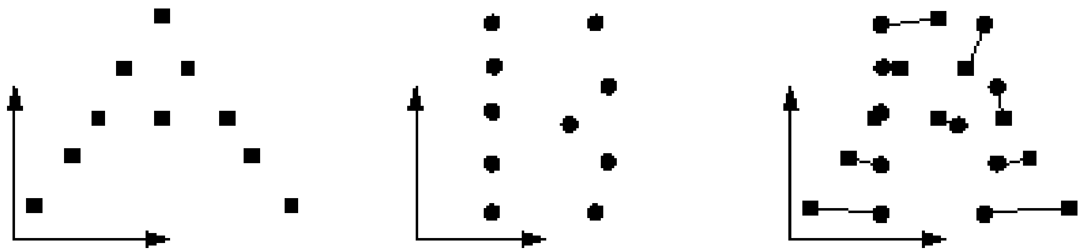
www.cs.uu.nl/centers/give

Geometric Pattern Matching

- point set - point set
- curve - curve
- region - region

Point Sets

- point set $A=\{a_1, \ldots, a_n\}$, query/data
- point set $B=\{b_1, \ldots, b_m\}$, model/target
- one-to-one or many-to-many
- graph theoretical approach:
  bipartite graph (V,E), $V=A\cup B$, E weighted



- geometric approach more efficient:
  "Geometry Helps in Matching" [Vaidya 89]

Bottleneck Distance 1

- 1-1 matching of sets A,B of n points
- min of max distance d(a,b) over all 1-1 matchings (a,b)
- using parametric search: $O(n^{1.5}\log n)$
- approximate matching $d(A,B)<(1+\varepsilon)d^*(A,B)$ using approximate nearest neighbors $O(n^{1.5}\log n)$

Bottleneck Distance 2

- decision problem for translation:
  decide whether t exists such that $d(A+t,B)< \varepsilon$
  $O(n^5 \log n)$
- approximation: compute t such that
  $d(A+t,B)<(1+ \varepsilon)d(A+t^*,B)$
  $O(n^{2.5} \log n)$
- optimization for rigid motion: $O(n^6 \log n)$
- optimization for translations: $O(n^5 \log n)$

Minimum Weight

- minimum total sum of the distances d(a,b) over all 1-1 matches (a,b)
- computation: $O(n^{2+\varepsilon})$
  but for L_∞: $O(n^2 \log^3 n)$

Most Uniform

- most uniform/balanced/fair
- minimum difference between maximum and minimum d(a,b) over all 1-1 matches (a,b)
- $O(n^{10/3}\log n)$

Minimum Deviation

- minimum difference between maximum and average d(a,b) over all 1-1 matches (a,b)
- $O(n^{10/3} \log n)$

Voting Schemes

- alignment [Huttenlocher 87]
- generalized Hough transform, pose clustering [Balard 81, Stockman 87]
- geometric hasing [Lamdan & Wolfson 88]
- query point set $A=\{a_1, \ldots, a_n\}$
- model point set $B_i=\{b_1, \ldots, b_m\}, i=1,\ldots N$

Alignment

- for each triplet of points in query set
  for each triplet from model set
  compute trafo between them
- with this trafo, transform all other points from model set
- if they match with query points, trafo receives a vote
- if number of votes is above threshold, trafo is accepted
- time complexity for matching: $O(Nm^4n^3)$

Pose Clustering

point sets
n-m

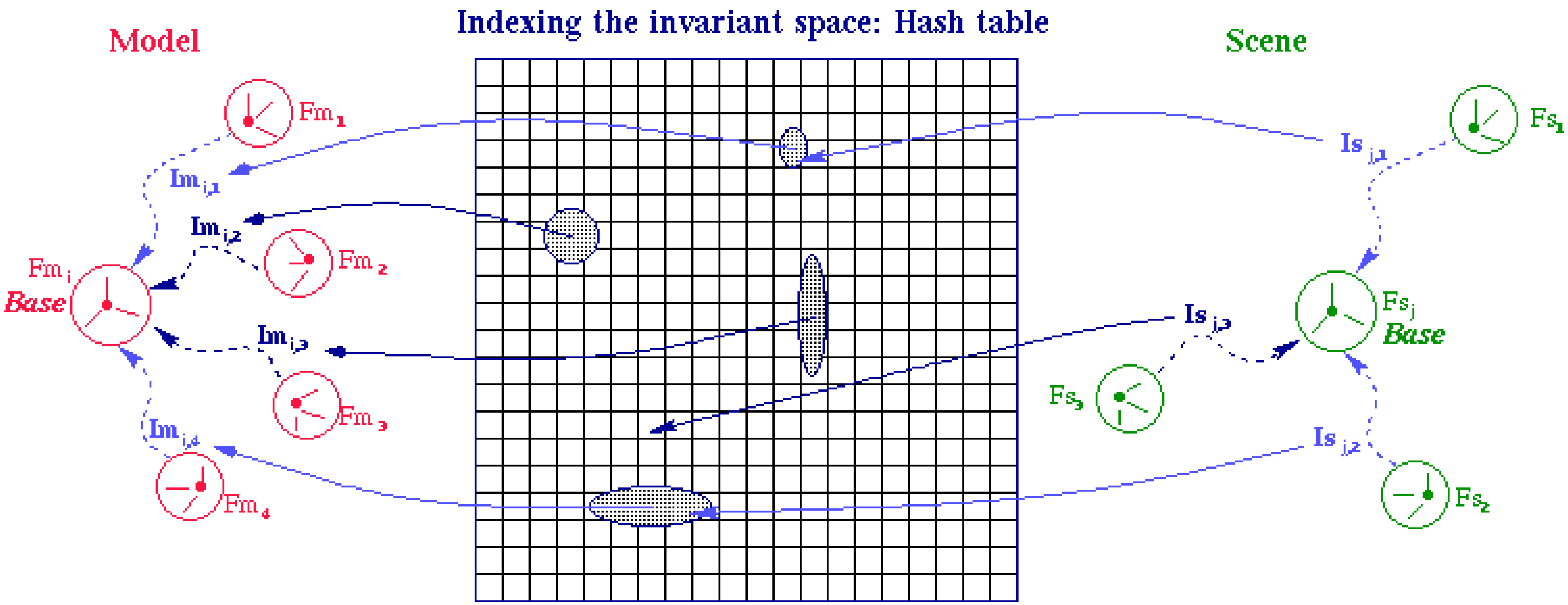
- affine trafo represented by 6 coefficients
- quantize trafo space into 6D table
- for all model point sets
  for each triplet of points in query set
  for each triplet of points model set
  compute trafo between two triplets
  tally a vote in corresponding entry
- highest score gives trafo between query and model
- time complexity for matching: $O(Nm^3n^3)$

Geometric Hashing 1

- choose 3 points e_0,e_1,e_2 from B_i
- $b_j=e_0+\alpha(e_1-e_0)+\beta(e_2-e_0)$
- quantize (α ,β)-plane into 2D table
- each (α ,β) is mapped to an index (u,v)
- for each 3 non-collinear points e_0,e_1,e_2,
  for each $b_i=e_0+\alpha(e_1-e_0)+\beta(e_2-e_0)$,
  append (i, e_0,e_1,e_2) to entry (u,v)
- time complexity for N models: $O(Nm^4)$

Geometric Hashing 1.5

point sets
n-m



Geometric Hashing 2

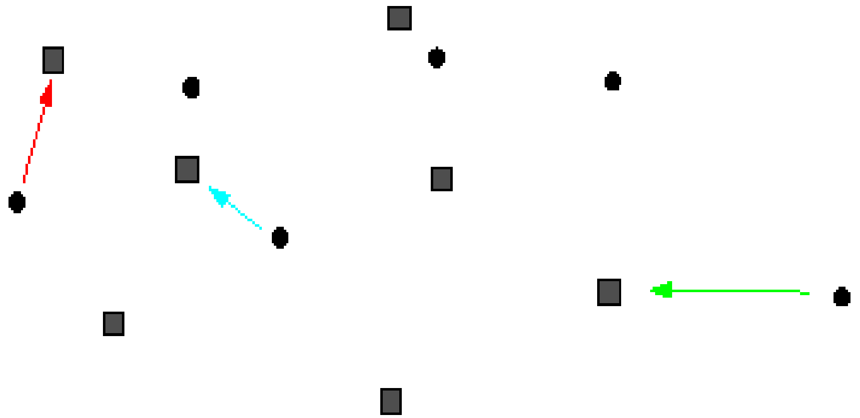
- choose 3 points e'_0, e'_1, e'_2 from A
- for each $a_i = e_0 + \alpha(e'_1 - e'_0) + \beta(e'_2 - e'_0)$, tally a vote for each (i, e_0, e_1, e_2) in entry (u,v)
- winner (i, e_0, e_1, e_2) indicates that point set B_i contains query set A
- affine trafo between (e'_0, e'_1, e'_2) and winner (e_0, e_1, e_2)
- time complexity for matching: $O(n)$

Geometric Hashing 3

- robust against occlusion of models : yes
- robust against noise in query : no
- variations:
  - balance the hashing table
  - avoid taking all $O(m^3)$ tuples

Hausdorff Distance 1

- $d(a,B)=\min_{b\in B} d(a,b)$
- directed Hausdorff distance: $h(A,B)=\max_{a\in A} d(a,B)$
- Hausdorff distance: $H(A,B)=\max\{h(A,B),h(B,A)\}$



Hausdorff Distance 2

- computation in time $O((n+m)\log(n+m))$ using Voronoi diagrams
- optimization under translation: $O(mn(m+n)\alpha(mn)\log(m+n))$
  with $\alpha()$ the inverse Ackermann function, a very slowly increasing function
- optimization under rigid motion: $O((m+n)^6\log(mn))$

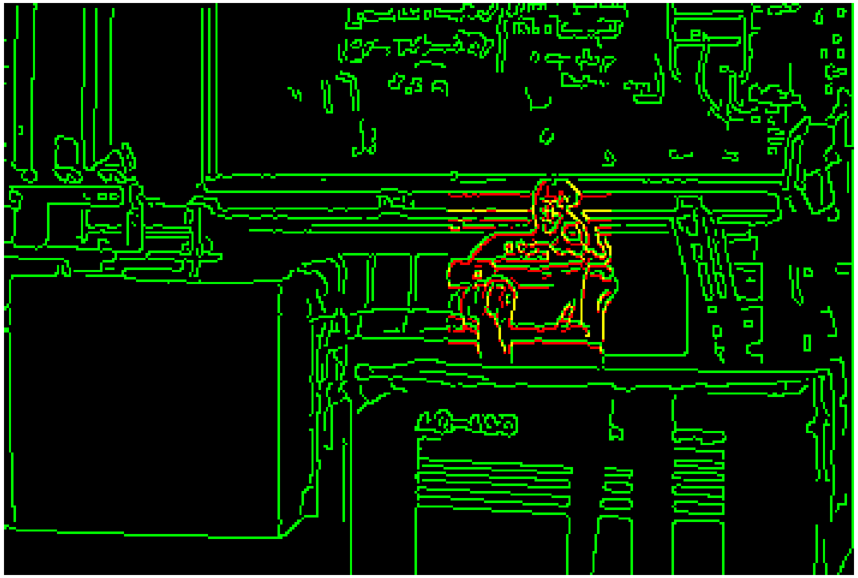
Hausdorff Distance 2

- partial directed Hausdorff distance:
  $h_k(A,B)=k^{th}{}_{a\in A}d(a,B)$
- not a metric

Hausdorff Distance 3

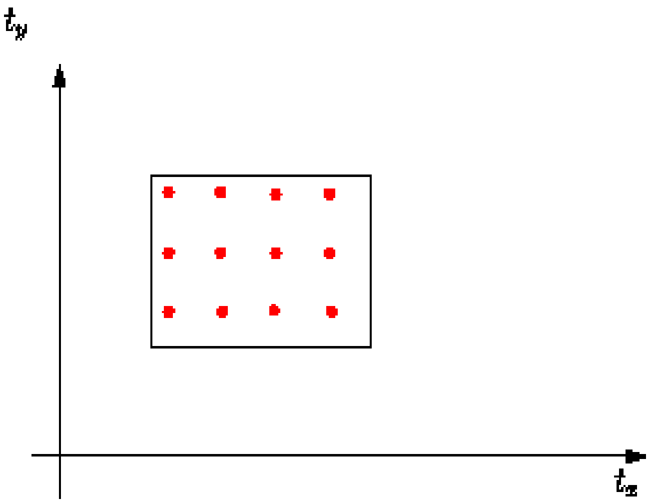
- application (Huttenlocher)
- model points:

- data points+match:



Template Matching

- because of high complexity of Hausdorff distance optimization problem under translation or rigid motions, probe transformation space:



- variations:
  - hierarchical template matching
  - adaptive template matching

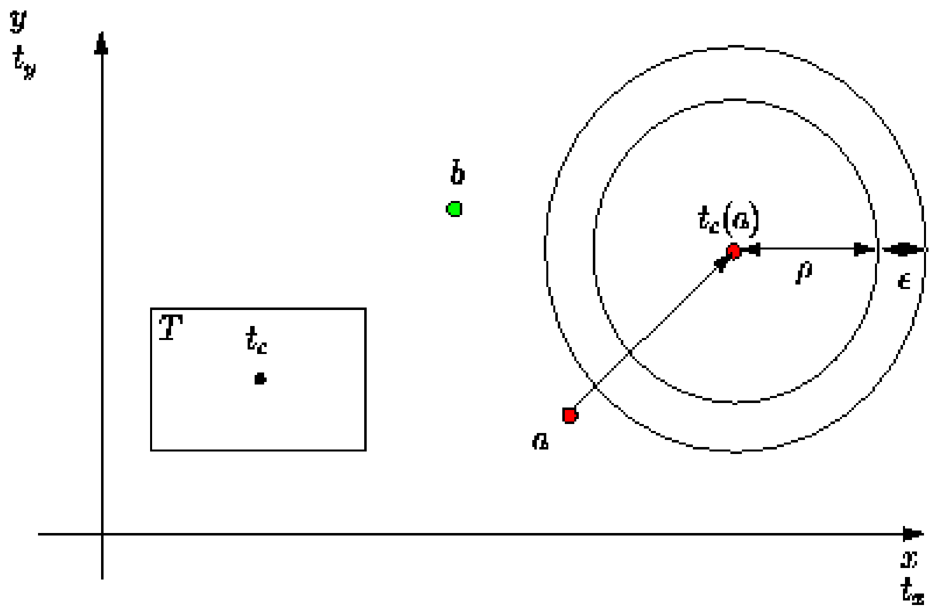
Which algorithm?

class of algorithms:

- subdivision schemes
  - decision problem, translation+scaling (Huttenlocher, Rucklidge)
  - optimization problem, affine transformation (Hagedoorn, Veltkamp)
  - combination with alignment (Mount)
  - combination with matchlist (Breuel)

Trafo Space Subdivision 1

- translation in 2D:



- $\rho(T) = \max_{t \in T} |(t - t_c)|$
- if $d(t_c(a), B) > \rho(T) + \varepsilon$
  then no $t \in T$ with $d(t(a), B) \leq \varepsilon$

Trafo Space Subdivision 2

- [Huttenlocher et al 93]
- U = translation + scaling:
- $\rho(U)=\max_{u \in U} |(s-s_c)\boldsymbol{a}+(t-t_c)|$
- $\boldsymbol{a} \geq a$ for all $a \in A$
- if $d(u_c(a),B) > \rho(U)+\varepsilon$
  then no $u \in U$ with $d(u(a),B) \leq \varepsilon$

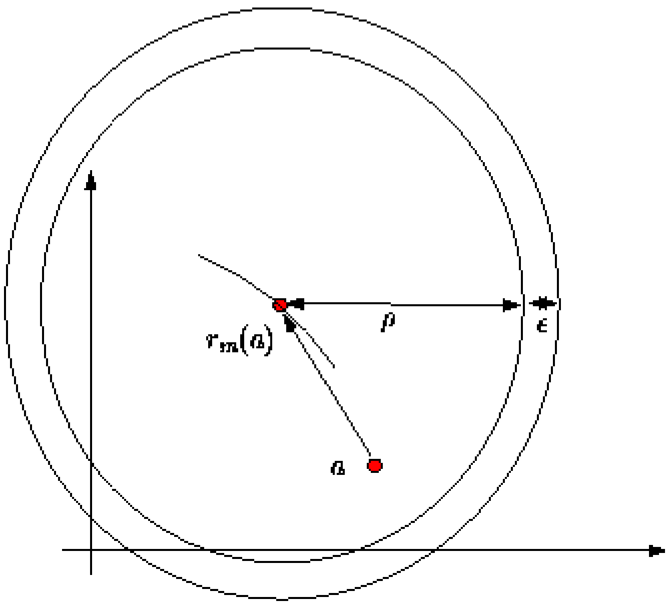
Trafo Space Subdivision 3

algorithm:

- if $h_k(u_c(A),B)>\varepsilon+\rho(U)$
- then return ϕ
- else if $h_k(u_c(A),B) \leq \varepsilon$
- then return U
- else split U and recur

Trafo Space Subdivision 4

- rotation:



- $Rot_{r1}(a)$-$Rot_{r2}(a)| \leq |a|\ |r1\text{-}r2|$
- M = rotation + translation
- sperical volume:
- $\rho(M)=\max_{r \in R} |r-r_c|\ |\boldsymbol{a}| + |(t-t_c)|$
- if $d(m_c(a),B) > \rho(M)+\varepsilon$
- then no $m \in M$ with $d(m(a),B) \leq \varepsilon$

Trafo Space Subdivision 5

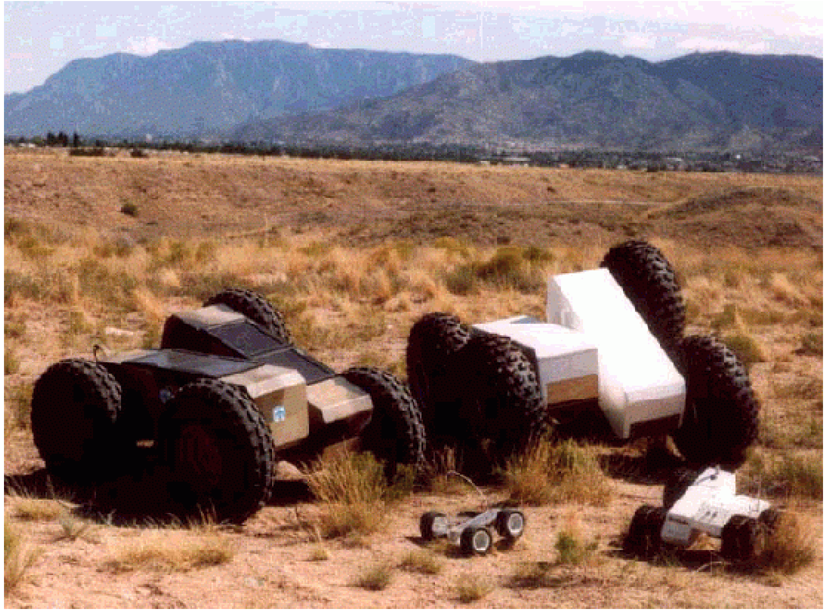
- [Hagedoorn & Veltkamp 97], traced volumes:
- $\tau(M,a) \supseteq \{x | x = m(a), m \in M\}$
- if $\#(a\!: \tau(M,a) \oplus \text{cube}(\varepsilon)) \cap B) = \phi > |A| - k$
- then no $m \in M$ with $h_k(m(A),B) \leq \varepsilon$

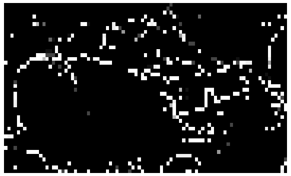
Algorithm:

- if $\#(a\!: \tau(M,a) \oplus \text{cube}(\varepsilon)) \cap B) = \phi > |A| - k$
- then return ϕ
- else if for all a, $\text{diam}(\tau(M,a) \leq \varepsilon)$
- then return M
- else split M

Trafo Space Subdivision 6

point sets
n-m





Trafo Space Subdivision 7

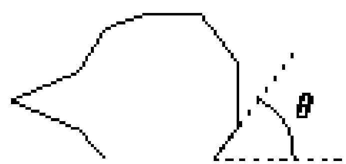
- model: 379 points
- test image: 15,520 points

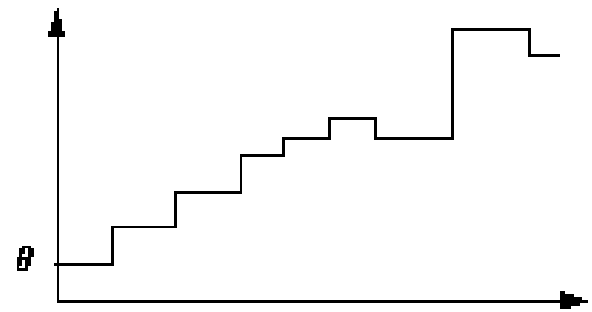
- transformation: 2D translation + scaling
- ε: 1 pixel width
- k: 360

- spherical volume: 5,786 cells, 9,375 seconds
- traced volume: 783 cells, 950 seconds

Turning Function 1

- cumulative angle function $\Theta_A(s)$:
  angle between ccw tangent and x-axis
  as a function of the arc length s
- for polylines:
  piecewise
  constant





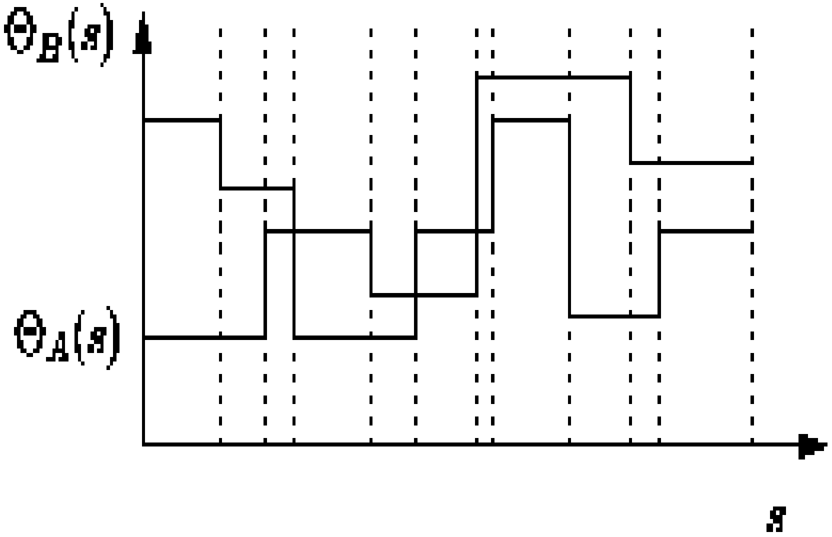
- increasing with left hand truns
- decreasing with right hand turns
- invariant under translation
- rotation over angle θ: vertical shift with θ

Turning Function 2

- dissimilarity function: L_p on functions
- $d_{A,B} = (\int | \Theta_A(s) - \Theta_B(s)|^p \, ds)^{1/p}$
- rotating by θ: $\Theta_A + \theta$
- invariance for rotation: minimize $d_{A,B}(\theta) = (\int |\Theta_A(s) - \Theta_B(s) + \theta|^p \, ds)^{1/p}$ for θ

Turning Function 3

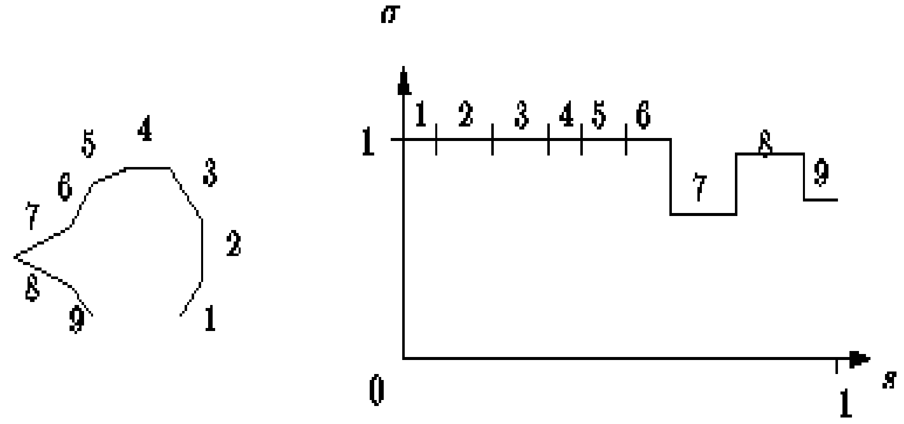
- matching: for p=2, $d_{A,B}(\theta)$ minimal for $\theta = \int \Theta_B(s)ds - \int \Theta_A(s)ds$
- polyline of unequal length: shift one along the other



- variation of [Arkin et al 91]

Signature Function

- less discriminative than turning function
- at every point: arc length to the left or on the tangent line



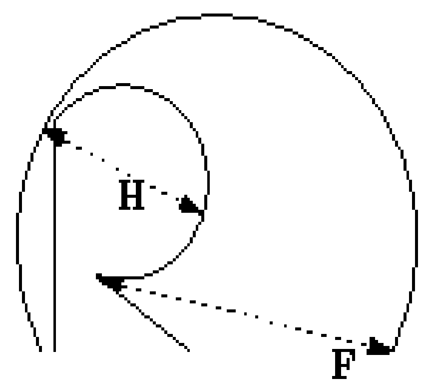
- invariant under translation, rotation, scaling
- convex curves: 1 everywhere
- constructed in $O(n^2)$ time [O'Rourke 85]

Signature Function

- matching two signature functions: ‘time warps’, pairing elements of A to elements of B
- using dynamic programming: O(nm) time

Fréchet Distance 1

- while walking forward along curves A and B, minimum over all walking speeds of max of distance between corresponding points
- can be larger than Hausdorff distance:



- more formally: min over all monotone increasing $\alpha(t),\beta(t)$ of max distance $d(A(\alpha(t)),B(\beta(t)))$

Fréchet Distance 2

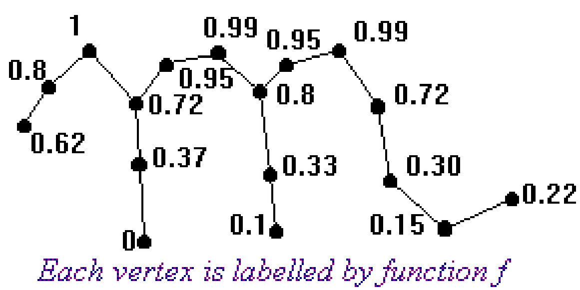
- polylines special case
- deciding $d_{A,B} < \varepsilon$: O(mn) time
- computing $d_{A,B}$:
  O(mn log(mn)) time (parametric search)
  O(mn(log (mn))^3) (simpler)
  O(mn(m+n) log (mn)) (still simpler)
- [Alt et al 95], [Godau 91]

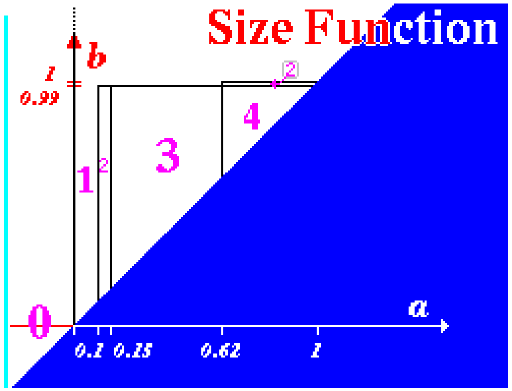
Fréchet Distance 3

- variation: drop monoticity condition
  - resulting distance $\delta_{A,B}$ is a semimetric
  - deciding $\delta_{A,B} < \varepsilon$: O(mn) time
  - computing $\delta_{A,B}$: O(mn log(mn)) time
- variation: partial matching, shifting one polyline along the other
  - deciding $\delta_{A,B} < \varepsilon$: O(mn log(mn)) time
  - computing $\delta_{A,B}$: $O(mn(\log(mn))^2)$ time

Size Function

- use measuring function f, e.g. normalized ordinate of vertex
- size function $z_f(a,b)$: number of connected components with f≤b that have at least one point with f ≤ a:

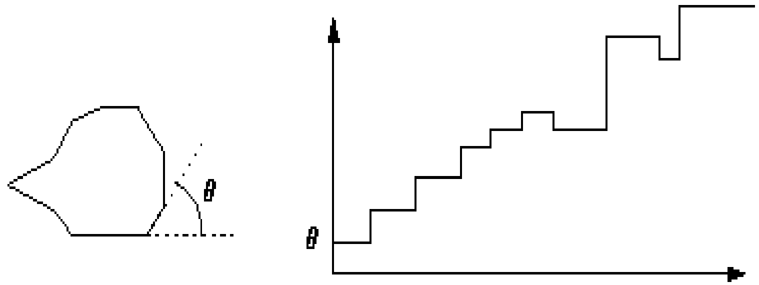




- matching: use corner points and multiplicity
- does not uniquely define a shape, but classes

Turning Function 1

- cumulative angle function $\Theta_A(s)$:
  angle between ccw tangent and x-axis
  as a function of the arc length s
- for polygons:
  piecewise
  constant



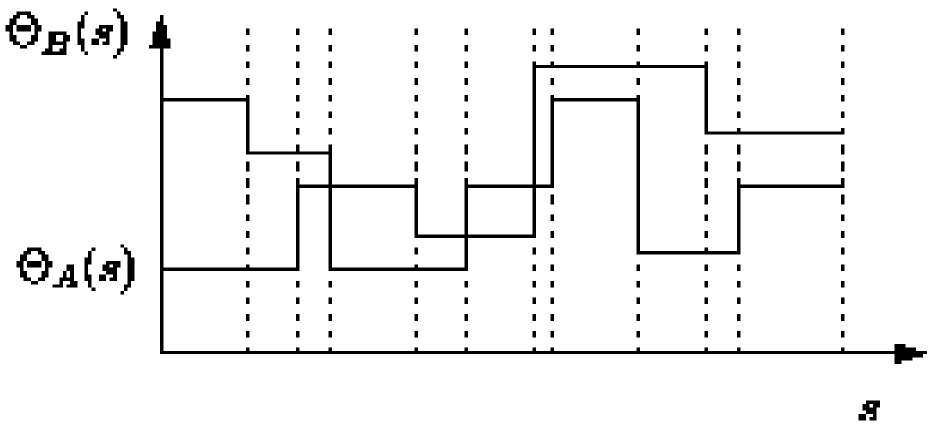
- increasing with left hand truns
- decreasing with right hand turns
- invariant under translation
- rotation over angle θ: vertical shift with θ

Turning Function 2

- dissimilarity function: L_p on functions
- $d_{A,B} = (\int | \Theta_A(s) - \Theta_B(s)|^p \, ds)^{1/p}$
- rescale polygons to have perimeter length 1
- shifting starting point by t: $\Theta_A(s+t)$
- rotating by θ: $\Theta_A + \theta$
- invariance for starting point and rotation: minimize $d_{A,B}(t,\theta) = (\int |\Theta_A(s+t) - \Theta_B(s) + \theta |^p \, ds)^{1/p}$ for t and θ
- unevenly spread noise is problematic

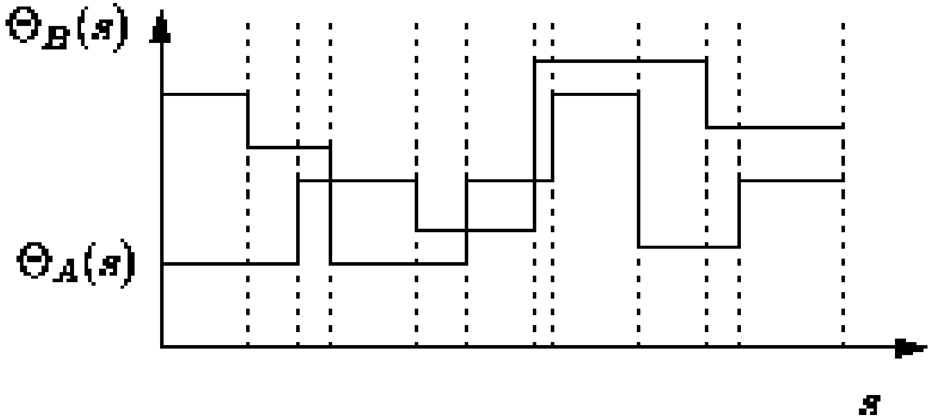
Turning Function 3

- for fixed t, and p=2, $d_{A,B}(t,\theta)$$ minimal for $\theta = \int \Theta_B(s)ds - \int \Theta_A(s)ds - 2\pi t$
- for polygons, Θ is step function
- $d_{A,B}(t, \theta)$ is sum of O(m+n) terms:



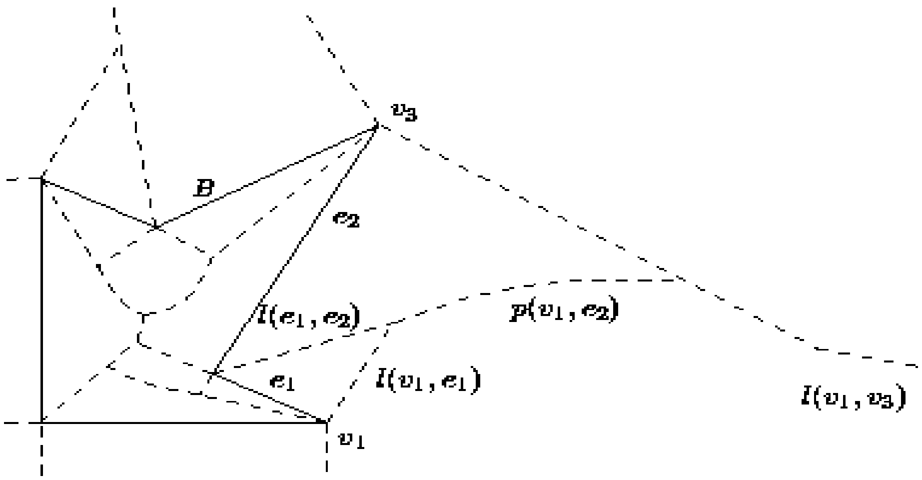
Turning Function 4

- minimum $d_{A,B}(t, \theta)$ when two steps coincide: O(mn) possibilities
- giving O(mn(m+n)) algorithm
- incremental evaluation: O(mn log(mn))
- [Arkin et al 91]



Hausdorff Distance 1

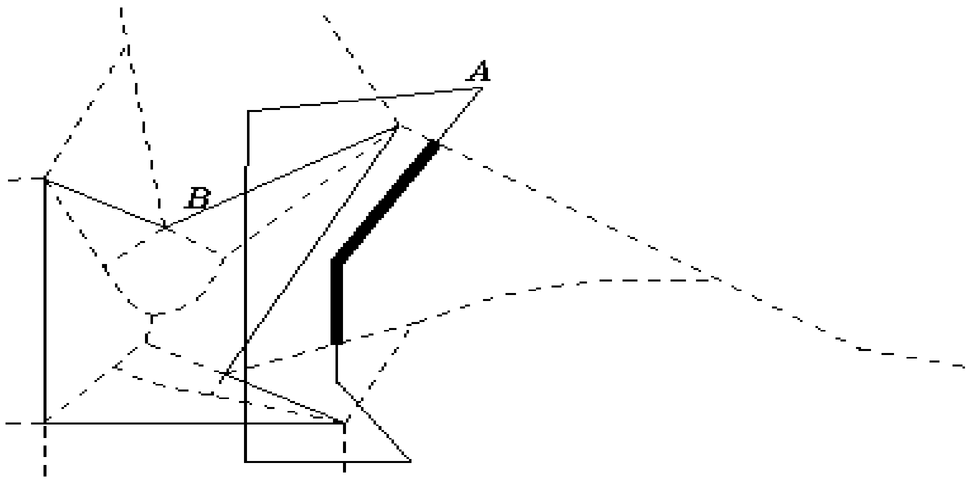
- computed with Voronoi diagram
- each vertex/edge of polygon B has region of points closer to that vertex/edge than to any other:



- line segments and parabolic segments
- Voronoi diagram of B has O(n) edges
- computed in time O(n log n)

Hausdorff Distance 2

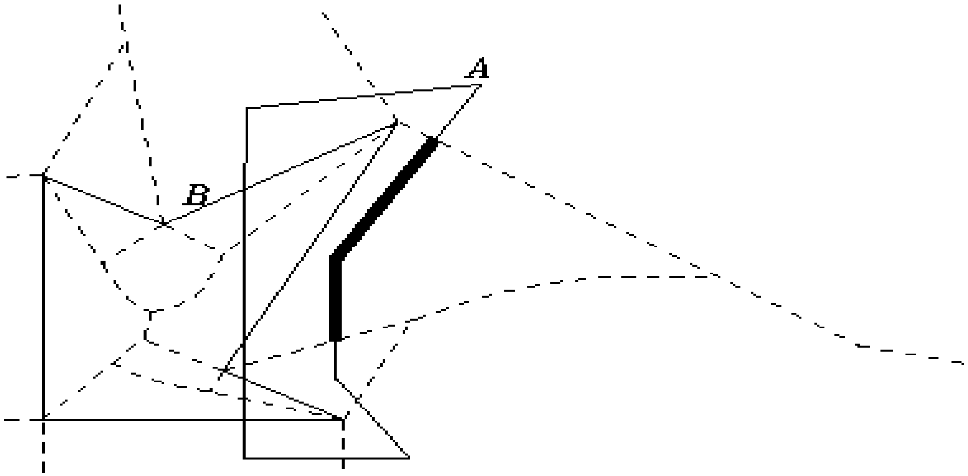
- directed Hausdorff dist from A to B: h(A,B)
- consider part of A falling in single region of VD(B):



- moving along the thick polyline, h(A,B) decreases, than increases
- max distance at intersection of thick segments with VD(B)

Hausdorff Distance 3

- in general:
  max distance
  at vertex of A
  or at intersection
  A with VD(B)



- at those points, compute distance to B and take the maximum
- with sweep line algorithm: O((m+n) log(m+n))
- same for h(B,A), take max{h(A,B),h(B,A)}
- [Alt et al 95]

Hausdorff Distance 4

- minimal Hausdorff dist under translation:
  - $O((mn)^2(\log(m+n))^3)$ time (parametric search)
  - $O((mn)^3(m+n)\log(m+n))$ time (simpler)
- approximation:
  - L_A: lower left corner of axis parallel bounding box
  - if f* gives minimal $h^*=h(f^*(A),B)$ then $d(L_{f^*(A)},L_B)\leq h^*\sqrt{2}$
  - so if f maps L_A onto L_B then $h(f(A),B) \leq(1+\sqrt{2})\, h$
  - determining f: $O(m+n)$ time
  - computing $d(f(A),B)$ still $O(m+n)\log(m+n)$

Hausdorff Distance 5

- minimal Hausdorff distance under rigid motions: $O((mn)^4(m+n)\log(m+n))$
- approximation:
  - K_A: centroid of edges of convex hull of A
  - suppose f* gives minimal h*=h(f*(A),B)
  - many rigid motions of A map K_A onto K_B
  - if g is the one that gives the smallest Hausdorff distance then $h(g(A),B) \le (4\pi+3)h(f^*(A),B)$
  - determining g: $O((mn)\log(mn)\log^*(mn))$ time
  - $O(\log^* n)$ is number of times that log has be applied to get down from n to below 1, for example $\log^*(2^{4294967296})$is only 6

Area of Overlap

- two simple polygons
- to computer area of overlap:
  - construct arrangement (combinatorial structure of point, edges, and facets of overlayof polygons)
  - time $O(n \log^* n+C)$, with C complexity of arrangement: number of vertices, edges, facets
  - after preprocessing of $O((mn)^2)$ time
  - overlap computed in time $O(\log(m+n))$
  - also for minimization under translation
- [Mount & Wu 96]

Area of Overlap

- convex polygons:
- optimization under translations: O((m+n) log(m+n)) time [de Berg et al 97]
- making centroids coincide gives overlap of at least 9/25 of the optimal

Area of Symmetric Difference

- making centroids coincide gives symmetric difference of at most 11/3 of the optimal under translations [Alt et al 96]
- also for trafos F other than translations:
  - if the centroid of A, c(A), is equivariant under transformations f: c(f(A))=f(c(A))
  - and if F closed under composition with translation

Concluding remarks

d(f(A),B), research issues:

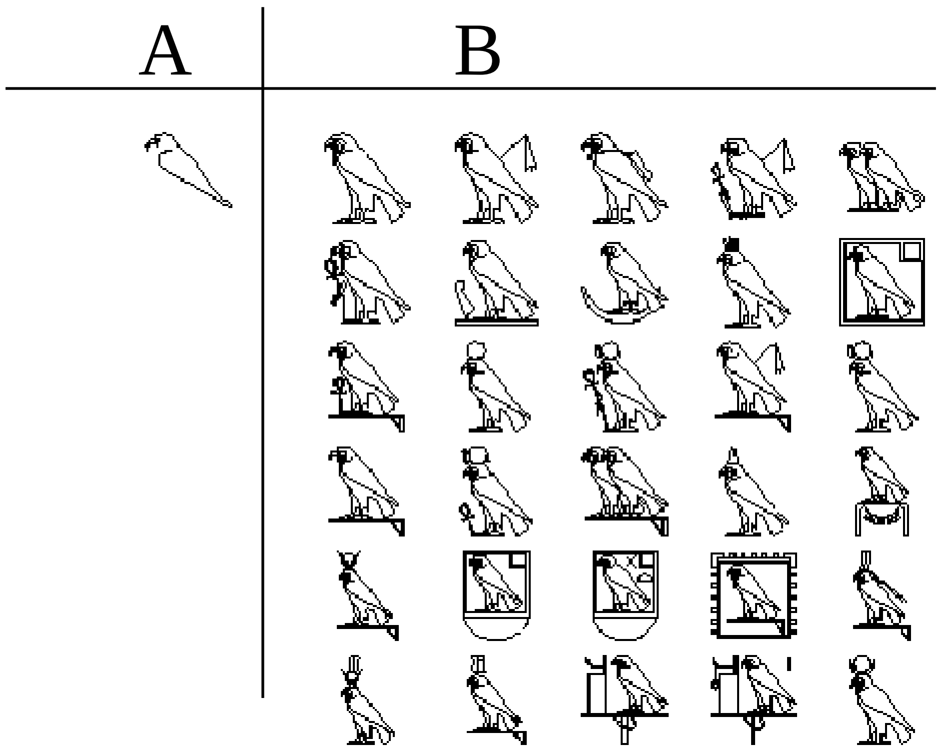
- partial matching: *d*
- matching two *sets* of curves, regions: *A*, *B*
- matching articulated figures : *A*, *B*
- matching under larger class of trafos: *f*

Example 1: application

6800 hieroglyphs, 72000 polylines from Center for Computer-aided Egyptological Research:

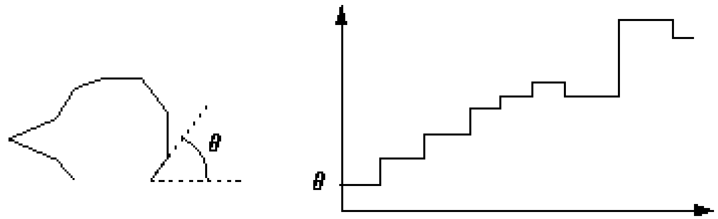
Example 1: problem

find B's for which there is g: d(g(A),B) $\leq \varepsilon$?



Example 1: similarity

turning function distance

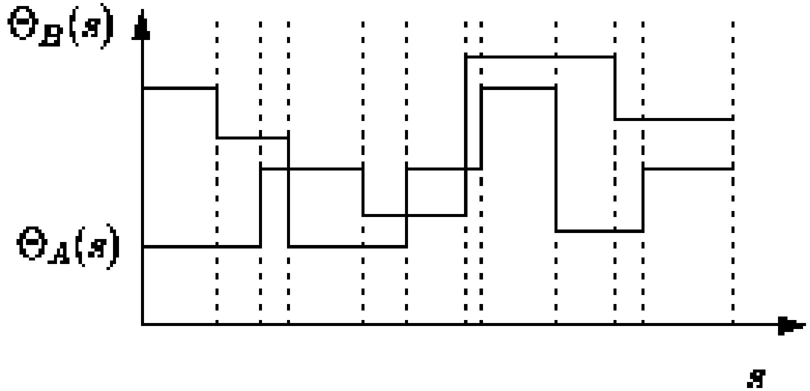


similarity is function distance

Example 1: similarity

turning function distance

$$d(A,B) = \left(\min_{t,\theta} \int_0^1 |\Theta_A(s+t) - \Theta_B(s) + \theta|^p \, ds \right)^{1/p}$$

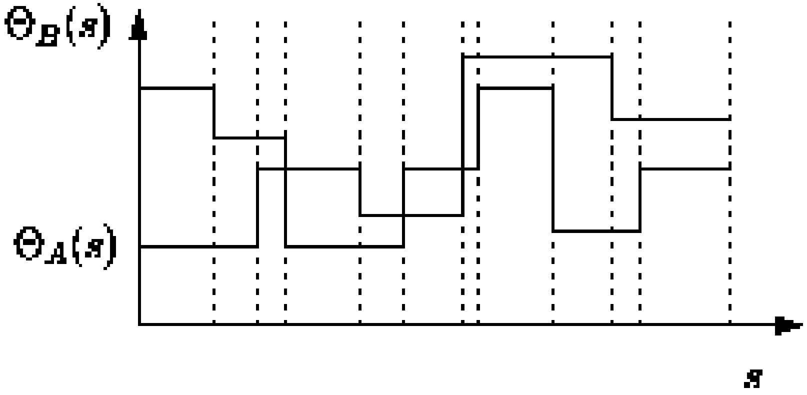


Example 1: properties

- triangle inequality
- translation, rotation invariant
- deformation robust

Example 1: algorithm

- $\theta^*(t) = \int g(s)\, ds - \int f(s)\, ds - 2\pi t$
- Naive evaluation: compute each of O(mn) shifts in O(m+n) time
- Incremental evaluation: O(mn log mn) time



Example 2: Weighted Point Set

- Certainty of position
  low weights should match easier
- Certainty of existence
  high weights should match easier
- Amount of some property
  match one distribution with another
- This example: amount of curvature

Example 2: similarity

Earth Mover's Distance between

$\{(p_1,w_1),\ldots,(p_m,w_m)\}$ and $\{(q_1,u_1),\ldots,(q_m,u_m)\}$

$$d = \frac{\min_F \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij} d_{ij}}{\min(\sum w_i, \sum u_j)}$$

$$f_{ij} \geq 0$$

$$\sum_j f_{ij} \leq w_i, \sum_i f_{ij} \leq u_j$$

$$\sum_i \sum_j f_{ij} = \min(\sum w_i, \sum u_j)$$

no triangle inequality for unequal total weights

Example 2: similarity

transportation distance between

$\{(p_1,w_1),\ldots,(p_m,w_m)\}$ and $\{(q_1,u_1),\ldots,(q_m,u_m)\}$

$$d = \frac{\min_F \sum_{i=1}^{n}\sum_{j=1}^{m} f_{ij} d_{ij}}{\sum_i w_i}$$

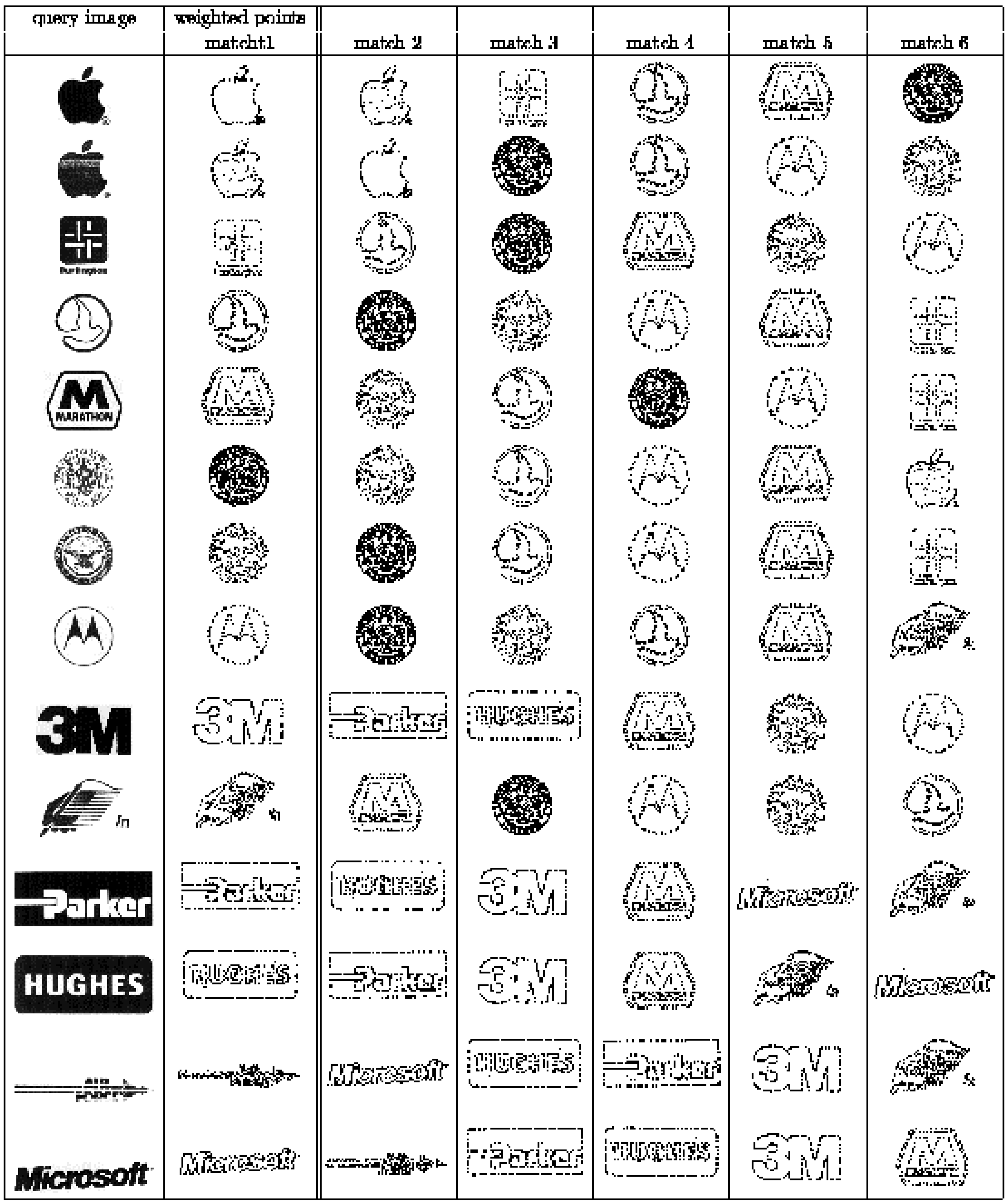
$$f_{ij} \geq 0, \quad \sum_i \sum_j f_{ij} = \sum_i w_i$$

$$\sum_i f_{ij} = \frac{u_j \sum_i w_i}{\sum_j u_j}, \quad \sum_j f_{ij} = w_i$$

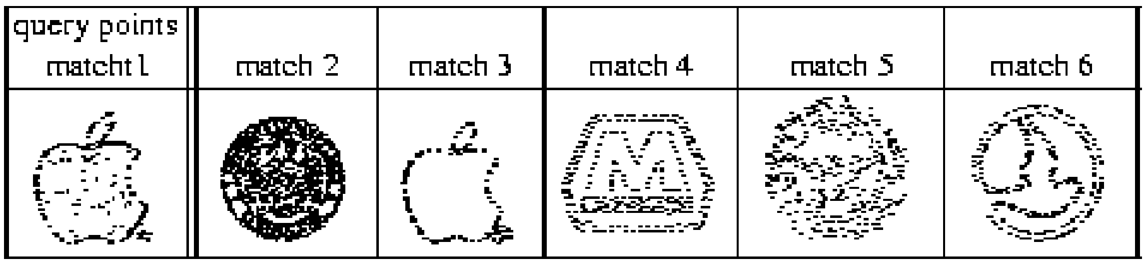
triangle inequality for unequal total weights

Example 2:

PTD Result



Example 2: EMD result

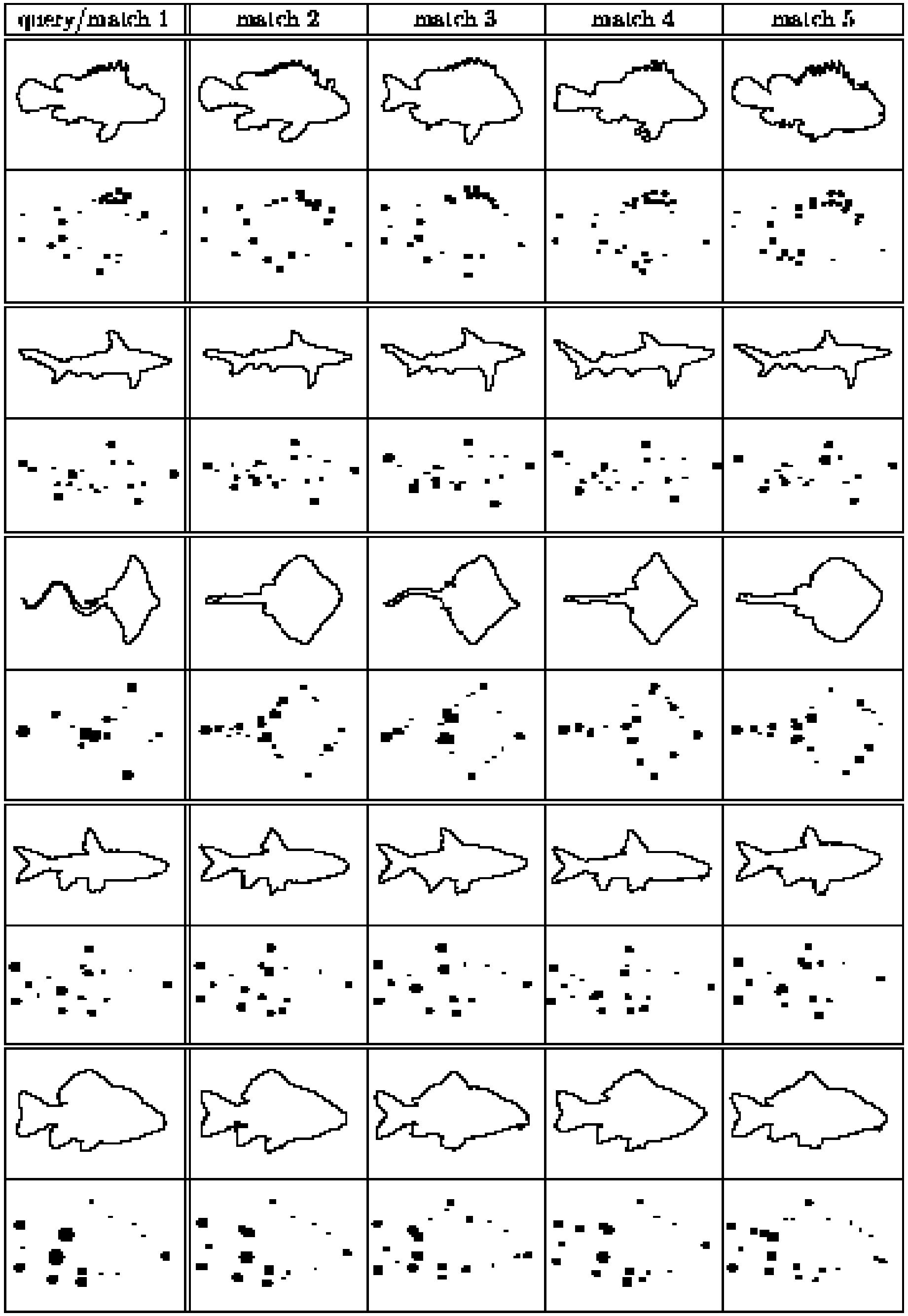


With EMD, 2nd apple not as 2nd match

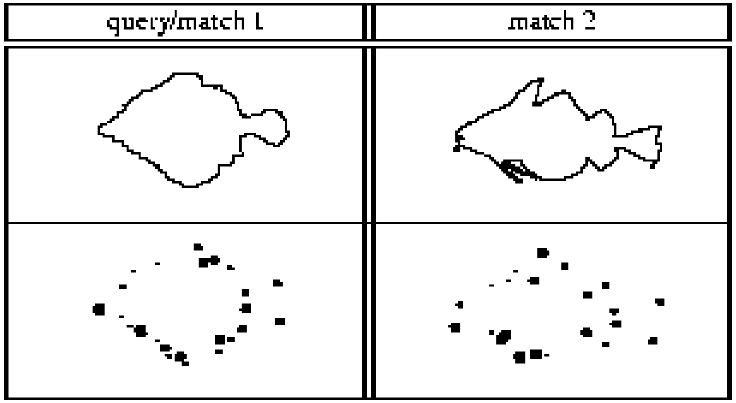
Example 2:

PTD Result

demo



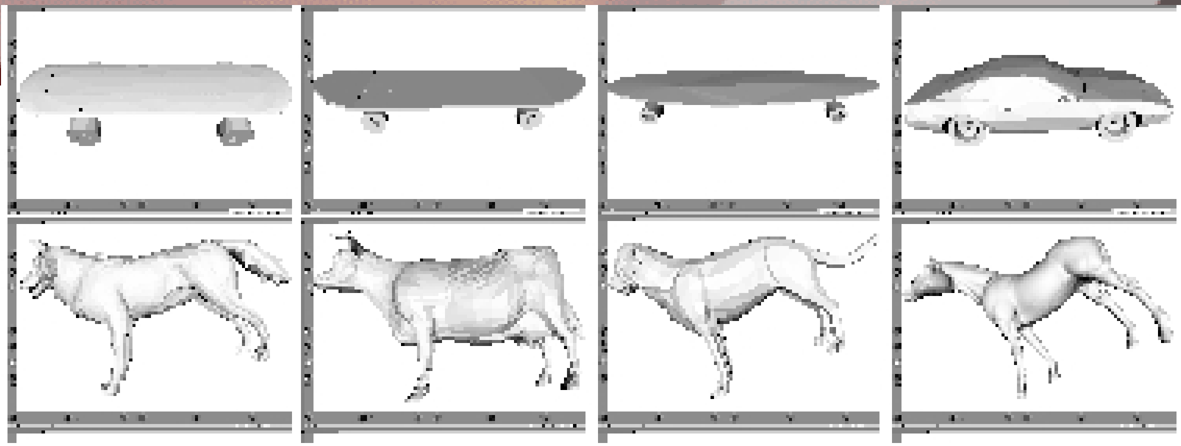
Example 2
Counter example



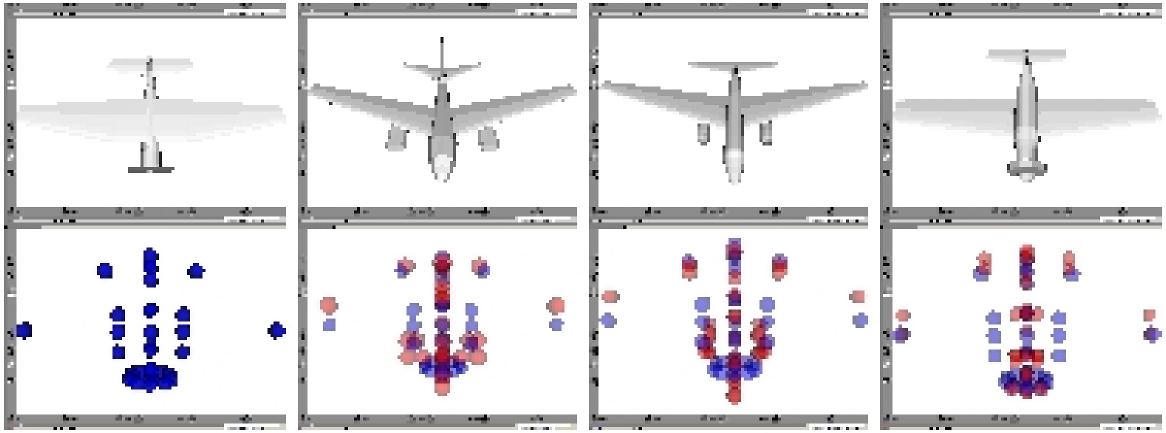
Example 2
3D models

- Polyhedral models
- In fixed grid, select those vertices with high curvature

Experimental results



“Princeton database”: 133 models classified by function into 25 classes



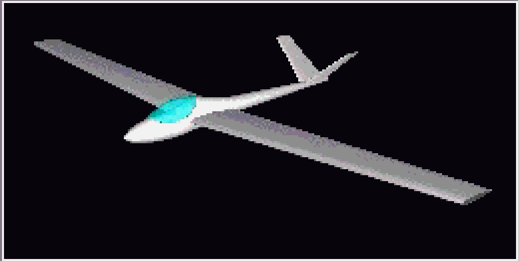
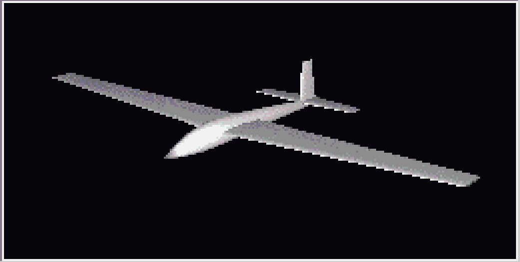
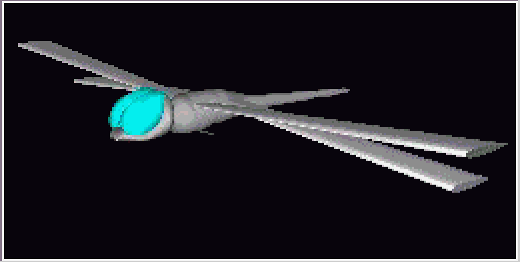
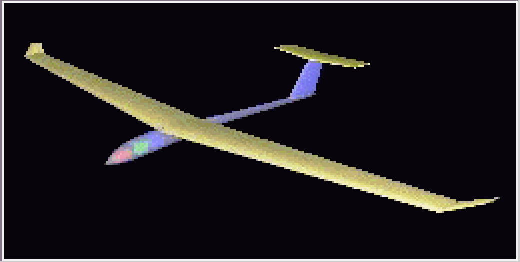
“Utrecht database”: 512 models classified by shape into 6 classes

Example 2:

PTD Result

demo

Results 1 to 10

Number: 349Gliders\Salto_H-101Distance: 0.0	vrml model feature comparison	
Number: 343Gliders\KobuzDistance: 0.0585621	vrml model feature comparison	
Number: 304Exotic\libellulaDistance: 0.0620732	vrml model feature comparison	
Number: 354Gliders\Ttail3Distance: 0.0686206	vrml model feature comparison	
Number: 337Gliders\ForeverFlierDistance: 0.0692359	vrml model feature comparison