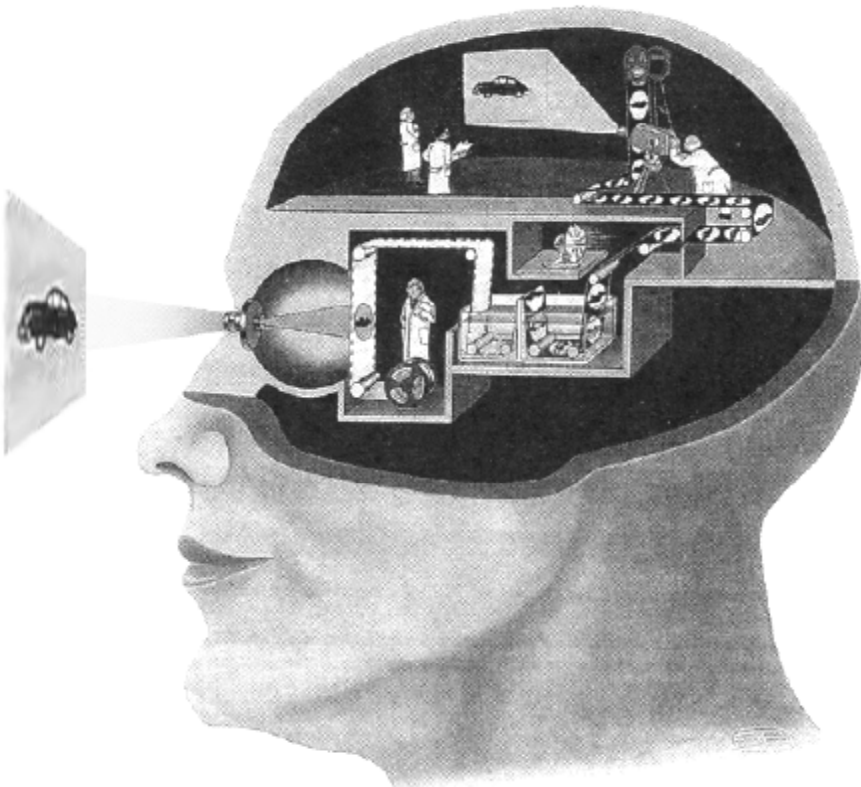


Shapes

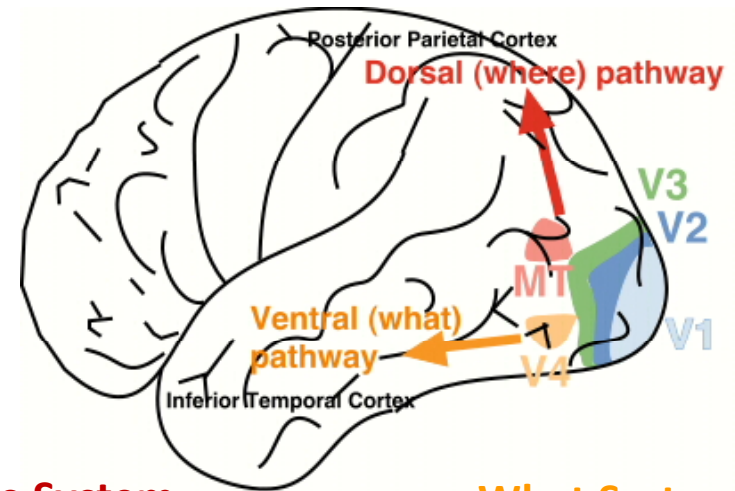
Giovanni Maria Farinella
gfarinella@dmi.unict.it



Human Visual System



Vision is light patterns processing (from light to information useful to the organism), not image transmission (Homunculus Theory).



Where System

- Location
- Motion perception
- Depth perception
- Spatial organization
- Figure/Ground segmentation

What System

- Object recognition
- Face recognition
- Color perception
- Shape perception

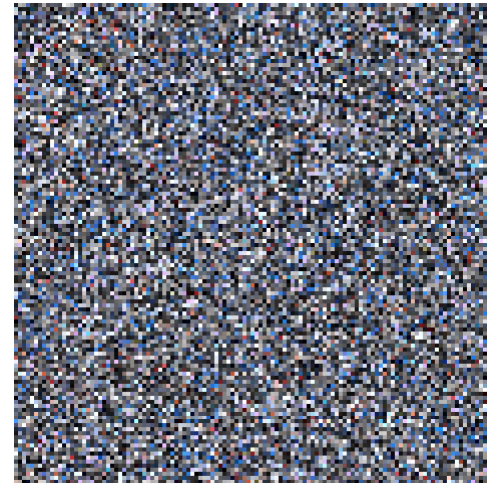
Among all the different visual information processed by the visual system, the shape certainly plays an important role.

You are reading characters on this slide... which are characterized by their shape!

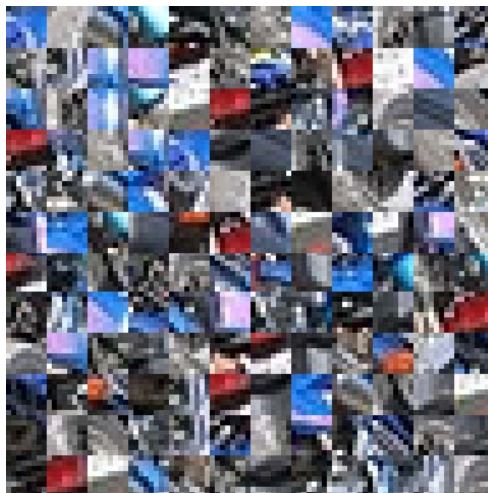
Which cue do we use most for object recognition?



► Colour



► Texture



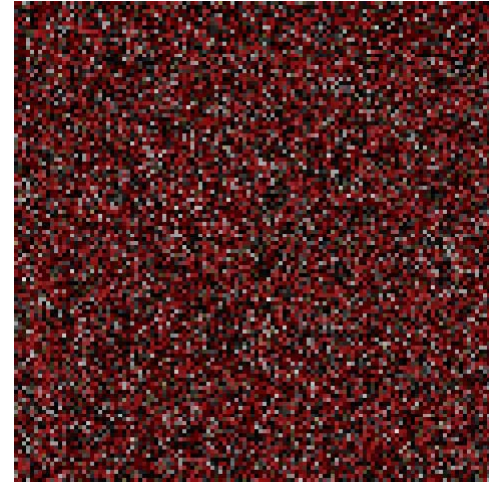
► Shape



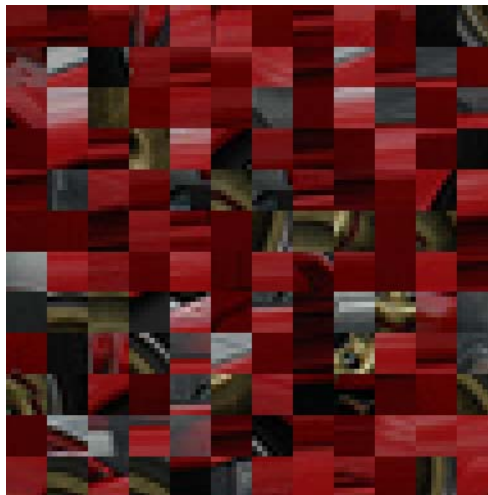
Guess the object



► Colour



► Texture



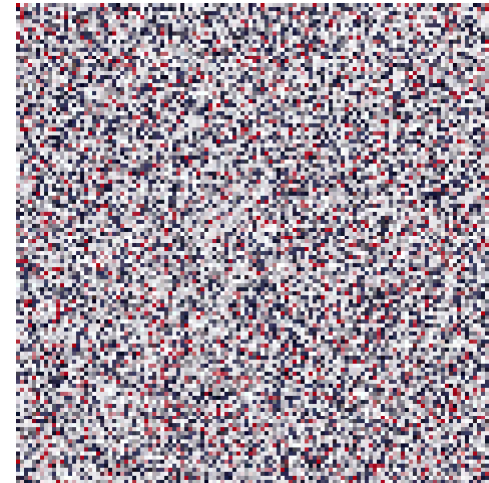
► Shape



Guess the object



► Colour



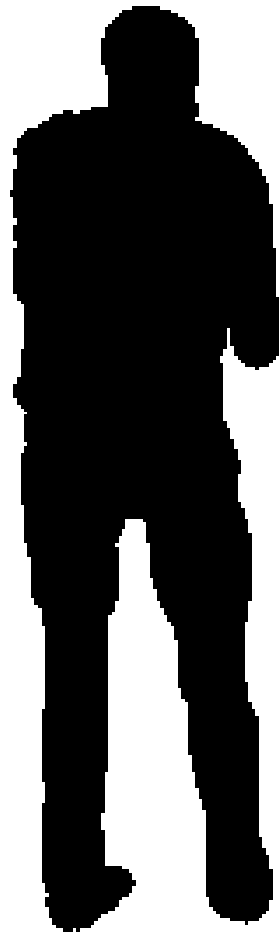
► Texture



► Shape



Shapes



The 2D-silhouette often conveys enough information to allow the recognition of an object.

Shape Recognition/Matching

- Though the recognition/matching of shapes is natural and simple for humans, the design of a robust computer vision system for this task is not straightforward.
 - Photometric Transformations (objects segmentation is a challenging task)
 - Low between class variability
 - High within class variability
 - Geometric Transformations
 - Deformations
 - Occlusion
 - Real-time processing

S

5

6

9

9

A E F H I K L M N T V W X Y Z

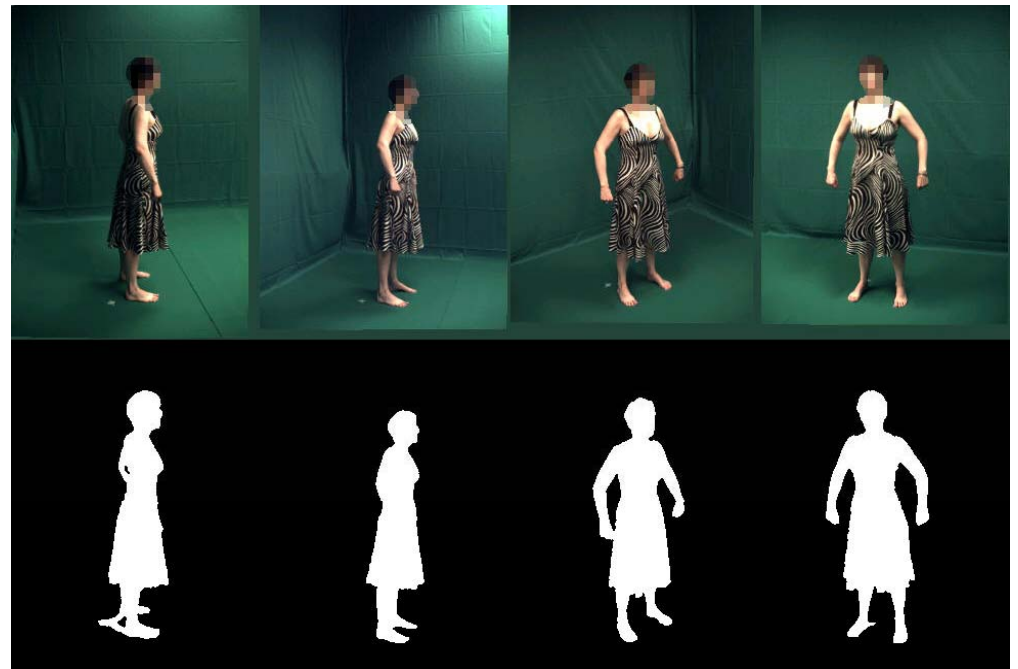
A E F H I K L M N T V W X Y Z

A E F H I K L M N T X W X Y Z

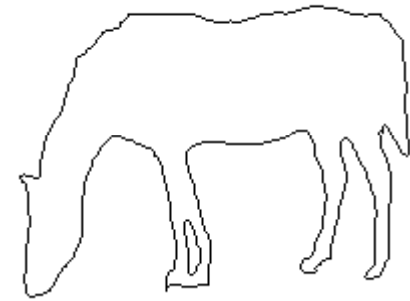
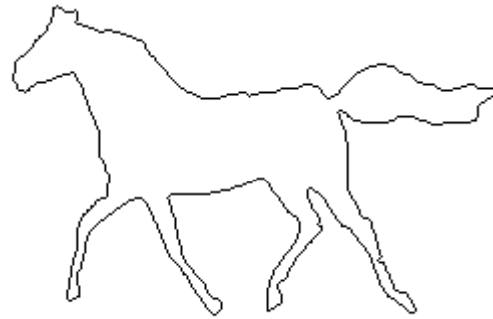
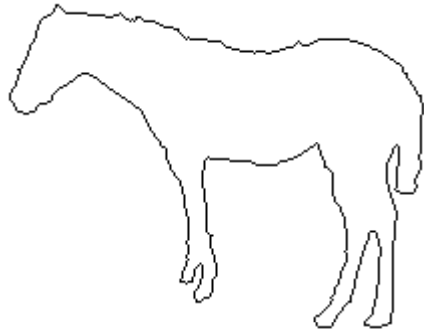


From 3D Object to 2D Shape

- When a 3-D real world object is projected onto a 2-D image plane, one dimension of object information is lost.
- As a result, the shape extracted from the image only partially represents the object.
- High variability



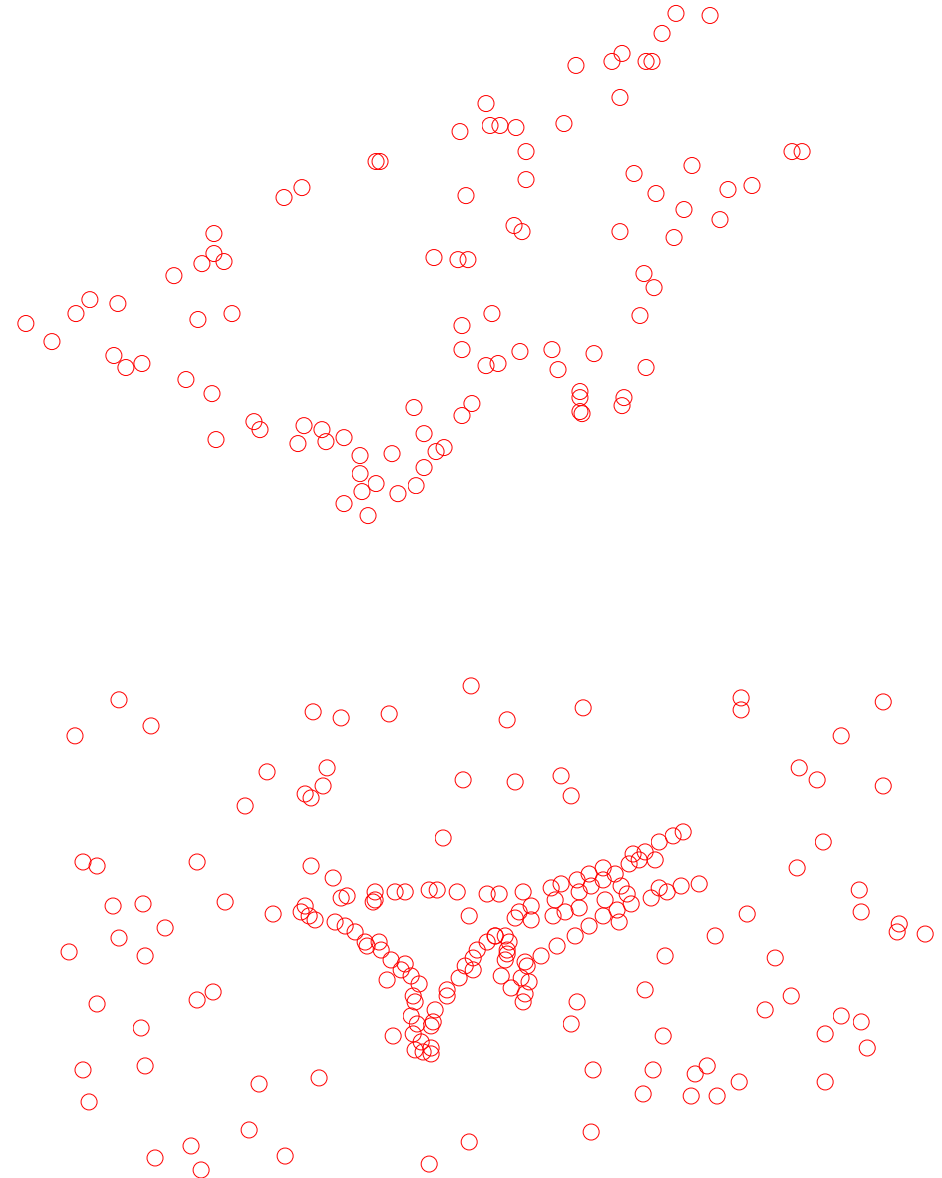
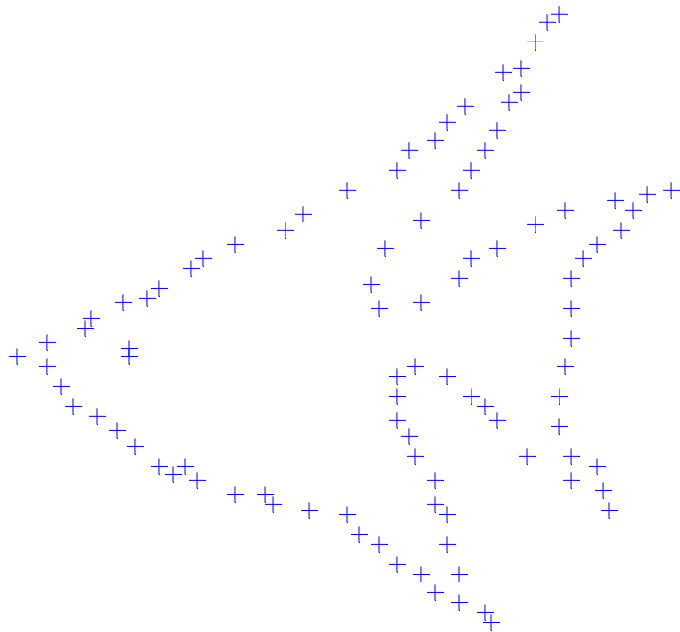
Deformable Objects



Occlusion



Noise and Outlier



Shape representation

- The algorithms for shape representation generally look for effective and perceptually important shape features based on either
 - shape boundary information
 - boundary plus interior content
- These approaches can be further distinguished into
 - space domain
 - transform domain

Contour shape techniques

- Contour shape techniques only exploit shape boundary information.
- Two types of different approaches for contour shape modeling:
 - continuous approach (global)
 - discrete approach (structural)

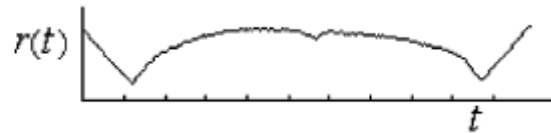
Contour-based (global)

- Continuous approaches do not divide shape into sub-parts
- Usually a feature vector derived from the integral boundary is used to describe the shape.
- The measure of shape similarity is usually a metric (or pseudo-metric) distance between the feature vectors describing the shapes.

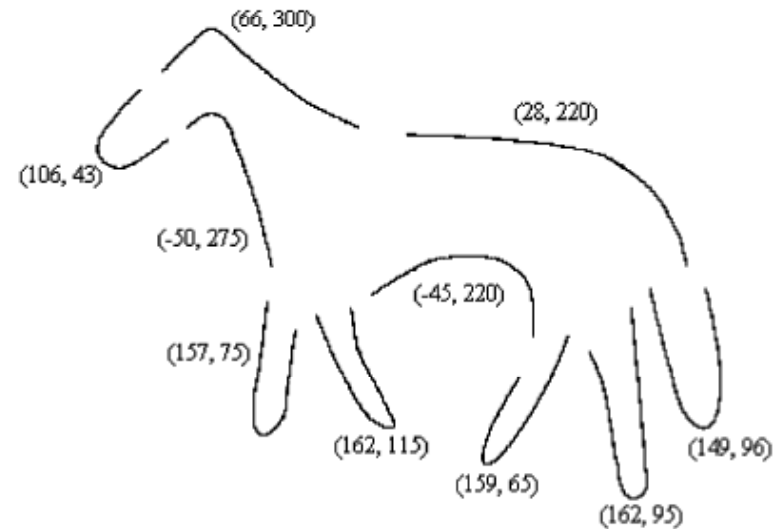
Contour-based (structural)

- Discrete approaches break the shape boundary into segments, called primitives using a specific criterion
- The final representation is usually a string or a graph (or tree)
- The similarity measure is done by string matching or graph matching.

Examples



Continuous approach:
an apple shape and its signature
obtained considering the distance of
the points of the shape from the centre
of mass

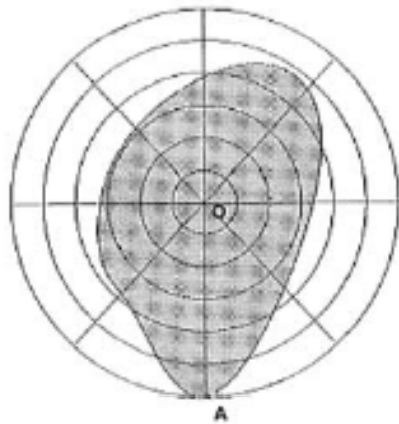


Structural methods: A horse shape has
been divided into different 'tokens'.
Numbers: the curvature and the
orientation of the token

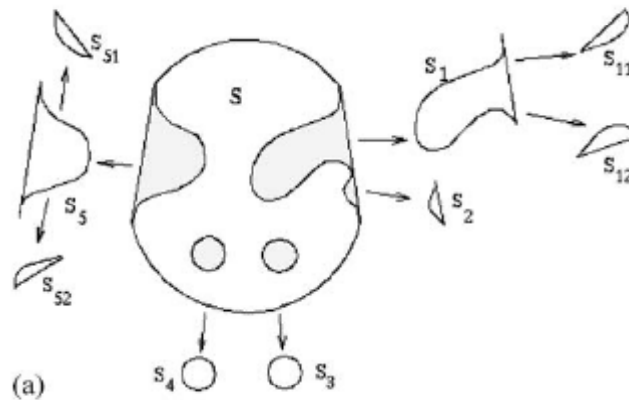
Region based techniques

- In region based techniques, all the pixels within a shape region are taken into account to obtain the shape representation
- Similar to contour based methods, region based shape methods can also be divided into global and structural methods

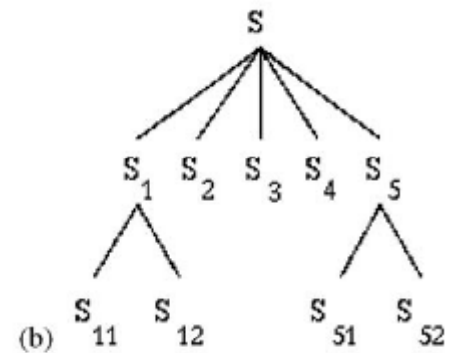
Examples



polar raster sampling
of shape



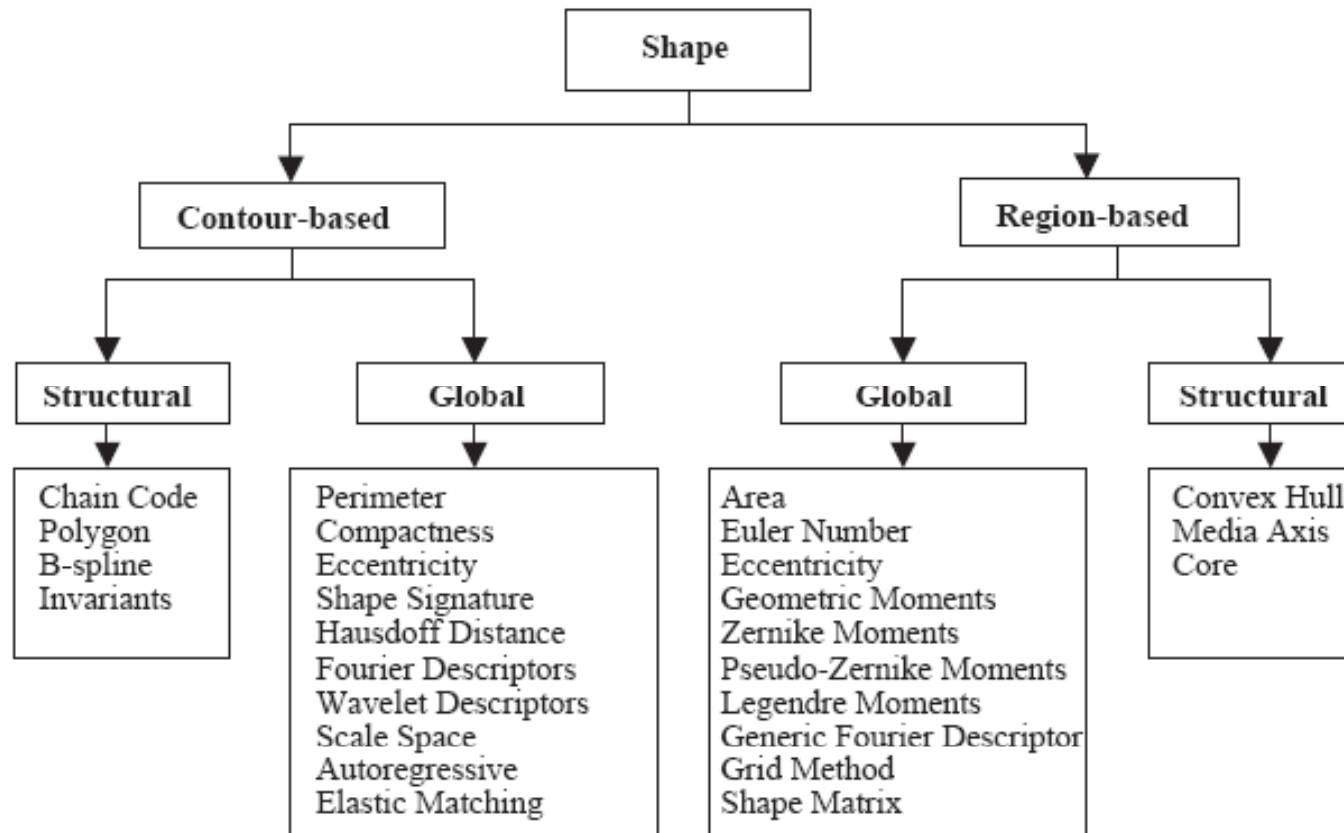
(a)



(b)

- a) Convex hull and its concavities;
- b) Concavity tree representation of convex hull

Classification of shape representation techniques



Common way for evaluation

- Shape features are often evaluated by how accurately they allow one to retrieve similar shapes from a given database.

MPEG-7 principles

- MPEG-7 has set several principles to measure a shape descriptor.
- Important characteristics:
 - good retrieval accuracy
 - robust retrieval performance
 - compact features
 - general application
 - low computation complexity
 - hierarchical coarse to fine representation

Retrieval accuracy

- Good retrieval accuracy requires a shape descriptor be able to effectively find perceptually similar shapes from a database.
- Perceptually similar shapes usually means rotated, translated, scaled shapes and affinely transformed shapes

Robustness requirement

- The descriptor should also be able to find noise affected shapes, variously distorted shapes and defective shapes, which are tolerated by human beings when comparing shapes

Computation complexity

- Low computation complexity means minimizing any uncertain or ad hoc factors that are involved in the derivation processes.
- The fewer the uncertain factors involved in the computation processes, the more robust the shape descriptor becomes.

*Low computation complexity =
clarity and stability*

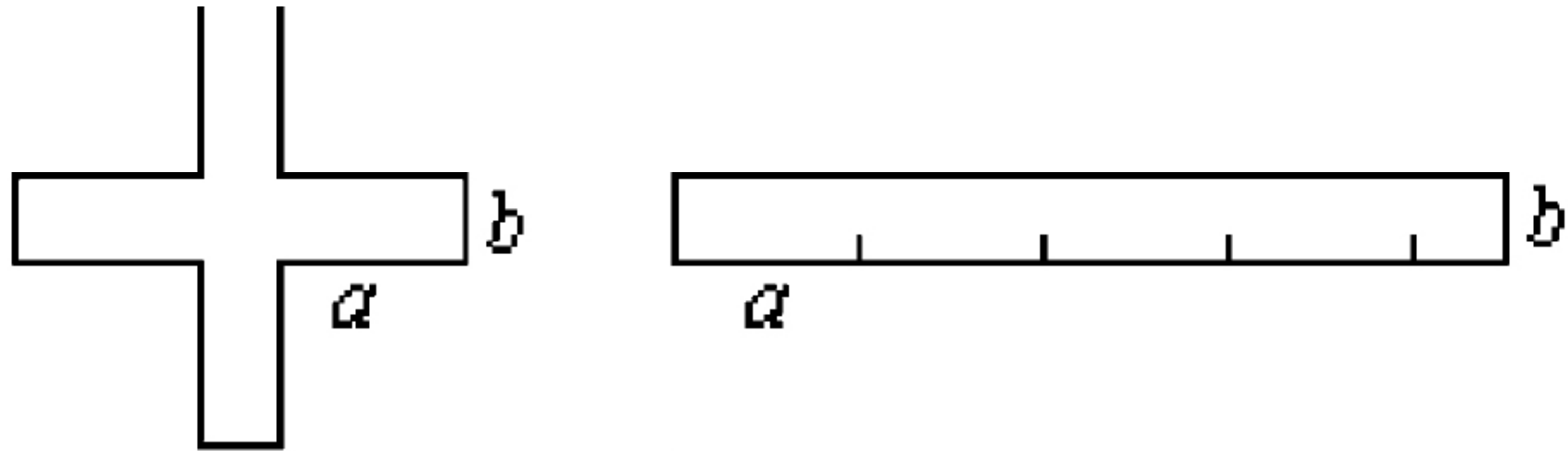
Other requirements

- Compact features are desirable for indexing and online retrieval.
- If a shape descriptor has a hierarchical coarse to fine representation characteristic, it can achieve a high level of matching efficiency.
 - shapes can be matched at coarse level to first eliminate large amount dissimilar shapes
 - at finer level, shapes can be matched in details.

Simple shape descriptors

- Common simple global descriptors are
 - *area*
 - *circularity* ($\text{perimeter}^2/\text{area}$)
 - *eccentricity* (length of major axis/length of minor axis)
 - *major axis orientation*
 - *etc*
- These descriptors can only discriminate shapes with large differences
- Usually used as filters to eliminate false hits or combined with other shape descriptors to discriminate shapes.

Simple shape descriptors (cont)



These two shapes have the same circularity ($\text{perimeter}^2/\text{area}$), however, they are very different shapes. In this case, eccentricity is a better descriptor.

Shape similarity and metrics

The measure of shape similarity is usually a metric distance between the considered feature vectors.

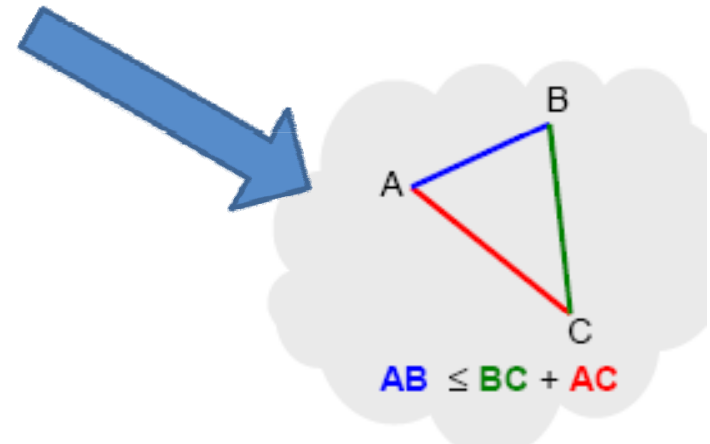
Metric

A function $d: X \times X \rightarrow \mathcal{R}$, satisfying for all

$x_1, x_2, x_3 \in X$

- *Non-negativity*: $d(x_1, x_2) \geq 0$
- *Indiscernability*: $d(x_1, x_2) = 0$ iff $x_1 = x_2$
- *Symmetry*: $d(x_1, x_2) = d(x_2, x_1)$
- *Triangle inequality*: $d(x_1, x_3) \leq d(x_1, x_2) + d(x_2, x_3)$

(X, d) is called a ***metric space***



Metrics



Euclidean



Manhattan



Geodesic

Correspondence-based shape matching

- Correspondence-based shape matching works in the space domain.
- In contrast to feature-based shape representation techniques, correspondence-based shape matching measures similarity between shapes using point-to-point matching.
- Every point on the shape is treated as a feature point.
- The matching is conducted on 2-D space.
- Example: Hausdorff distance, Chamfer Matching, Shape Context

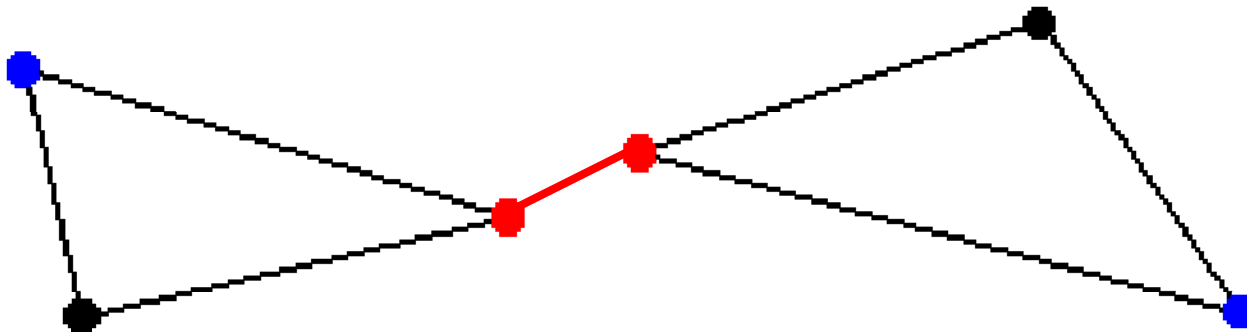
Distances between sets of points

- When talking about distances, we usually mean the shortest.
- Formally, this is called a *minmin* function:

$$D(A, B) = \min_{a \in A} \left\{ \min_{b \in B} \{d(a, b)\} \right\}$$

- It reads like: “for every point a of A , find its smallest distance to any point b of B ; finally, keep the smallest distance found among all points a ”

Example of minimin distance



The shortest distance doesn't consider the whole shape.

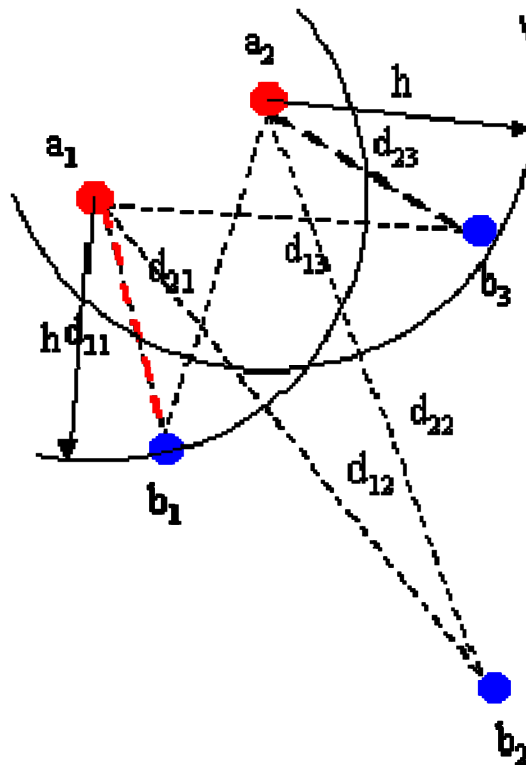
Hausdorff distance

- Hausdorff distance is the “*maximum distance of a set to the nearest point in the other set*”
- More formally, Hausdorff distance from set A to set B is a *maxmin* function, defined as:

$$h(A, B) = \max_{a \in A} \left\{ \min_{b \in B} \{d(a, b)\} \right\}$$

where a and b are points of sets A and B respectively, and $d(a, b)$ is any metric between these points

Example



Now we can say that
 any point of A is at
 distance $h(a_i, B)$ from B.
 Hence, the Hausdorff distance of
 A and B is $\max\{h(a_i, B)\}$.

$$h(A, B) = \max\{h(a_i, B)\}$$

to some point of B.

Hausdorff distance (cont)

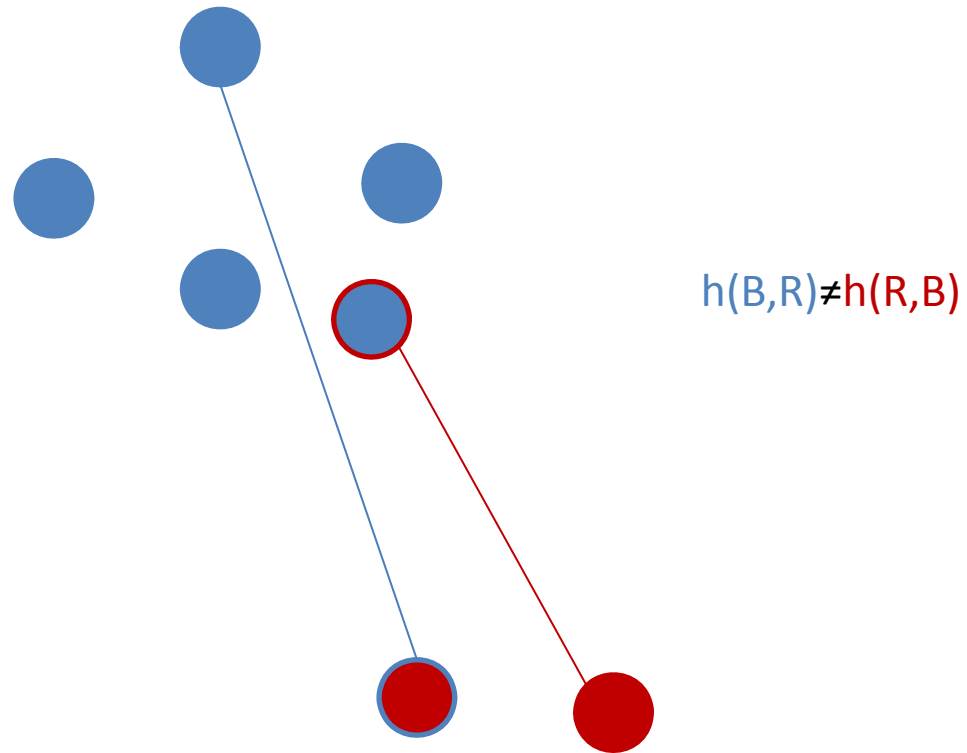
- Usually $d(a,b)$ is the Euclidean distance:

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

- It should be noted that Hausdorff distance is *oriented* (or *asymmetric*):

$$h(A, B) \neq h(B, A)$$

- Using this definition, the Hausdorff distance is not a metric!



Hausdorff distance (cont)

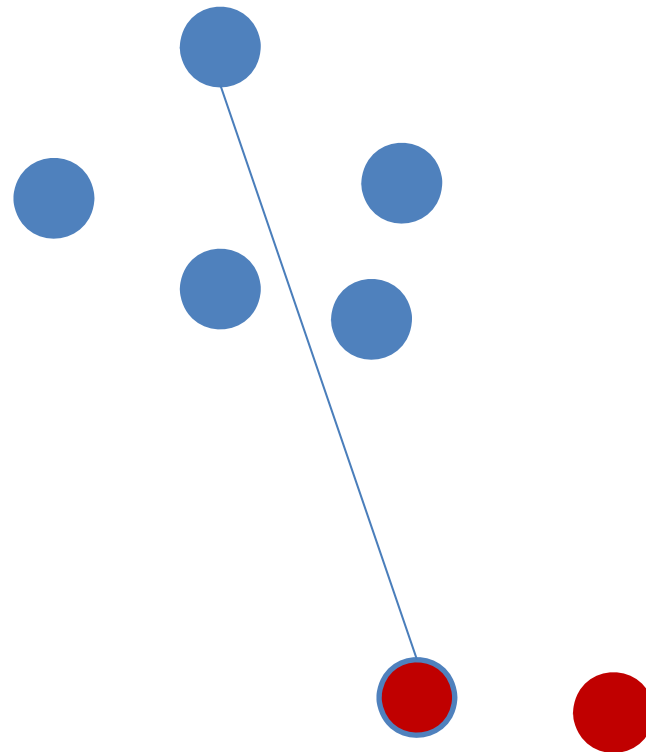
- A more general definition for the Hausdorff distance (which “fixes” this asymmetry) is:

$$H(A, B) = \max\{h(A, B), h(B, A)\}$$

which is a *pseudo-metric* and defines the distance between A and B

- Unlike a metric space, points in a pseudometric space need not be distinguishable:

one may have $d(x, y) = 0$ for $x \neq y$



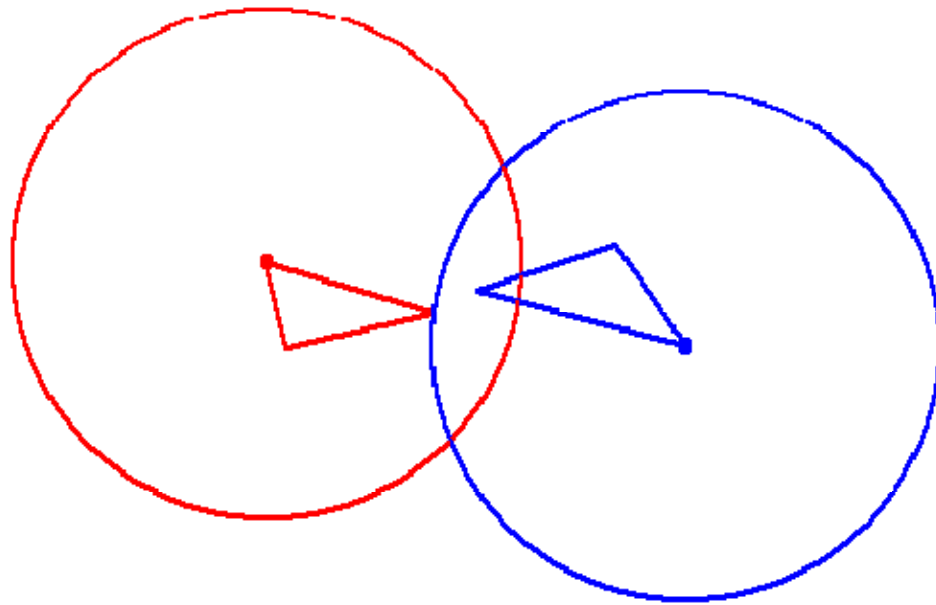
$$\text{Max}(H(\text{B,R}), H(\text{R,B})) = h(\text{B,R})$$

Hausdorff distance (cont)

- Hausdorff distance is a classical correspondence-based shape matching method, it has often been used to locate objects in an image and measure similarity between shapes
- Shape can be matched partially

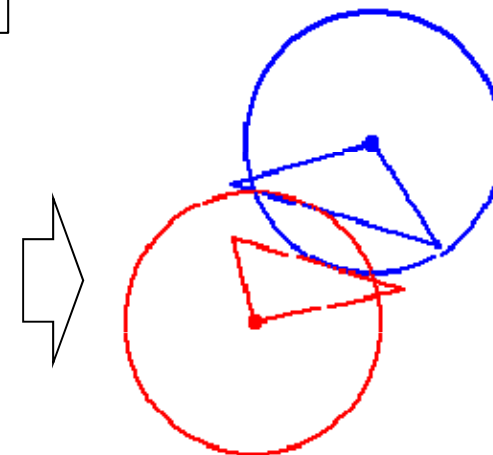
However:

- This distance measure is too sensitive to noise or outlier
- This distance is not translation, scale and rotation invariant.



Hausdorff distance shown around extremum of each triangles. Each circle has a radius of $H(P_1, P_2)$.

Hausdorff distance for the triangles at the same shortest distance, but in different position



Hausdorff distance (cont)

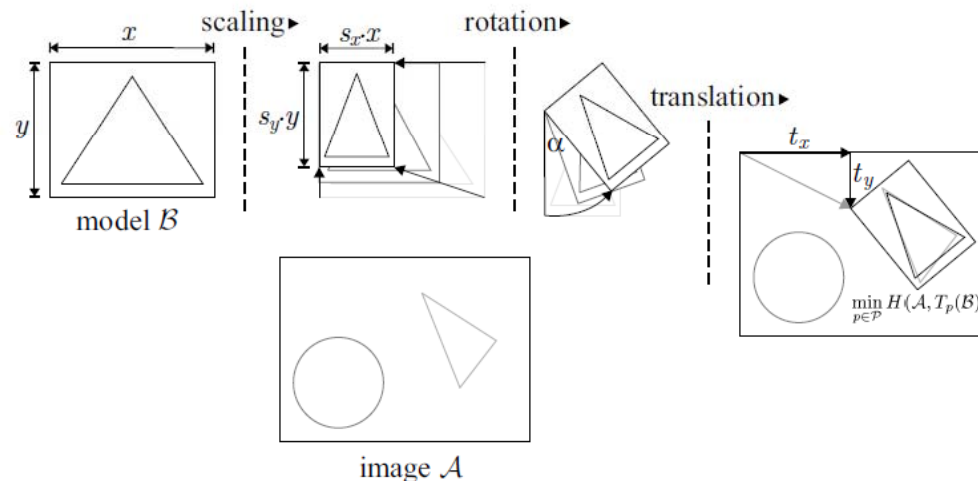
- In order to match a model shape with a shape in the image, the model shape has to be overlapped on the image in different positions, orientations and scales
- A single point in A that is far from anything in B will cause $h(A, B)$ to be large.
- Therefore, a modified Hausdorff distance is introduced by Rucklidge:

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

- Where $f^{th}_{x \in X} g(x)$ denotes the f^{th} quantile value of $g(x)$ over set X . In practice, f is usually set to be $\frac{1}{2}$ (the median)

Rucklidge algorithm

- Rucklidge extends Hausdorff distance matching into affine invariant matching
- A set of affine models are generated from the model shape.
- Since the space of affine transformations from the model shape is large, an efficient matching scheme is introduced by only examining a small part of the space of the affine transformations.
- Despite the efficiency effort, the matching load is still high



Example

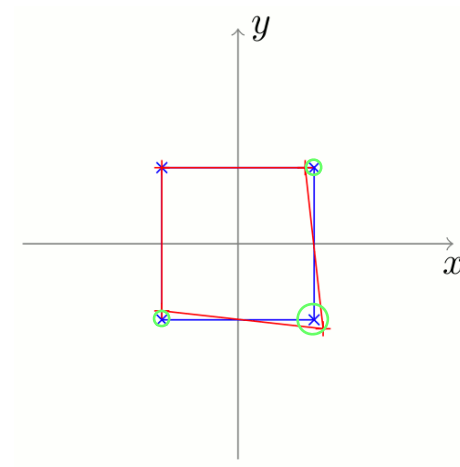
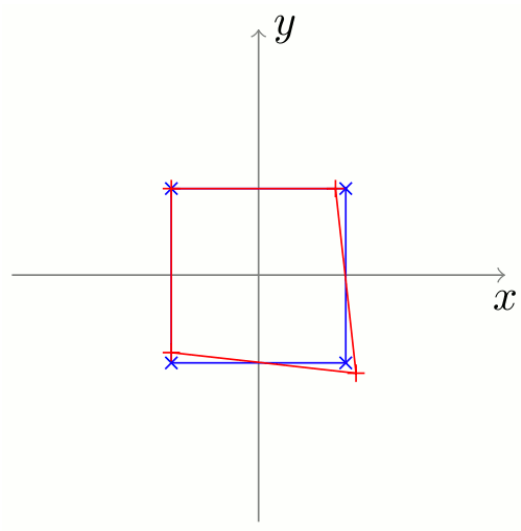
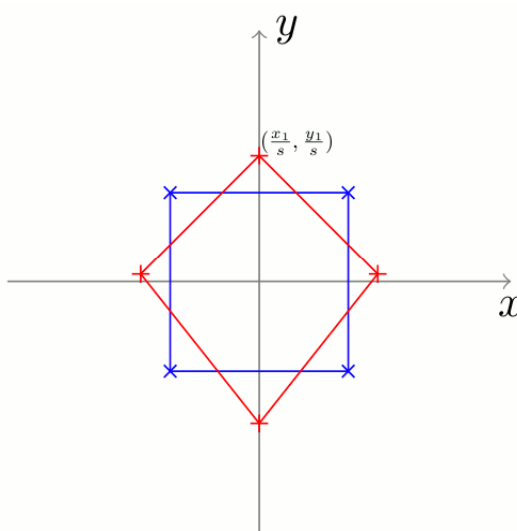
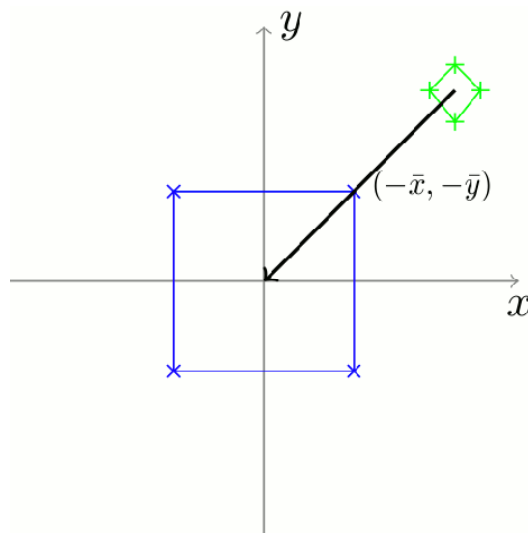
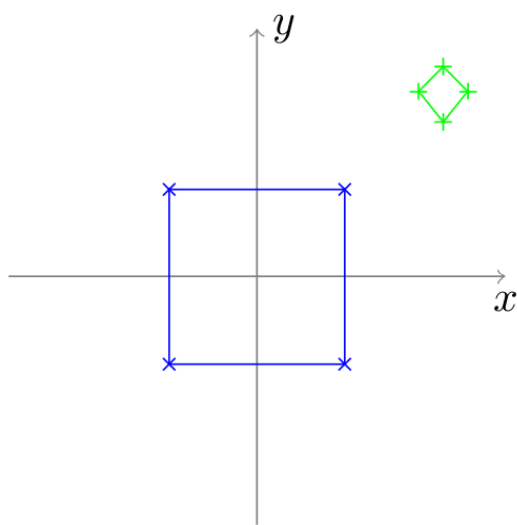


Application of Edge Detection algorithm

Procrustean distance

- The Procrustean distance is a least-squares type shape metric that requires shapes with one-to-one point correspondence.
- To determine the Procrustean distance between two shapes involves four steps:
 1. Compute the centroid of each shape.
 2. Re-scale each shape to have equal size.
 3. Align w.r.t. position the two shapes at their centroids.
 4. Align w.r.t. orientation by rotation.

Procrustean distance (cont)



Procrustean distance (cont)

- Mathematically the squared Procrustean distance between two shapes, x_1 and x_2 , is the sum of the squared point distances after alignment:

$$P_d^2 = \sum_{j=1}^n [(x_{j1} - x_{j2})^2 + (y_{j1} - y_{j2})^2]$$

Procrustean distance (cont)

- The *centroid* of a shape can be interpreted as center of mass of the physical system consisting of unit masses at each landmark. Thus to compute the centroid:

$$(\bar{x}, \bar{y}) = \left(\frac{1}{n} \sum_{j=1}^n x_j, \frac{1}{n} \sum_{j=1}^n y_j \right)$$

Procrustean distance (cont)

- To *re-scale each shape to have equal size* we need to establish a *size metric*.
- A **shape size metric** $S(\mathbf{x})$ is any positive real valued function of the shape vector that fulfils the following property:

$$S(a\mathbf{x}) = aS(\mathbf{x})$$

Procrustean distance (cont)

- We can use the *Frobenius norm* as a shape size metric:

$$S(x) = \sqrt{\sum_{j=1}^n [(x_j - \bar{x})^2 + (y_j - \bar{y})^2]}$$

- Another often used scale metric is the *centroid size*:

$$S(x) = \sum_{j=1}^n \sqrt{(x_j - \bar{x})^2 + (y_j - \bar{y})^2}$$

Procrustean distance (cont)

- To filter out the rotational effects the following *singular value decomposition* technique is suggested by Bookstein [1997]:
 1. Arrange the size and position aligned \mathbf{x}_1 and \mathbf{x}_2 as a $n \times k$ matrix
 2. Calculate the SVD, \mathbf{UDV}^T , of $\mathbf{x}_1^T \mathbf{x}_2$
 3. Then the rotation matrix needed to optimally superimpose \mathbf{x}_1 upon \mathbf{x}_2 is \mathbf{VU}^T . In the planar case:

$$\mathbf{VU}^T = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$$

Procrustean distance (cont)

- As an alternative Cootes et al. [2000] suggest a variation on Procrustean distance-based alignment by minimizing the closed form of $|T(\mathbf{x}_1) - \mathbf{x}_2|^2$ where T in the Euclidean case is:

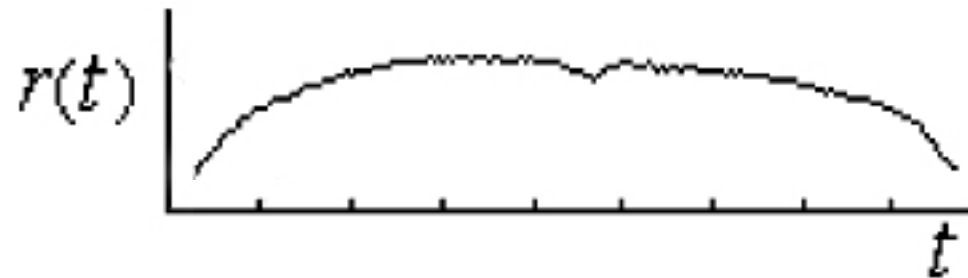
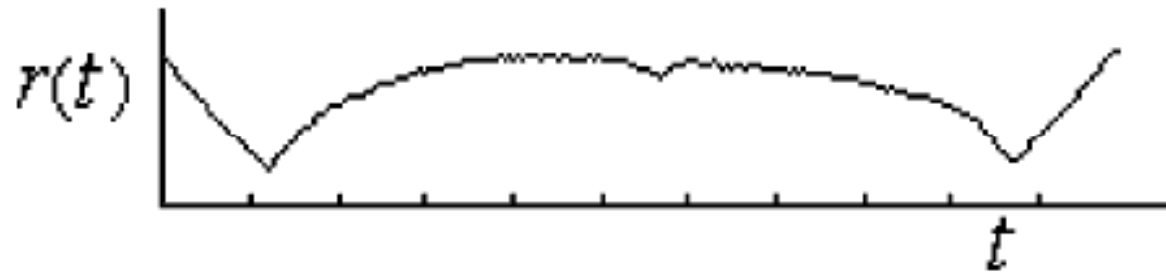
$$T \begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} a & -a \\ b & b \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

- The term $|T(\mathbf{x}_1) - \mathbf{x}_2|^2$ is then differentiated w.r.t. (a, b, t_x, t_y) .
- This transformation changes the actual shape.

Shape signature

- A Shape signature represents a shape by a one dimensional function derived from shape boundary points.
- Many shape signatures exist. They include centroidal profile, complex coordinates, centroid distance, tangent angle, cumulative angle, curvature, area and chord-length, etc..

Shape signature – centroidal profile



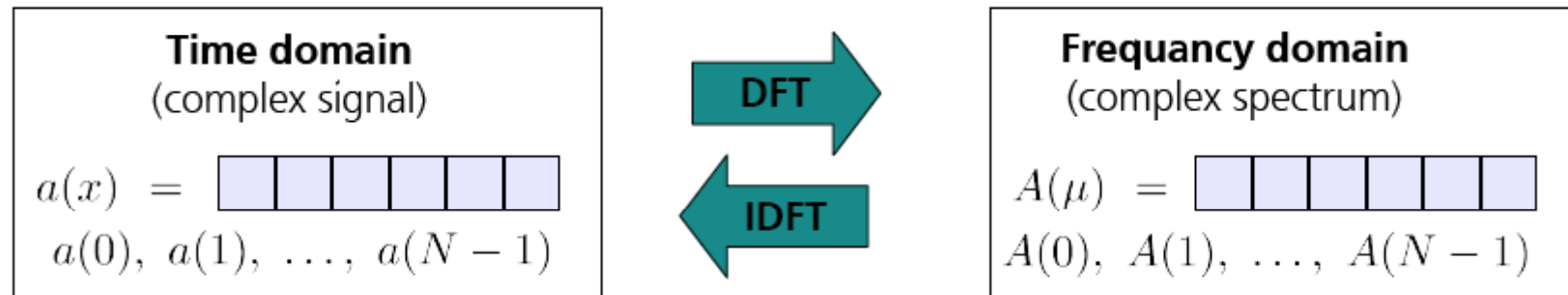
Shape signature (cont)

- Shape signatures are usually normalized to obtain translation and scale invariance.
- In order to compensate for orientation changes, shift matching is needed to find the best matching between two shapes.
- Most of the signature matching is normalized to shift matching in 1-D space,
- Some signature matching requires shift matching in 2-D space, such as the matching of centroidal profiles
- In either case, the matching cost is too high for online retrieval.
- In addition to the high matching cost, shape signatures are sensitive to noise, and slight changes in the boundary can cause large errors in matching.
- Therefore, it is undesirable to directly describe shape using a shape signature

Spectral descriptors

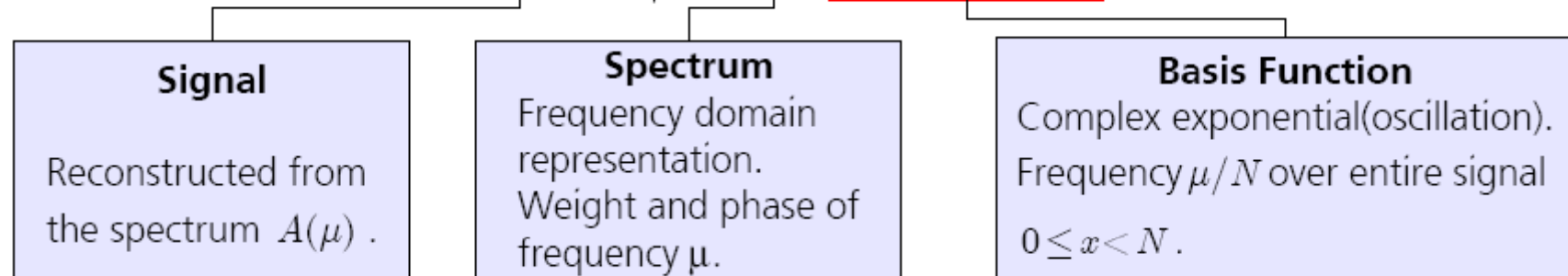
- Spectral descriptors overcome the problem of noise sensitivity and boundary variations by analyzing shape in spectral domain.
- Spectral descriptors include Fourier descriptors (FDs) and wavelet descriptors (WDs)
- They are derived from spectral transforms on 1-D shape signatures

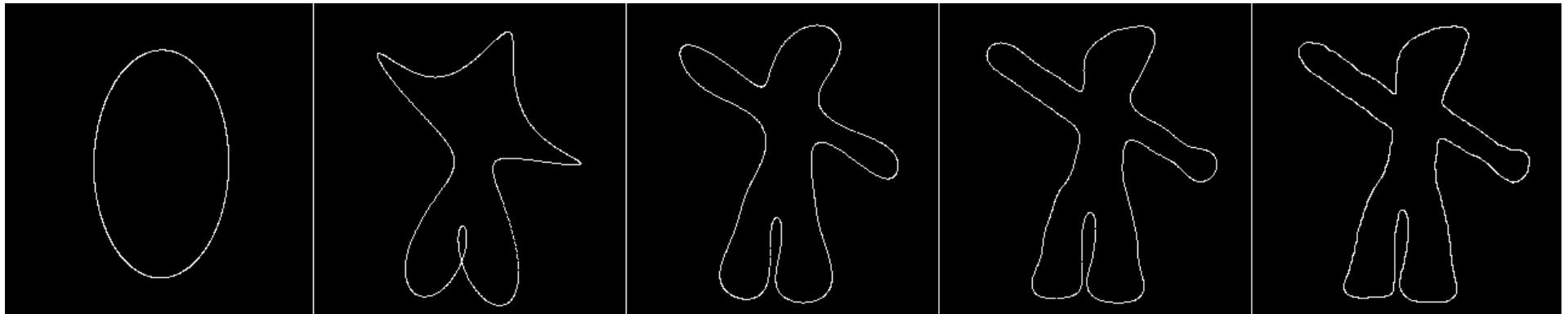
Fourier Transform



$$A(\mu) = \sum_{x=0}^{N-1} a(x) \exp\left(-\frac{2\pi i}{N} \mu x\right) \quad \text{Forward DFT (MATLAB `fft`)}$$

$$a(x) = \sum_{\mu=0}^{N-1} A(\mu) \exp\left(\frac{2\pi i}{N} \mu x\right) \quad \text{Inverse DFT (MATLAB `ifft`)}$$



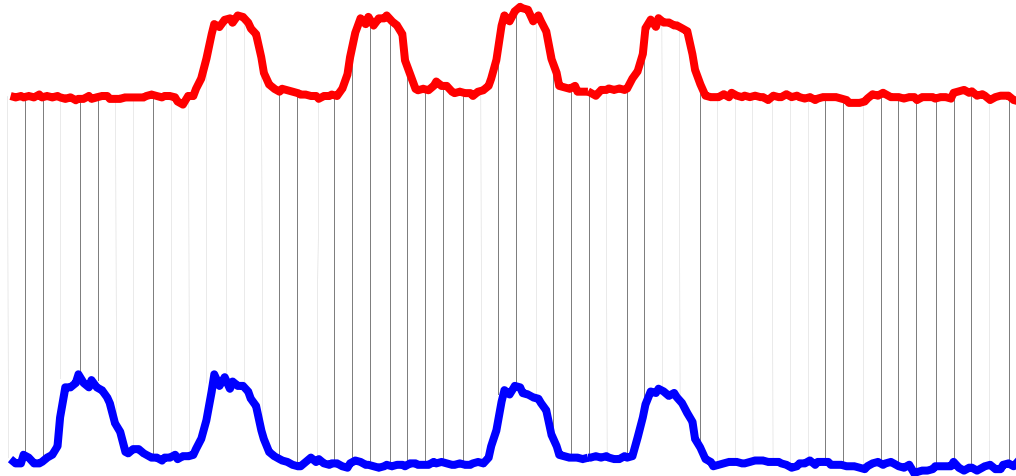


The above figures show the reconstruction based on different percentages of frequency components (descriptors) after the Fourier transform of the pixels on the boundary of the figure in the image. The percentages used are, from left to right 0.1%, 1%, 2%, 5% and 100%. It can be seen that the reconstructed figures using 5% of the frequency components are very similar to the actual figure (using one hundred percent of the frequency components on the right).

Dynamic Time Warping

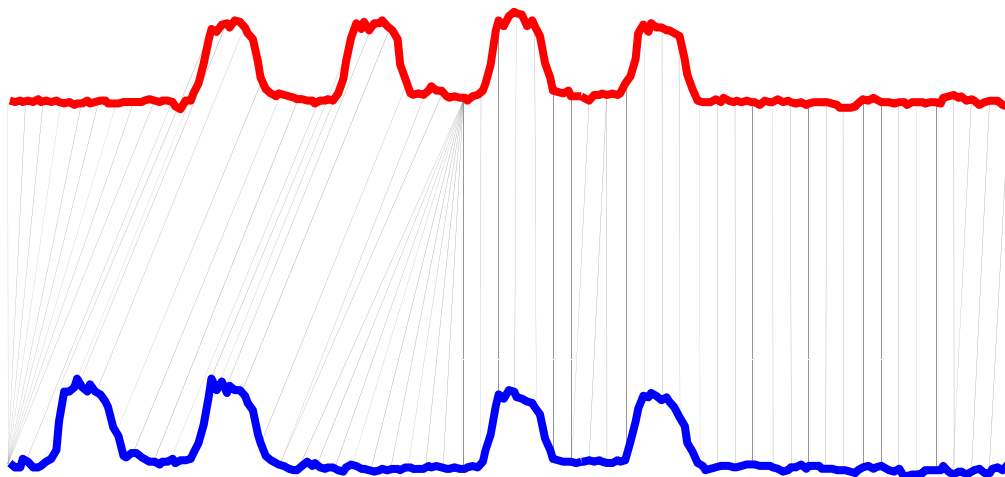
- Dynamic time warping (DTW) is an algorithm for measuring similarity between two sequences which may vary in time or speed.
- In general, DTW is a method that allows a computer to find an optimal match between two given sequences (e.g. time series) with certain restrictions.
- The sequences are "warped" non-linearly in the time dimension to determine a measure of their similarity independent of certain non-linear variations in the time dimension.
- One example of the restrictions imposed on the matching of the sequences is on the monotonicity of the mapping in the time dimension. Continuity is less important in DTW than in other pattern matching algorithms; DTW is an algorithm particularly suited to matching sequences with missing information, provided there are long enough segments for matching to occur.
- The extension of the problem for two-dimensional "series" like images (planar warping) is NP-complete, while the problem for one-dimensional signals like time series can be solved in polynomial time.

Euclidean Vs Dynamic Time Warping



Euclidean Distance

Sequences are aligned "one to one".



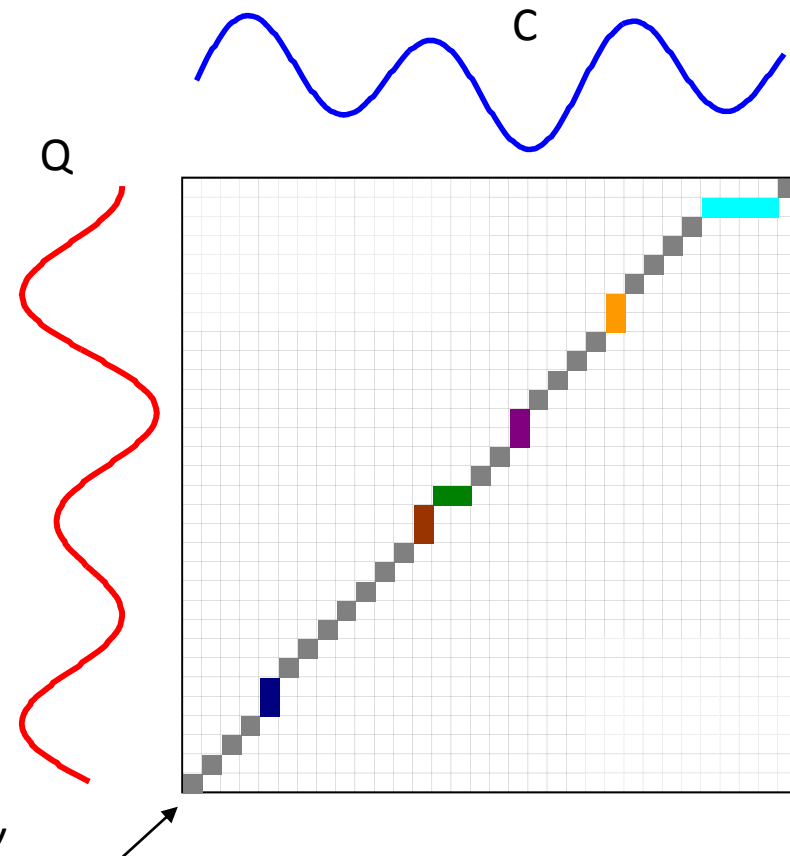
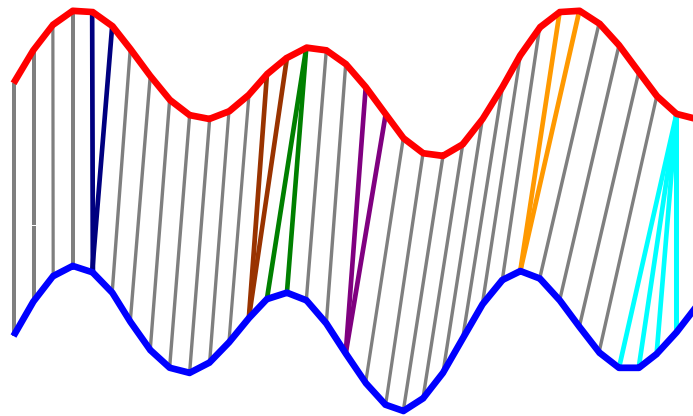
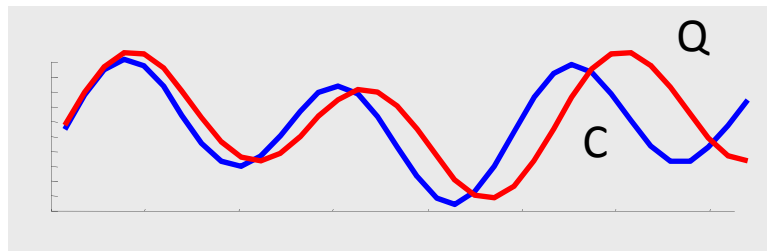
"Warped" Time Axis

Nonlinear alignments are possible.

How is DTW Calculated?

$$DTW(Q, C) = \min \left\{ \sqrt{\sum_{k=1}^K w_k} / K \right\}$$

$$\gamma(i, j) = d(q_i, c_j) + \min \{ \gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1) \}$$



DWT Optimization

The major optimisations to the DTW algorithm arise from observations on the nature of good paths through the grid. These are outlined in Sakoe and Chiba and can be summarised as:

- *Monotonic condition*: the path will not turn back on itself, both the i and j indexes either stay the same or increase, they never decrease.
- *Continuity condition*: The path advances one step at a time. Both i and j can only increase by 1 on each step along the path.
- *Boundary condition*: the path starts at the bottom left and ends at the top right.
- *Adjustment window condition*: a good path is unlikely to wander very far from the diagonal. The distance that the path is allowed to wander is the window length r .
- *Slope constraint condition*: The path should not be too steep or too shallow. This prevents very short sequences matching very long ones. The condition is expressed as a ratio n/m where n is the number of steps in the x direction and m is the number in the y direction. After n steps in x you must make a step in y and vice versa.

Hough Transforms (HT)

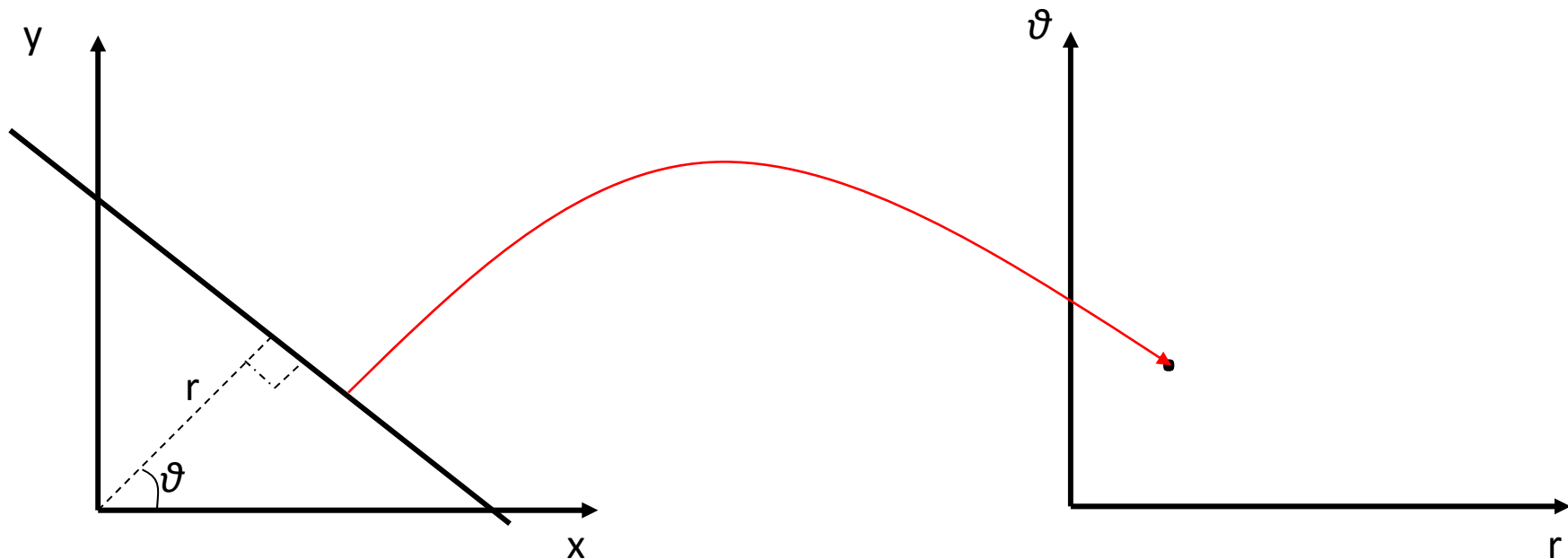
- The HT is capable of identifying digital straight line segment in cluttered and/or noisy images, and also give an estimate of their respective parameters.
- There are many HT variations; here we refer to the traditional approach, reported in Hough's patent
 - The transform acts as a mapping from the image space into a parameter space
- The Hough transform was initially developed to detect analytically defined shapes (e.g., lines, circles, ellipses etc.).

HT – Continuous case

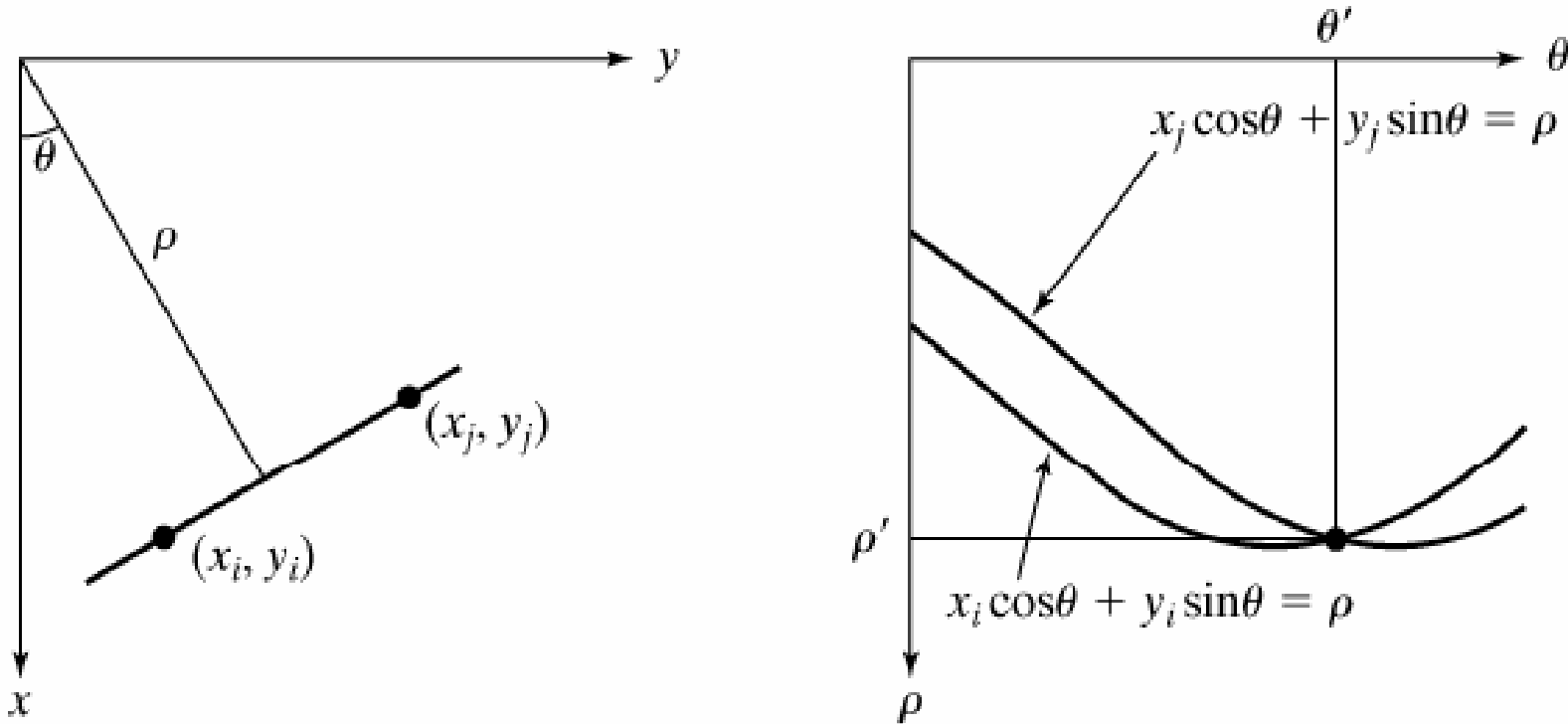
- Basic principle easy to understand if we consider both the image and the parameter spaces as being continuous
- Parameters:
 - The normal distance between the straight line and the coordinate origin (r)
 - the normal angle (θ) defined between the line normal and the x-axis
- A straight line corresponds to the set of points in the plane satisfying the equation $r = \underline{x} \cdot \cos(\theta) + \underline{y} \cdot \sin(\theta)$

HT – Continuous case (cont)

$$x \cos \vartheta + y \sin \vartheta = r$$



HT – Continuous case (cont)



A straight line in the continuous image space is mapped into a continuous collection of sinusoids, in the continuous normal parameter space. All such sinusoids intercept at the single point (θ', ρ') corresponding to the parameters of the original line L .

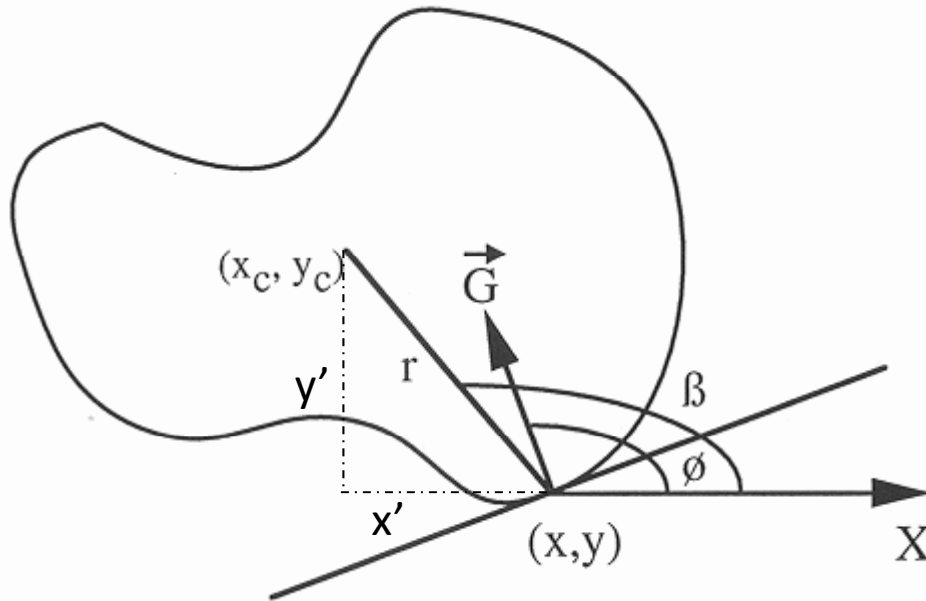
Generalized Hough Transform

- The generalized Hough transform can be used to detect arbitrary shapes (i.e., shapes having no simple analytical form).
- The essential idea is that, instead of using the parametric equation of the curve, we use a look-up table to define the relationship between the boundary coordinates and orientation, and the Hough parameters.
- The look-up table values must be computed during a training phase using a prototype shape

Generalized Hough Transform (cont)

- Suppose we know the scale and orientation of the required object (special case).
- The first step is to select an arbitrary reference point (x_c, y_c) in the object (e.g., the center of mass)
- Then the shape of the object is described in terms of:
 - distance between a boundary point and the reference point.
 - angle between the line defined by the boundary point and the reference point and the tangent at boundary point

GHT (cont)



$$x = x_c + x' = x_c + r \sin(\pi - \beta) = x_c - r \cos(\beta)$$

$$y = y_c + y' = y_c + r \cos(\pi - \beta) = y_c - r \sin(\beta)$$

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$$

$$\beta = \tan^{-1} \left(\frac{y - y_c}{x - x_c} \right)$$

R-table

$\phi_1 = 0$	$(r, \beta)_{1_1}$	$(r, \beta)_{1_2}$	\cdots	$(r, \beta)_{1_{n_1}}$
\cdots	\cdots	\cdots	\cdots	\cdots
ϕ_j	$(r, \beta)_{j_1}$	$(r, \beta)_{j_2}$	\cdots	$(r, \beta)_{j_{n_1}}$
\cdots	\cdots	\cdots	\cdots	\cdots
$\phi_k = \pi$	$(r, \beta)_{k_1}$	$(r, \beta)_{k_2}$	\cdots	$(r, \beta)_{k_{n_1}}$

GHT: object detection

(1) Quantize the parameter space:

$$P[x_{c_{\min}} \cdots x_{c_{\max}}][y_{c_{\min}} \cdots y_{c_{\max}}]$$

(2) for each edge point (x, y)

(2.1) Using the gradient angle ϕ , retrieve from the *R-table* all the (α, r) values indexed under ϕ .

(2.2) For each (α, r) , compute the candidate reference points:

$$\begin{aligned}x_c &= x + r \cos(\alpha) \\ y_c &= y + r \sin(\alpha)\end{aligned}$$

(2.3) Increase counters (voting):

$$++(P[x_c][y_c])$$

(3) Possible locations of the object contour are given by local maxima in $P[x_c][y_c]$

- If $P[x_c][y_c] > T$, then the object contour is located at (x_c, y_c)

We have assumed that we know the orientation and the scale of the shape!

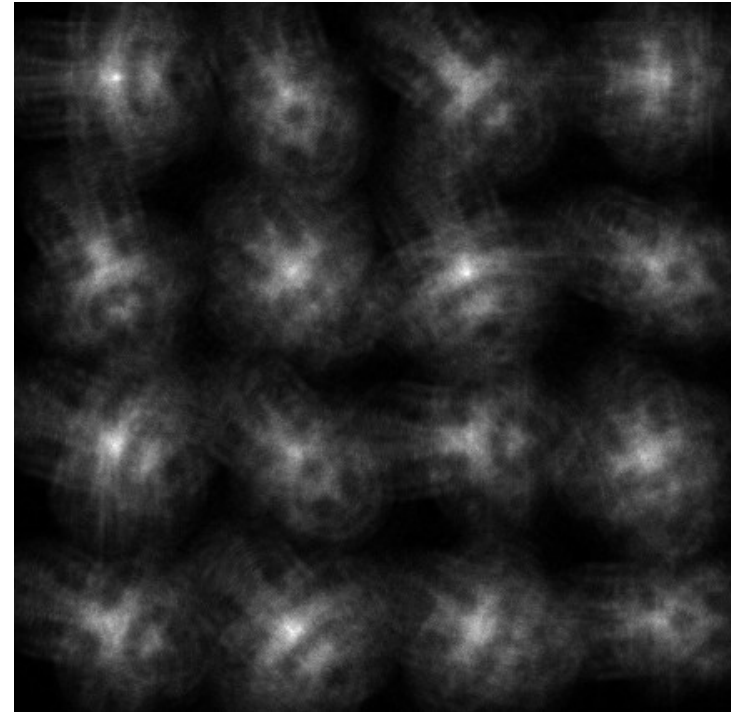
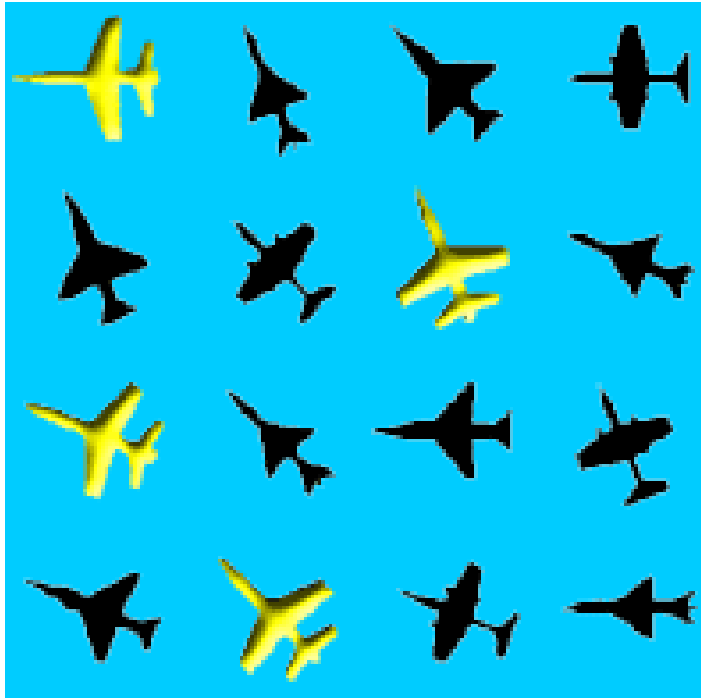
GHT – object localization

- If we do not know orientation and scale of the shape, we have to extend the accumulator by incorporating two extra parameters (α, s) to take changes in orientation and scale into consideration.
- Thus we should use a four-dimensional accumulator indexed (x_c, y_c, α, s) (quantized) and the following formula

$$x_c = x + s \cdot r \cdot \cos(\beta + \alpha)$$

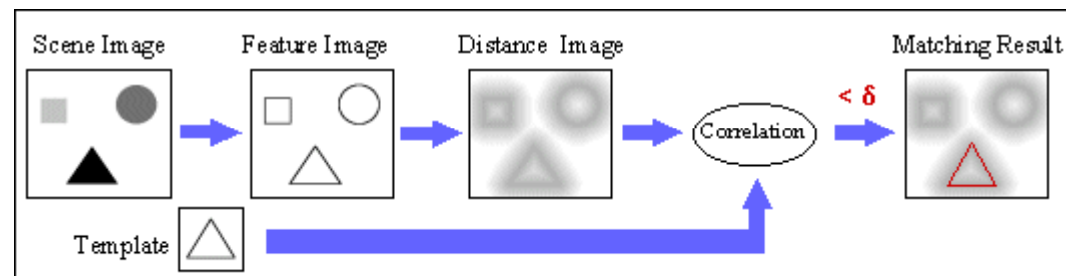
$$y_c = y + s \cdot r \cdot \sin(\beta + \alpha)$$

Example – generic shape

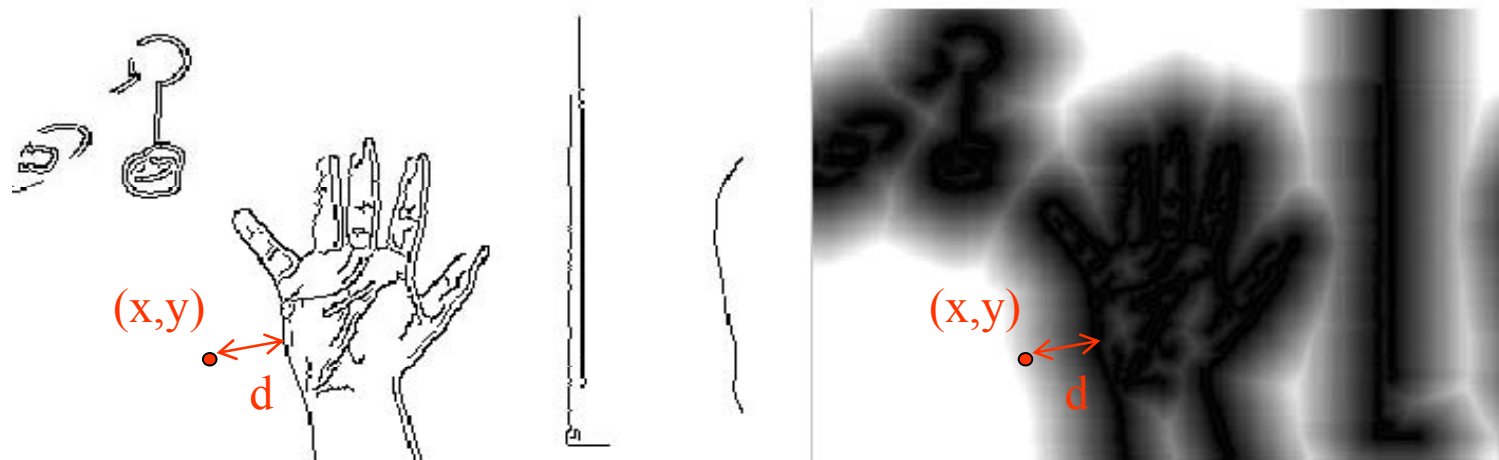


Chamfer System for shape-based object detection

- It covers the detection of arbitrary-shaped objects, whether parametrized (e.g. rectangles and ellipses) or not (e.g. pedestrian outlines).



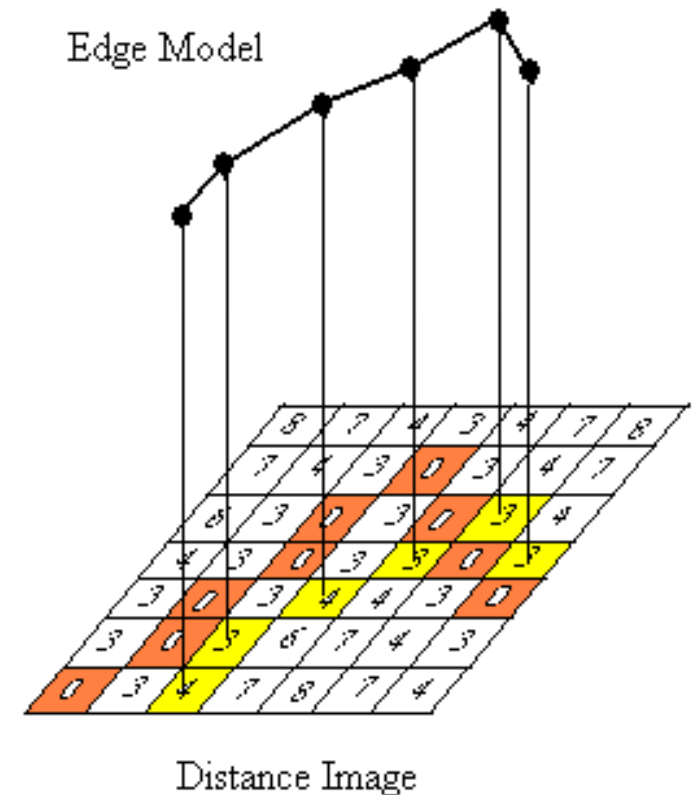
Distance Transform



- Distance image gives the distance to the nearest edge at every pixel in the image
- Calculated only once for each frame

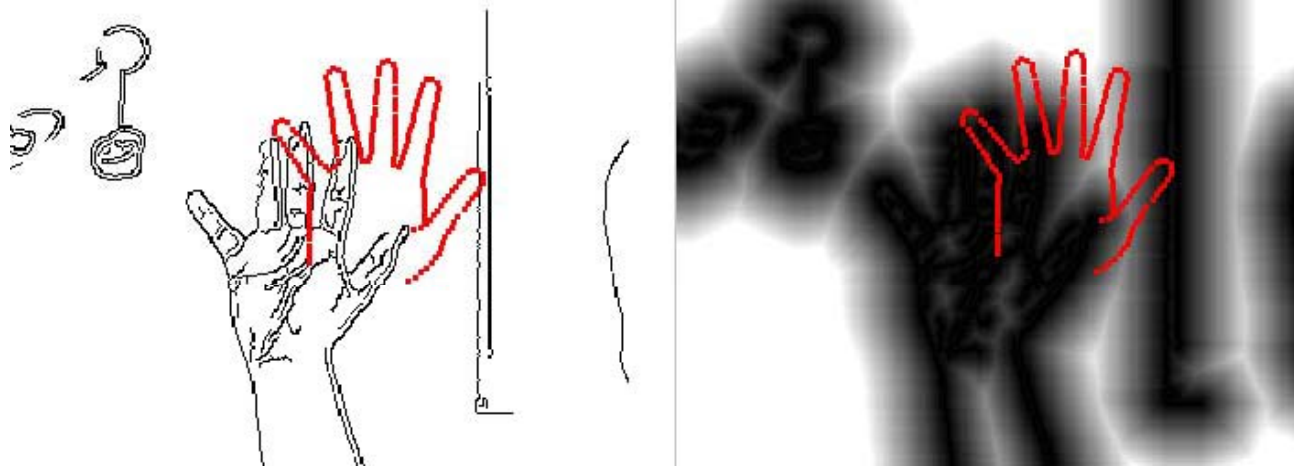
Chamfer Matching

- Edge-model translated over Distance Image.
- At each translation, edge model superimposed on distance image.
- Average of distance values that edge model hits gives Chamfer Distance.



Chamfer Distance = 1.12

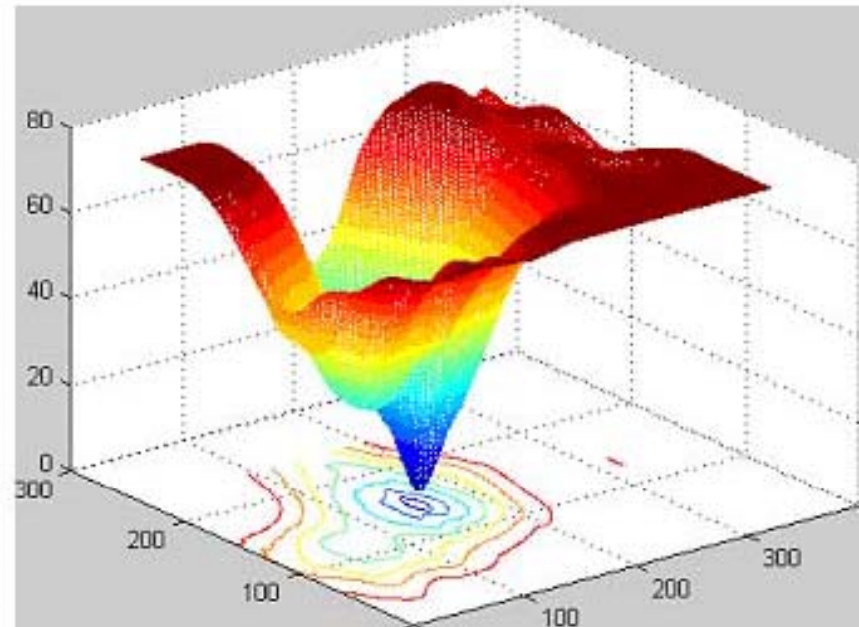
Chamfer Matching



- Chamfer score is the average of distance transform values related to the template position

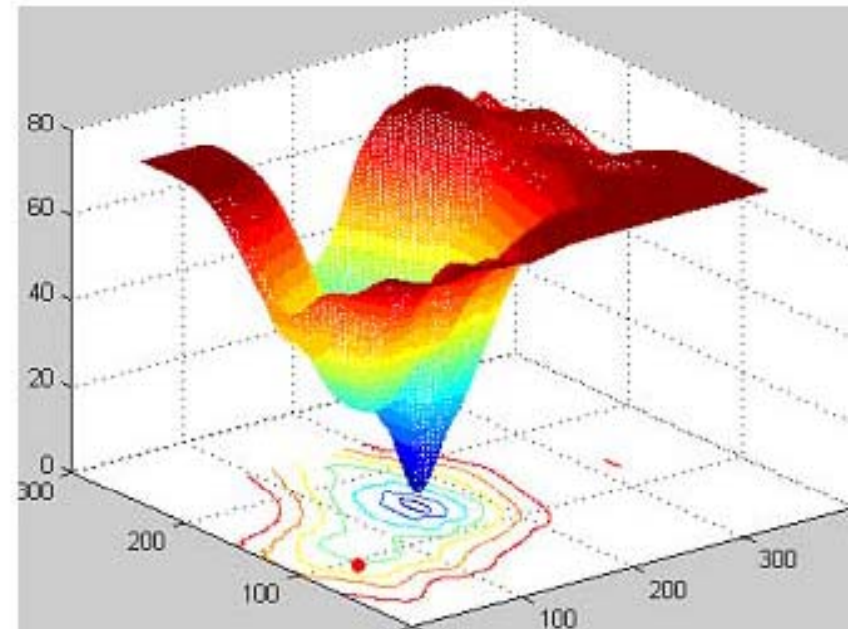
Chamfer Matching

- Distance image provides a smooth cost function

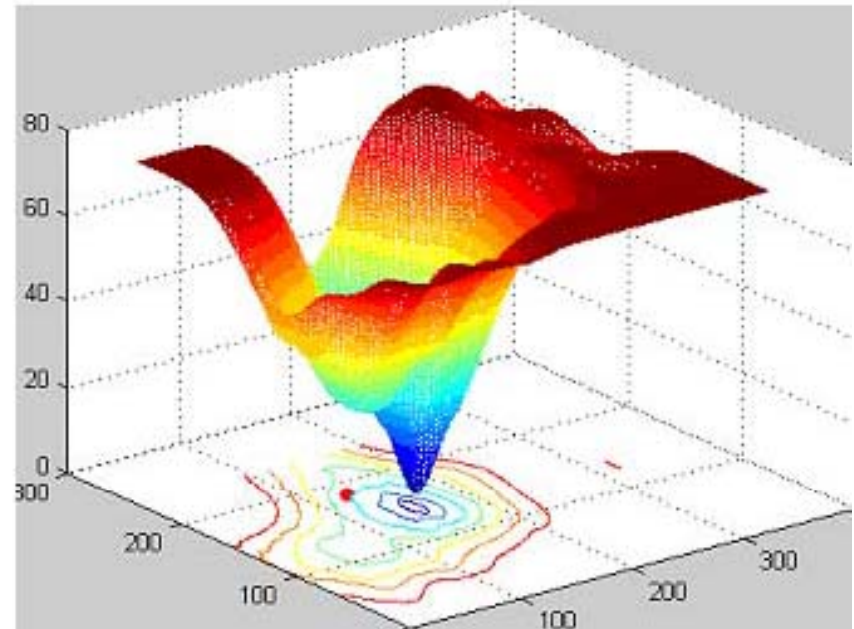
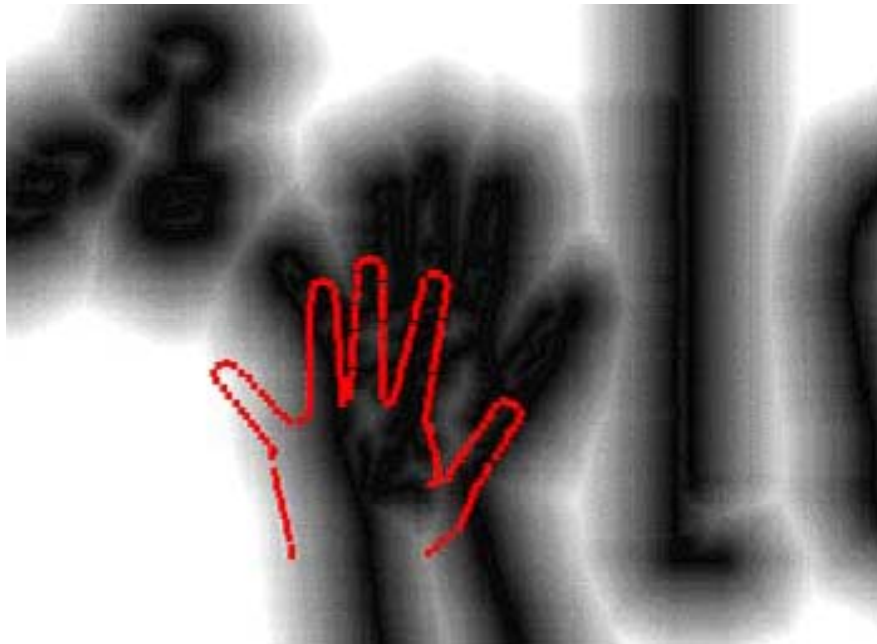


- Efficient searching techniques can be used to find correct template

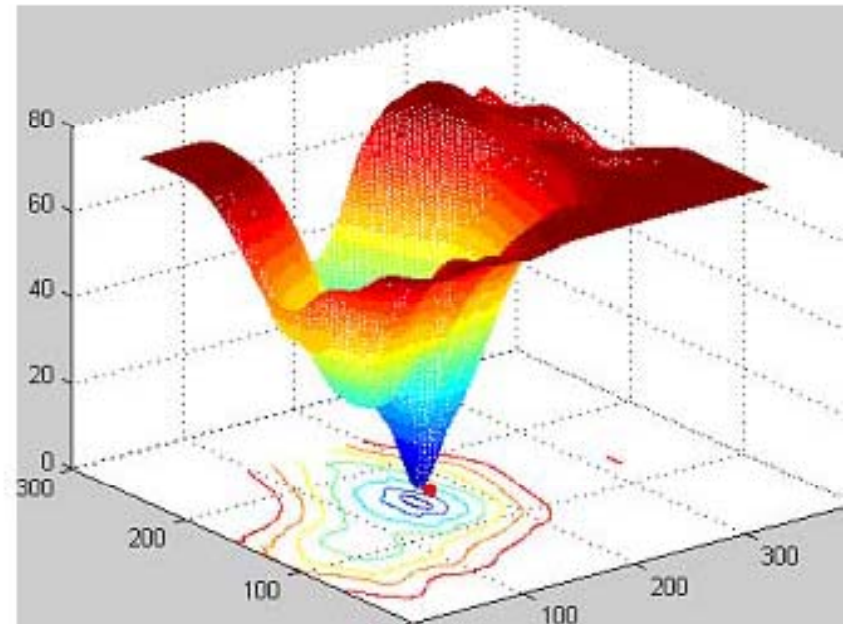
Chamfer Matching



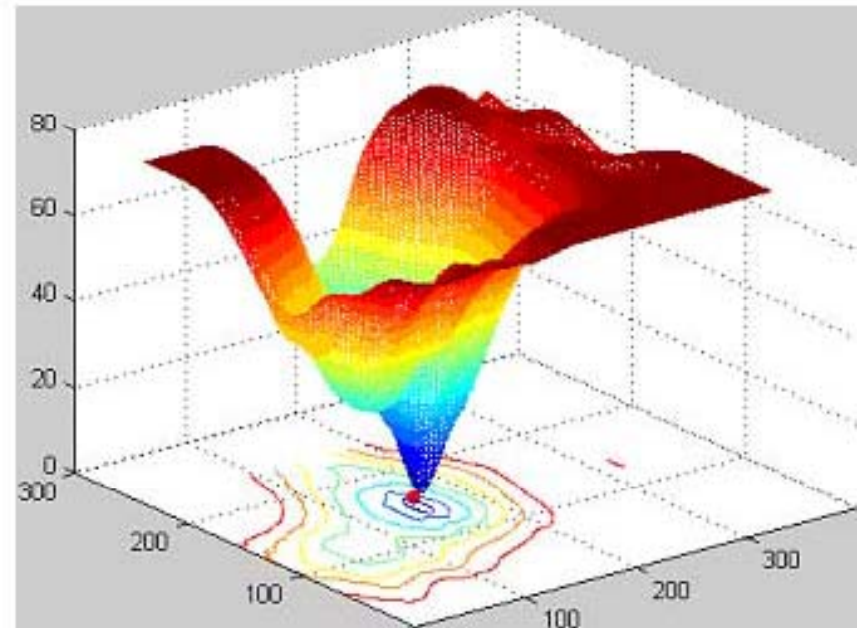
Chamfer Matching



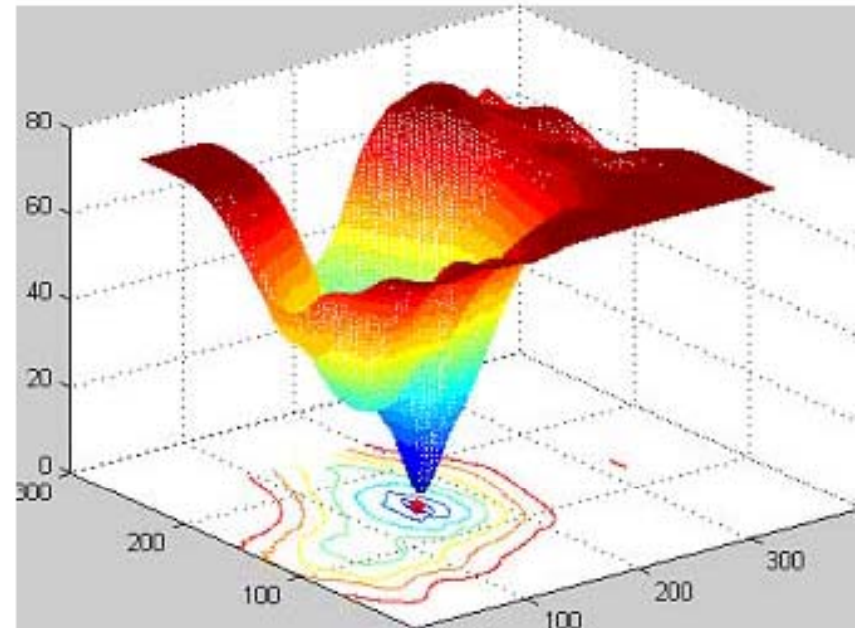
Chamfer Matching



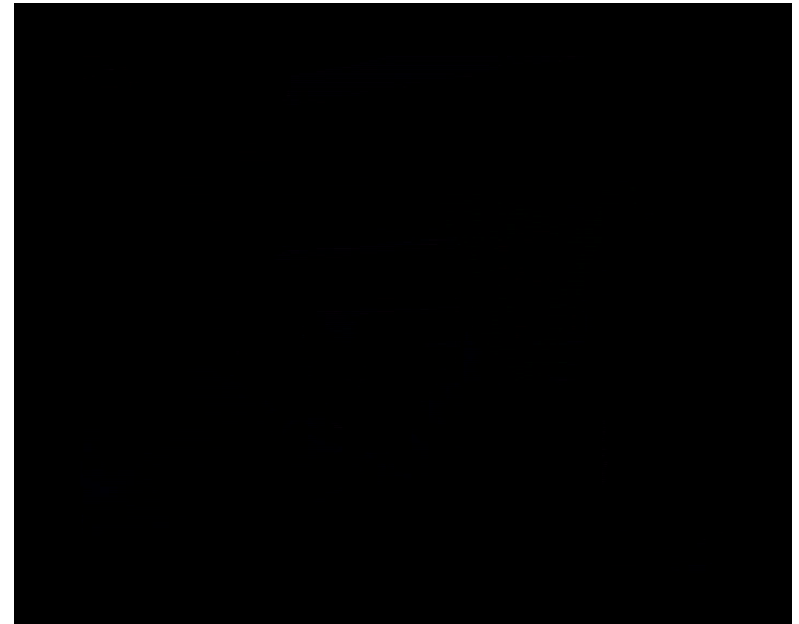
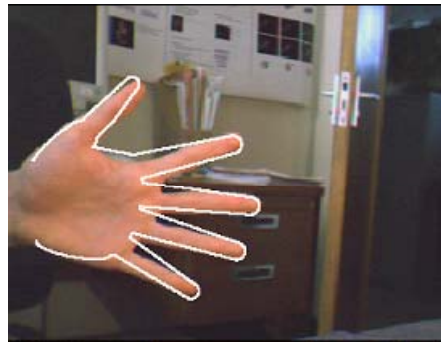
Chamfer Matching



Chamfer Matching



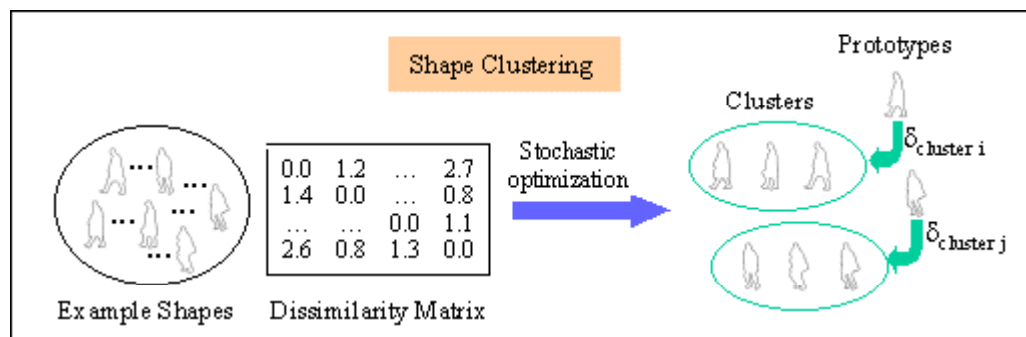
Hand Detection/Recognition



Considerations

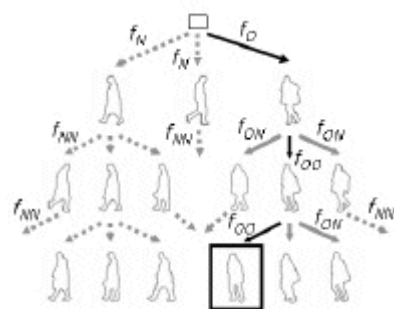
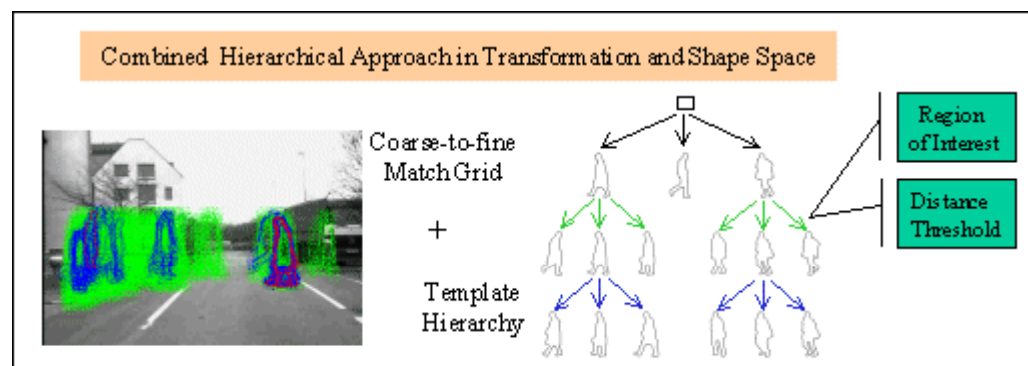
- Chamfer Distance
 - Variant to scale and rotation
 - Sensitive to small shape changes
 - Need large number of template shapes
- But**
- Robust to clutter
 - Computationally cheap

Chamfer Based Pedestrian Detection



The approach chosen groups similar templates together and represents each group by two entities: a “prototype” template and a distance parameter.

When applied recursively, shape-clustering leads to a template tree.



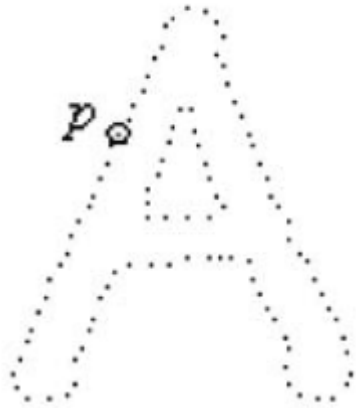
in 1999



Shape Context

- Belongie et al. [2002] proposed a correspondence-based shape matching method using shape contexts
- It extracts a global feature, “the shape context”, for each corresponding point
- The matching between corresponding points is then the matching between the context features

Shape context (cont)



- Let's consider a character shape
- Then extract the edges and sample them
- To extract the shape context at a point p , the vectors of p to all the other boundary points are found

Shape context (cont)

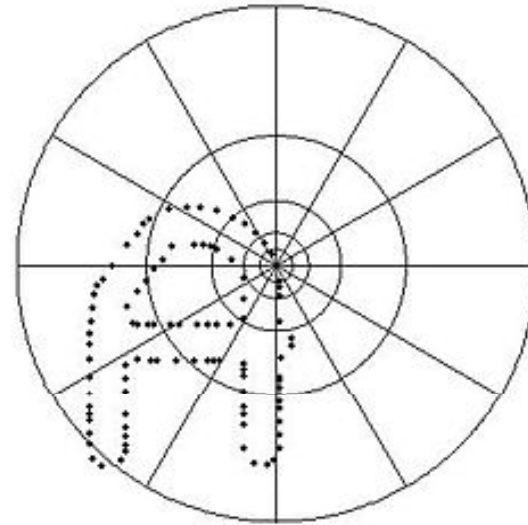
- For a point p_i on the shape a histogram h_i of the relative coordinates of the remaining $n-1$ points is computed :

$$h_i(k) = \#\{q \neq p_i : (q - p_i) \in \text{bin}(k)\}$$

- This histogram is defined to be the shape context of p_i

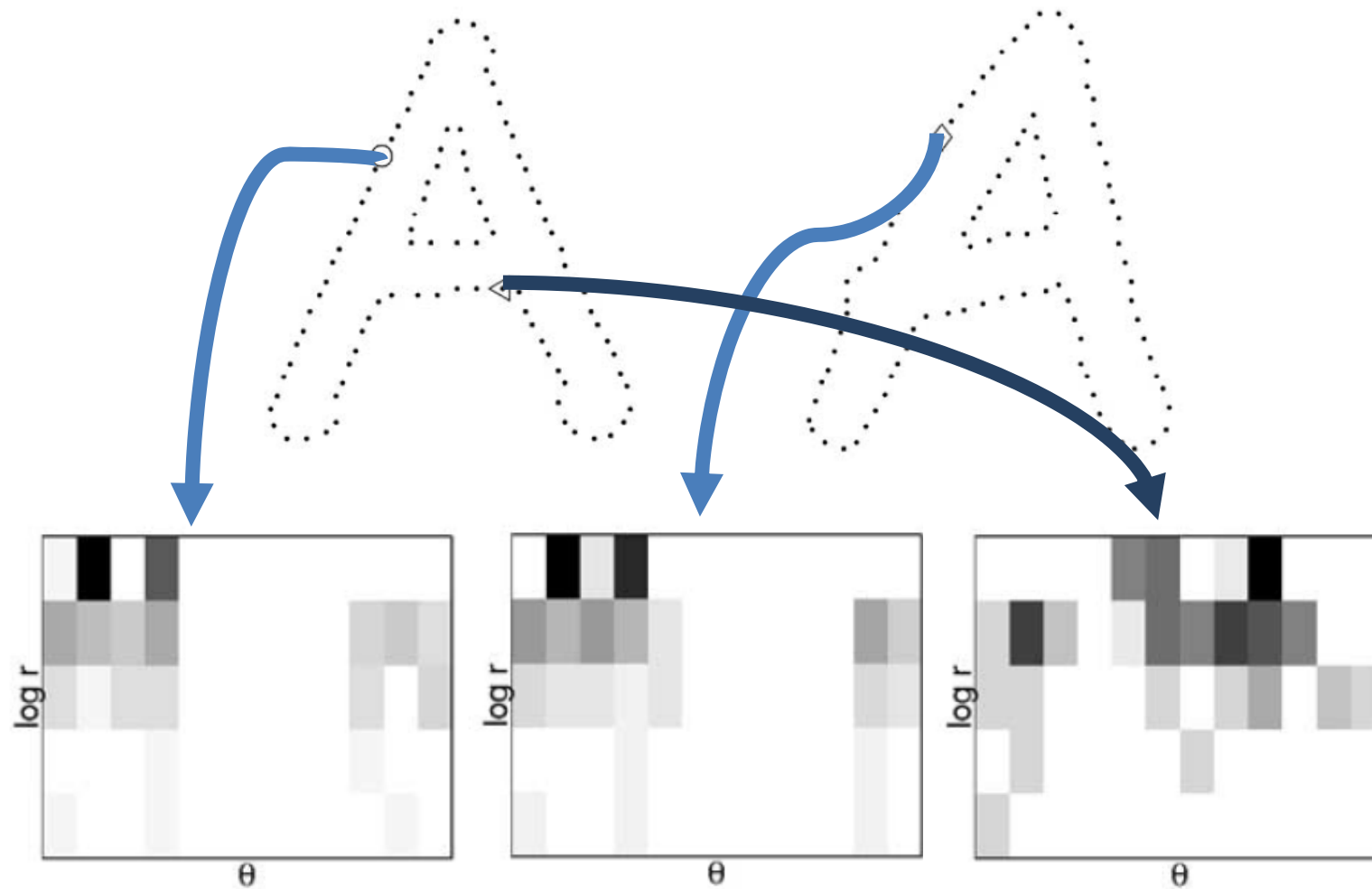
Shape context (cont)

- Vectors are put into log-polar space to make the histogram more sensitive to positions of nearby points than to those of points farther away
- This choice corresponds to a linearly increasing positional uncertainty with distance from p_i , reasonable if the transformation between the shapes around p_i can be locally approximated as affine.



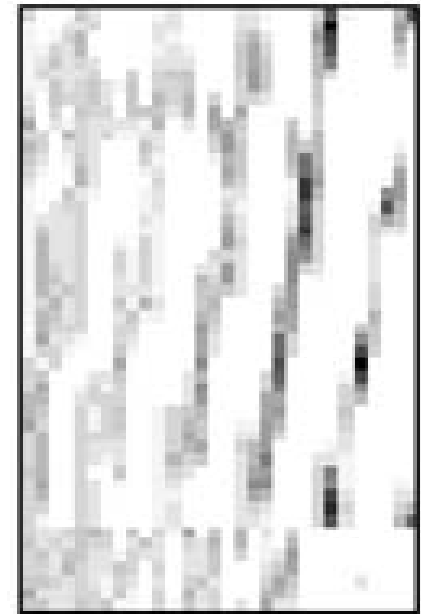
- 5 bins are used for $\log r$
- 12 bins are used for θ

Shape context - examples



Shape context - matching

- The histogram of each point is flattened and concatenated to form the context of the shape
- The matching of two shapes is done by matching two context maps of the shapes, which is a matrix-based matching.
- It minimizes the total cost of matching between one context matrix and all the permutations of another context matrix.

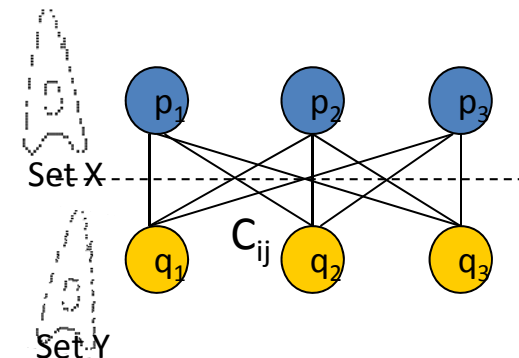


Shape context – matching (cont)

- Consider a point p_i on the first shape and a point q_j on the second shape. Let $C_{ij} = C(p_i, q_j)$ denote the cost of matching these 2 points.
- Shape context are distributions represented as histogram, then we can use the χ^2 test statistic:

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

where $h_i(k)$ and $h_j(k)$ denote the K -bin normalized histogram at p_i and q_j , respectively

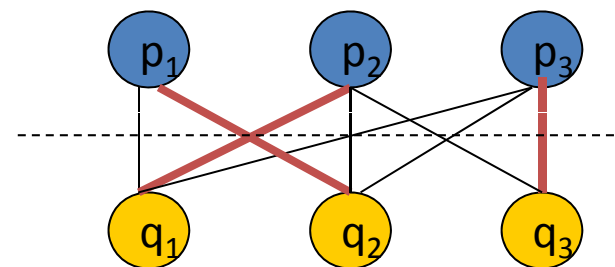


Shape context – matching (cont)

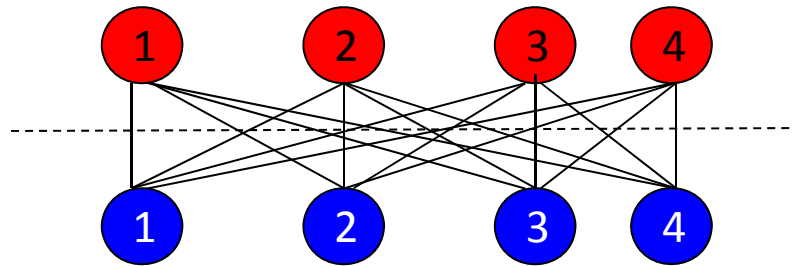
- Given the sets of costs between all pairs of points p_i on the first shape and q_j on the second shape, we want to minimize the total cost of matching

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

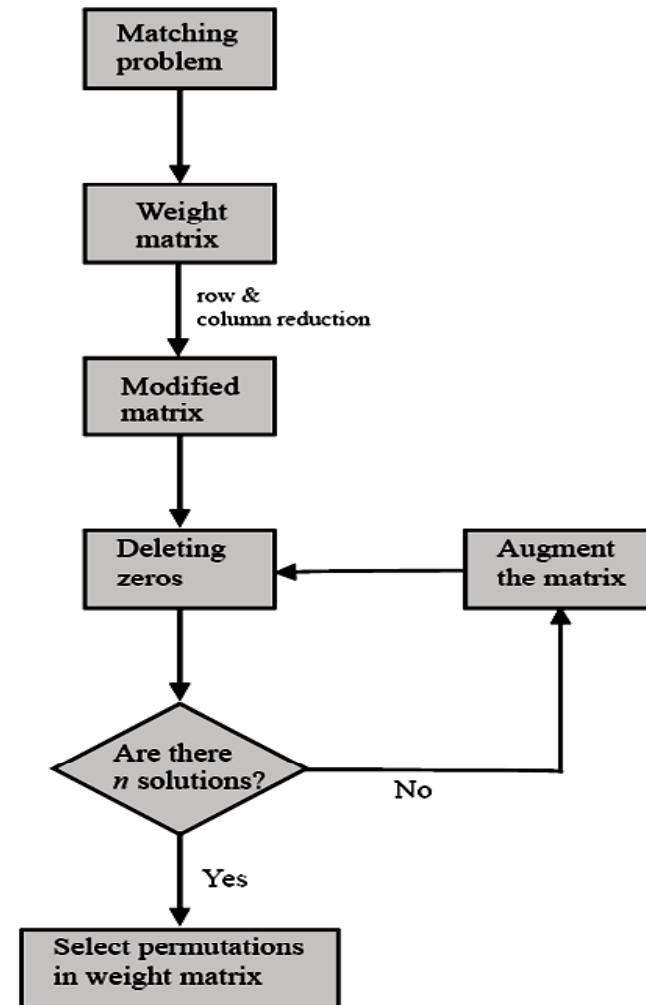
- Constraint: the matching must be one-to-one, *i.e.* π is a permutation
- An algorithm to solve the assignment problem can be used to minimize the cost function (e.g., Hungarian Algorithm, The Shortest Augmenting Path Algorithm, etc).



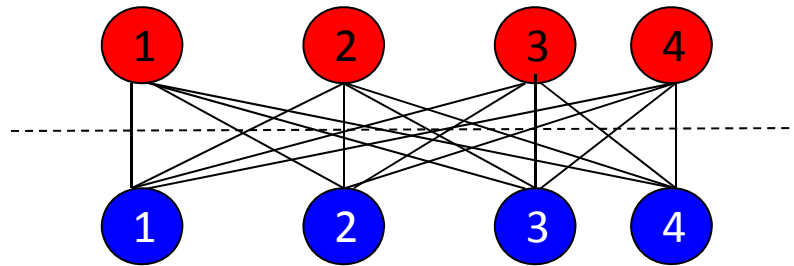
Hungarian Algorithm



	Red1	Red2	Red3	Red4
Blue1	8	17	3	23
Blue2	39	4	11	20
Blue3	13	2	41	6
Blue4	22	8	9	2



Hungarian Algorithm

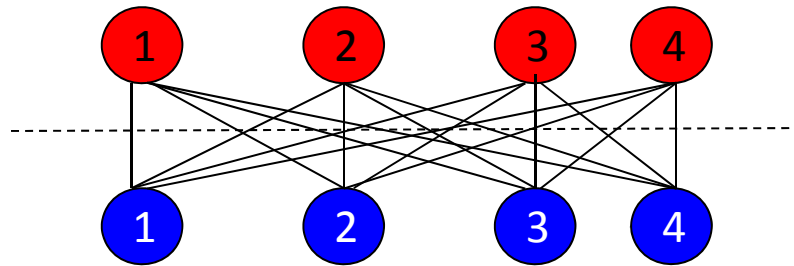


Step 1

First for each row we subtract the row minimum from the rest of the row.

	Red1	Red2	Red3	Red4	
Blue1	8	17	3	23	-3
Blue2	39	4	11	20	-4
Blue3	13	2	41	6	-2
Blue4	22	8	9	2	-2

Hungarian Algorithm

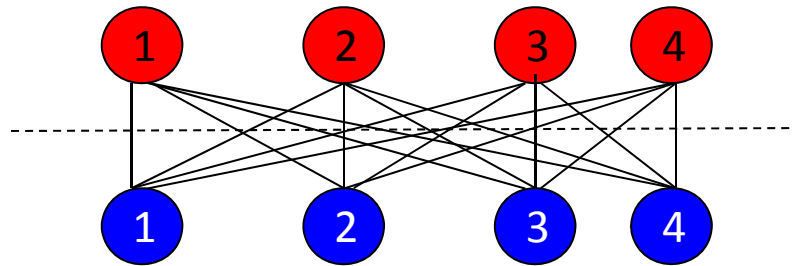


Step 1

First for each row we subtract the row minimum from the rest of the row.

	Red1	Red2	Red3	Red4
Blue1	5	14	0	20
Blue2	35	0	7	16
Blue3	11	0	39	4
Blue4	20	6	7	0

Hungarian Algorithm

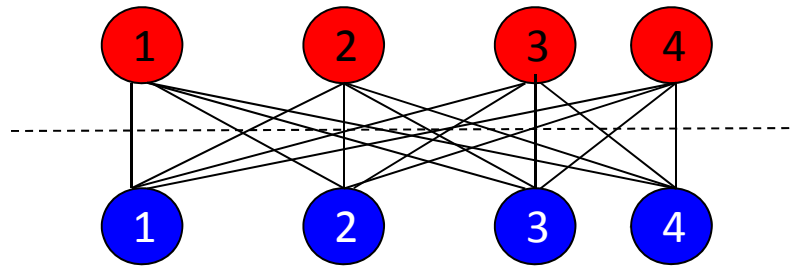


Step 2

Then for each column we subtract the column minimum from the rest of the column.

	Red1	Red2	Red3	Red4
Blue1	5	14	0	20
Blue2	35	0	7	16
Blue3	11	0	39	4
Blue4	20	6	7	0
	-5	-0	-0	-0

Hungarian Algorithm

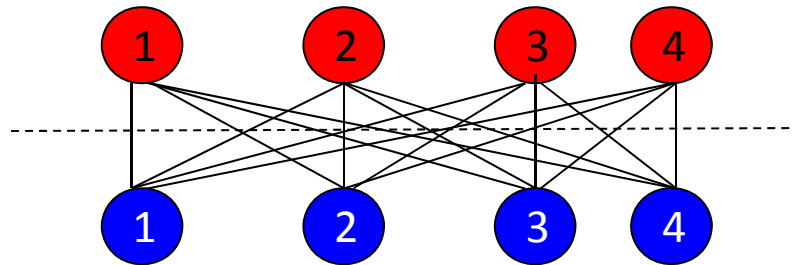


Step 2

Then for each column we subtract the column minimum from the rest of the column.

	Red1	Red2	Red3	Red4
Blue1	0	14	0	20
Blue2	30	0	7	16
Blue3	6	0	39	4
Blue4	15	6	7	0

Hungarian Algorithm

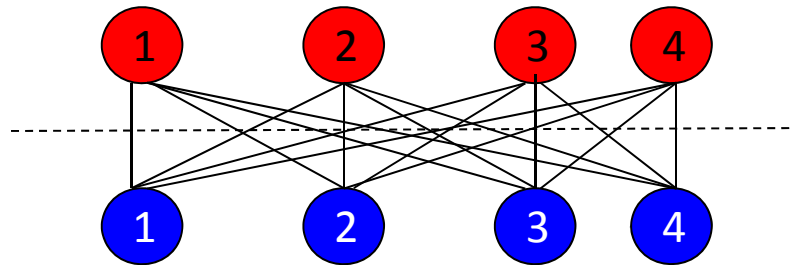


	Red1	Red2	Red3	Red4
Blue1	0	14	0	20
Blue2	30	0	7	16
Blue3	6	0	39	4
Blue4	15	6	7	0

Step 3

Cover the minimum number of rows or columns required to cover all of the zeros in the matrix (shown in light green). If the number of rows or columns required is equal to the number of rows/columns in the matrix then we are finished. Otherwise, we have to proceed to the next step. In this case, we can cover all zeroes with only three lines so we continue.

Hungarian Algorithm



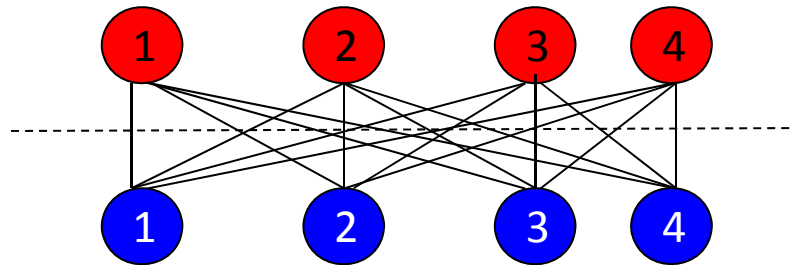
Step 4

From matrix, we first have to find the minimum value not covered (6=[Blue3,Red1]).

This value is then subtracted from each uncovered entry and added to each entry covered by both a vertical and horizontal line ([Blue1,Red2] and [Blue1,Red4]) .

	Red1	Red2	Red3	Red4
Blue1	0	14	0	20
Blue2	30	0	7	16
Blue3	6	0	39	4
Blue4	15	6	7	0

Hungarian Algorithm



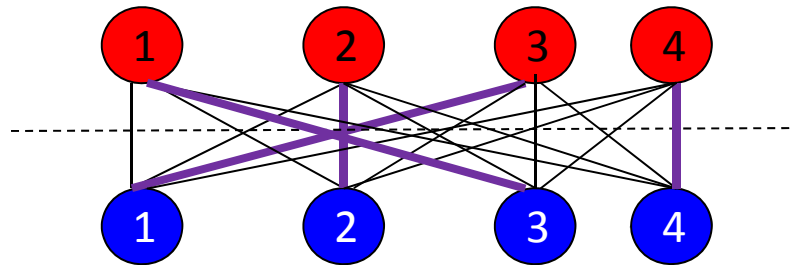
Step 4

From matrix, we first have to find the minimum value not covered (6=[Blue3,Red1]).

This value is then subtracted from each uncovered entry and added to each entry covered by both a vertical and horizontal line ([Blue1,Red2] and [Blue1,Red4]) .

	Red1	Red2	Red3	Red4
Blue1	0	20	0	26
Blue2	24	0	1	16
Blue3	0	0	33	4
Blue4	9	6	1	0

Hungarian Algorithm



	Red1	Red2	Red3	Red4
Blue1	0	20	0	26
Blue2	24	0	1	16
Blue3	0	0	33	4
Blue4	9	6	1	0

Repeat until converge

Step 3 is then repeated. This continues until four rows or columns are needed to cover all zeroes and we are finished.

In this case, the matrix shown on the left is finished. The zeroes in the matrix determine the optimal matching between nodes. The only possible matching is shown by the zeroes in the purple cells. The final node matches are listed below.

Shape Context: invariance

- Invariance to translation is intrinsic to the shape context definition since all measurements are taken with respect to points on the object.
- To achieve scale invariance all radial distances are normalized by the mean distance α between the n^2 point pairs in the shape.
- Shape contexts are inherently insensitive to small perturbations of parts of the shape
- The descriptor is not rotation invariant. Rotation invariance can be obtained treating the tangent vector at each point as the positive x -axis, instead of using the absolute frame.

Shape Context Demo

- [Demo1](#)
- [Demo2](#)

Circular blurred shape model

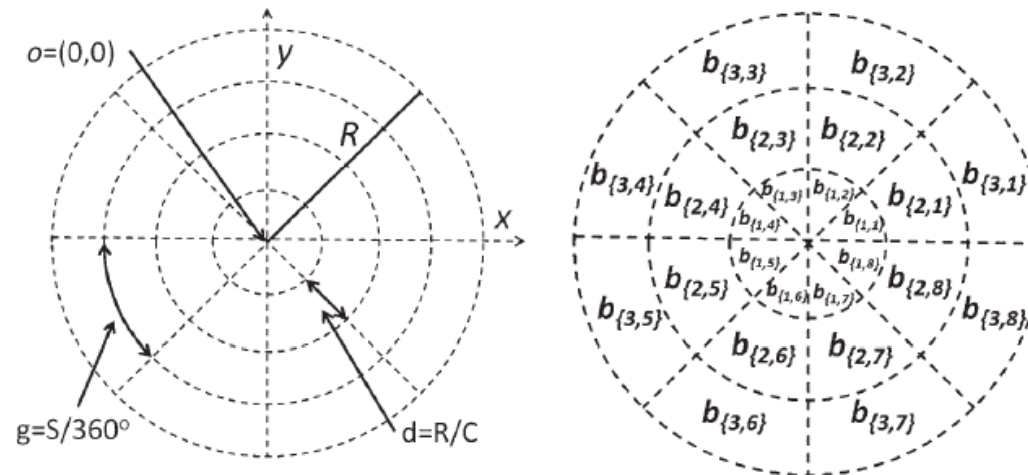
- Feature extraction is performed capturing the spatial arrangement of significant object characteristics in a correlogram structure (from the center of the object region).
- *Shape information from objects is shared* among correlogram regions, being tolerant to the irregular deformations

Correlogram structure (1)

- Correlogram definition: Given a number of circles C , a radius R , a number of sections S , and an image region I , a correlogram

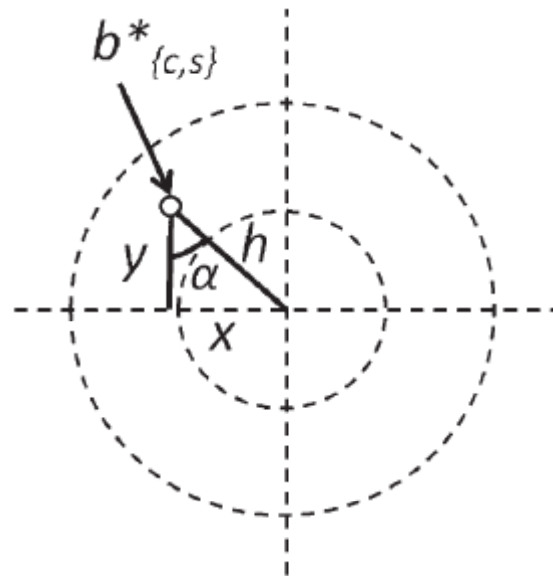
$$B = \{ b\{1,1\}, \dots, b\{C,S\} \}$$

is defined as a radial distribution of sub-regions of the image.

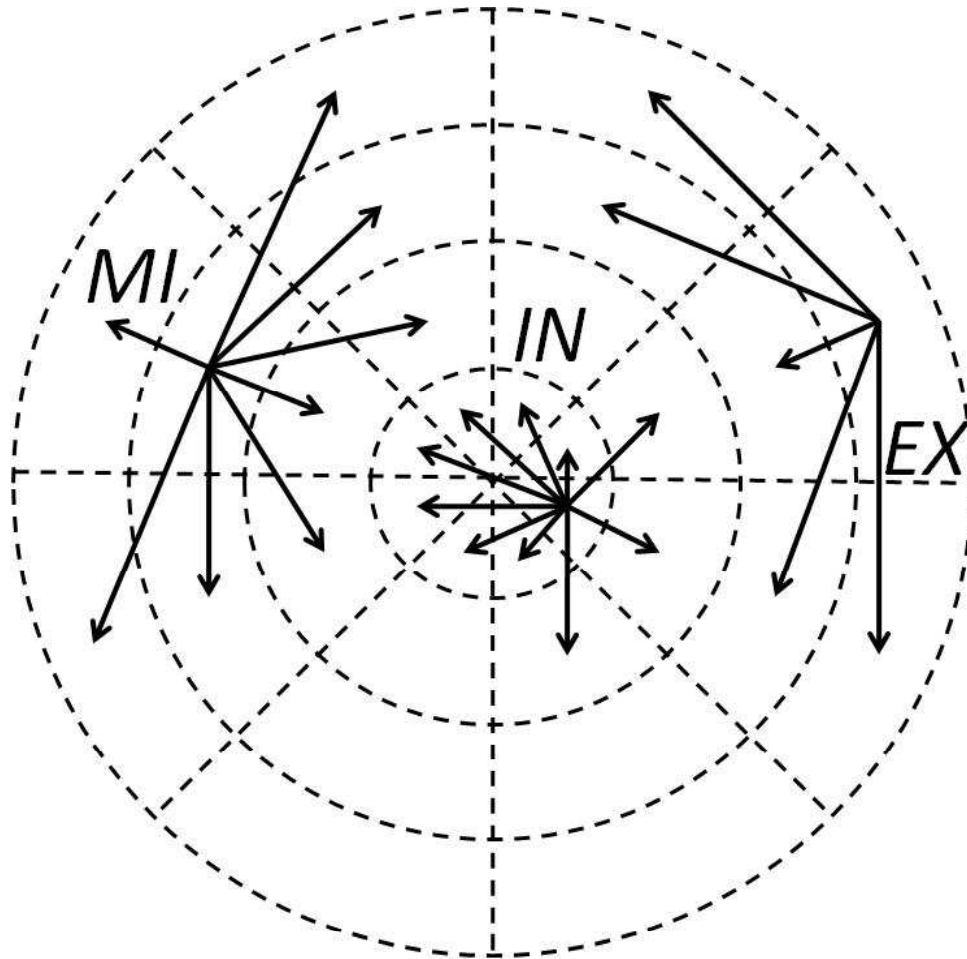


Correlogram structure (2)

- Each region b is defined by its centroid coordinate b^* . Then, the regions around b are defined as the neighbors of b



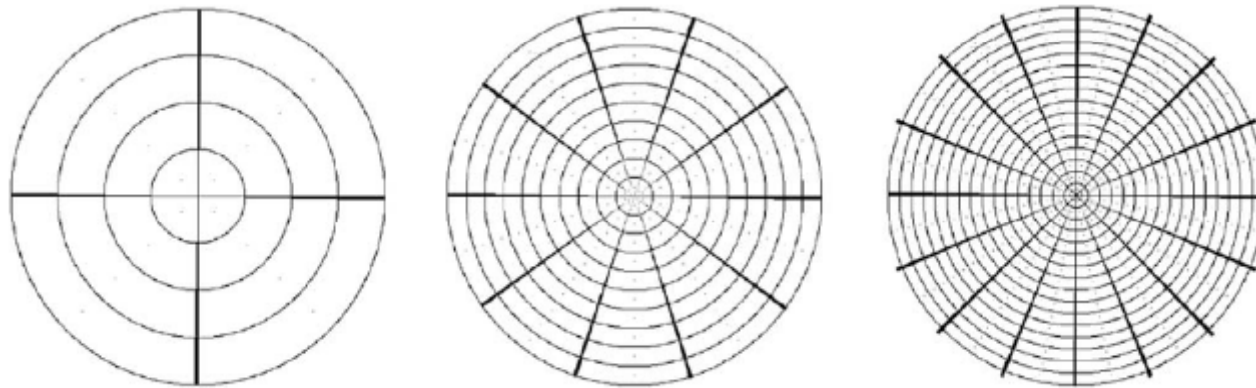
Correlogram structure (3)



The region around b are defined as the neighbors of b .

There is a different number of neighbors depending on the location of b .

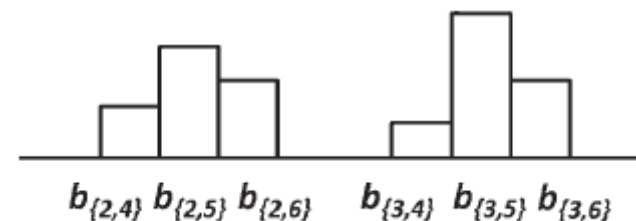
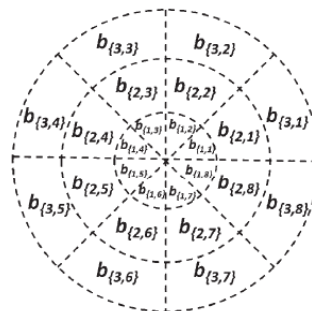
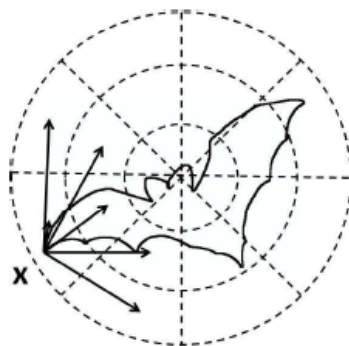
Correlogram structure (4)



Different correlograms structures for different values of C and S.

Descriptor computation

- Given the object contour map, each contour point from the image is considered.
- The distances from the contour point x to the centroids of its corresponding region and neighboring regions are computed. The inverse of these distances is normalized by the sum of the total distances.
- These values are then added to the corresponding positions of the descriptor vector v



Descriptor

- The description is tolerant to irregular deformations
- The length of ν defined by parameters C and S , defines the degree of spatial information taken into account in the description process
- As the number of regions increase, the description becomes more local.
 - An optimal parameters of C and S should be obtained for each particular problem

Rotationally invariant descriptor

- In order to make the description rotationally invariant we look for the main diagonal G_i of correlogram B that maximize the sum of the descriptor values.
- This diagonal is then the reference to rotate the descriptor.
- The orientation in the rotational process, so that G_i is aligned with the x -axis, is that corresponding to the highest description density at both sides of G_i

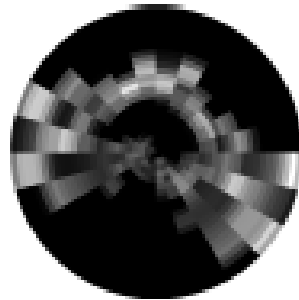
Examples of rotational invariance



Bat1



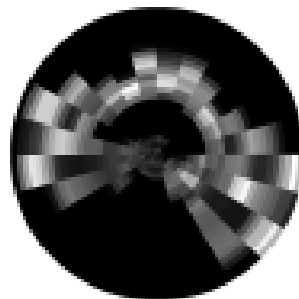
Bat2



Bat1 24×24



Bat1 54×54

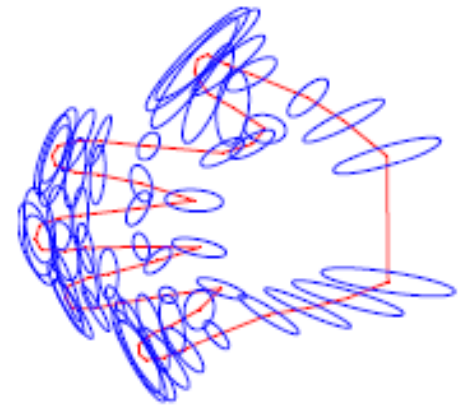
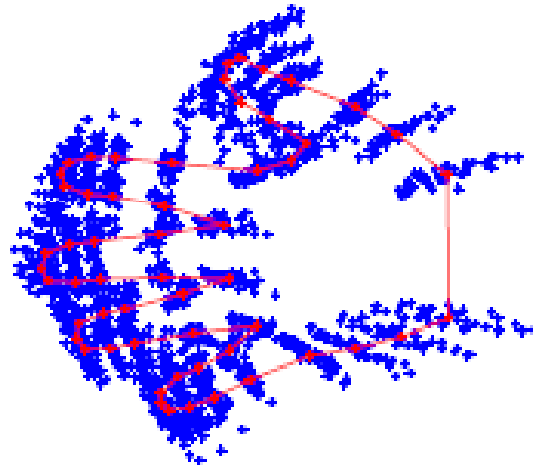
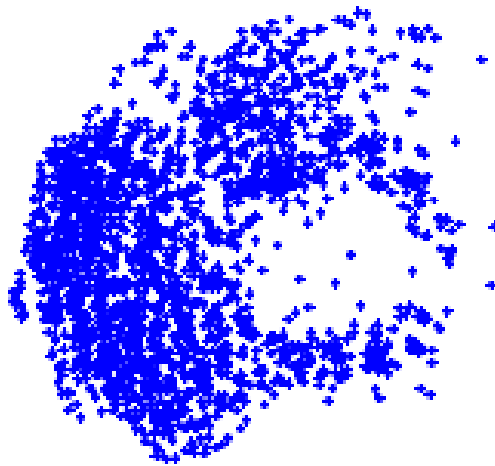
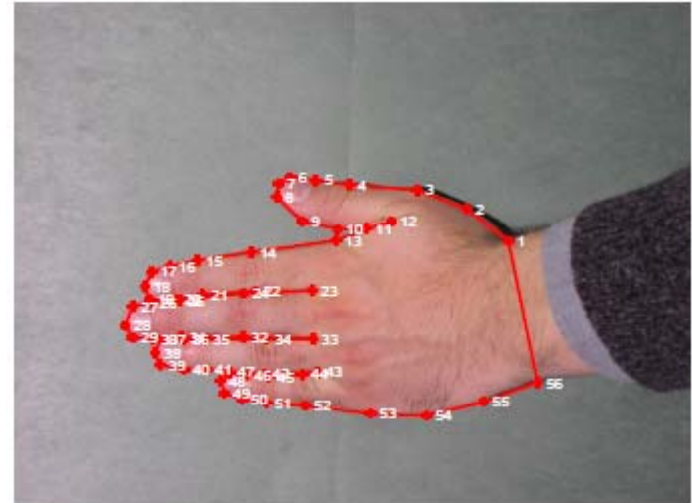


Bat2 24×24



Bat2 54×54

Modeling Shape Variation through PCA



Modeling Shape Variation through PCA



(a) $b_1 = -3\sqrt{\lambda_1}$



(b) $b_1 = 0$



(c) $b_1 = +3\sqrt{\lambda_1}$



(d) $b_2 = -3\sqrt{\lambda_2}$



(e) $b_2 = 0$



(f) $b_2 = +3\sqrt{\lambda_2}$



(g) $b_3 = -3\sqrt{\lambda_3}$



(h) $b_3 = 0$



(i) $b_3 = +3\sqrt{\lambda_3}$

Statistical Shape Analysis

$$\mathbf{s} = [x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_n]^T$$

$$\mathbf{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_m\}$$

$$\mathbf{S}' = \{\mathbf{s}'_1, \dots, \mathbf{s}'_m\}$$

$$\mathbf{S}'_{\Phi} \quad \Phi : R^{2n} \rightarrow R^{n_{\Phi}}$$

$$\mathbf{x} = [x_1, x_2]$$

$$\Phi_{Linear_Kernel}(\mathbf{x}) = \mathbf{x}$$

$$\Phi_{Polynomial_Kernel}(\mathbf{x}) = [x_1^2, x_2^2, \sqrt{2}x_1 x_2]$$

$$\overline{\mathbf{s}'_{\Phi}} = \frac{1}{m} \sum_{i=1}^m \Phi(\mathbf{s}'_i)$$

$$\mathbf{C}_{\Phi} = \frac{1}{m} \sum_{i=1}^m \left[\left(\Phi(\mathbf{s}'_i) - \overline{\mathbf{s}'_{\Phi}} \right) \left(\Phi(\mathbf{s}'_i) - \overline{\mathbf{s}'_{\Phi}} \right)^T \right]$$

$$\mathbf{C}_{\Phi} \mathbf{e}_j^{\Phi} = \lambda_j^{\Phi} \mathbf{e}_j^{\Phi} \quad j = 1, \dots, n_{\Phi}$$

$$\mathbf{e}_j^{\Phi T} \mathbf{e}_j^{\Phi} = 1 \quad j = 1, \dots, n_{\Phi}$$

$$\{\mathbf{e}_j^{\Phi}\}_{j=1}^{n_{\Phi}}$$

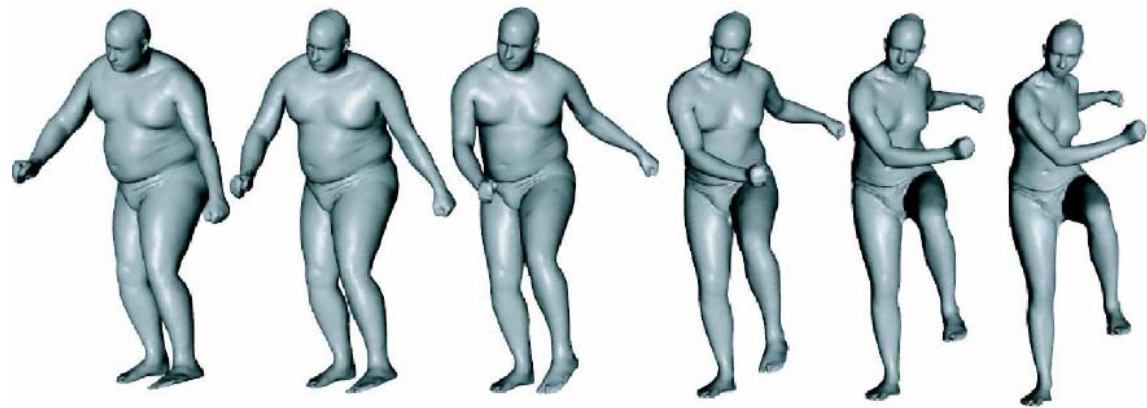
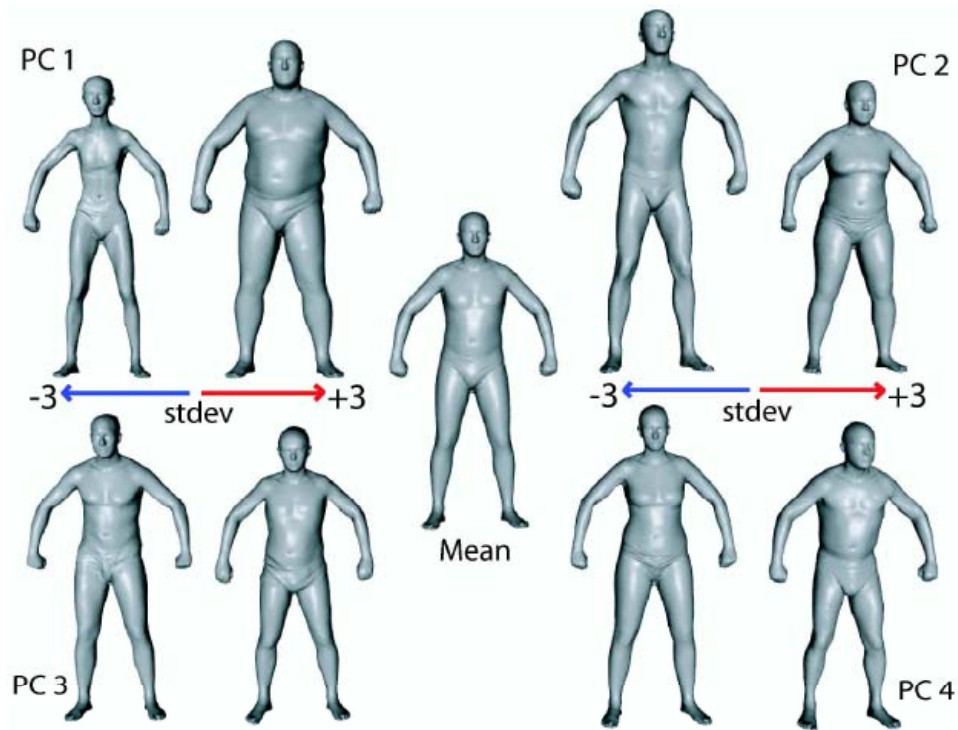
$$\Phi(\mathbf{s}'_i) = \overline{\mathbf{s}'_{\Phi}} + \sum_{j=1}^{n_{\Phi}} a_{i,j}^{\Phi} \mathbf{e}_j^{\Phi}$$

$$a_{i,j}^{\Phi} = \mathbf{e}_j^{\Phi T} \left(\Phi(\mathbf{s}'_i) - \overline{\mathbf{s}'_{\Phi}} \right)$$

Modeling Shape Variation through PCA

- [Demo](#)

SCAPE



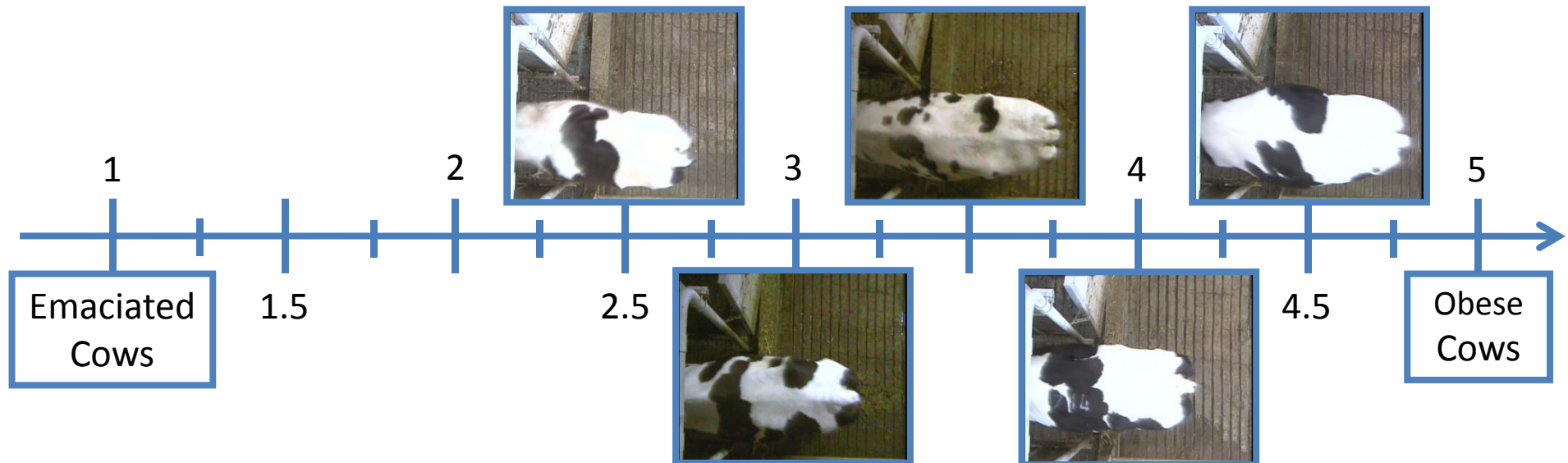
Case study: Objective Estimation of Body Condition Score

Azzaro G., M. Caccamo, J. D. Ferguson,
S. Battiato, G. M. Farinella, G. C. Guarnera,
G. Puglisi, R. Petriglieri and G. Licitra

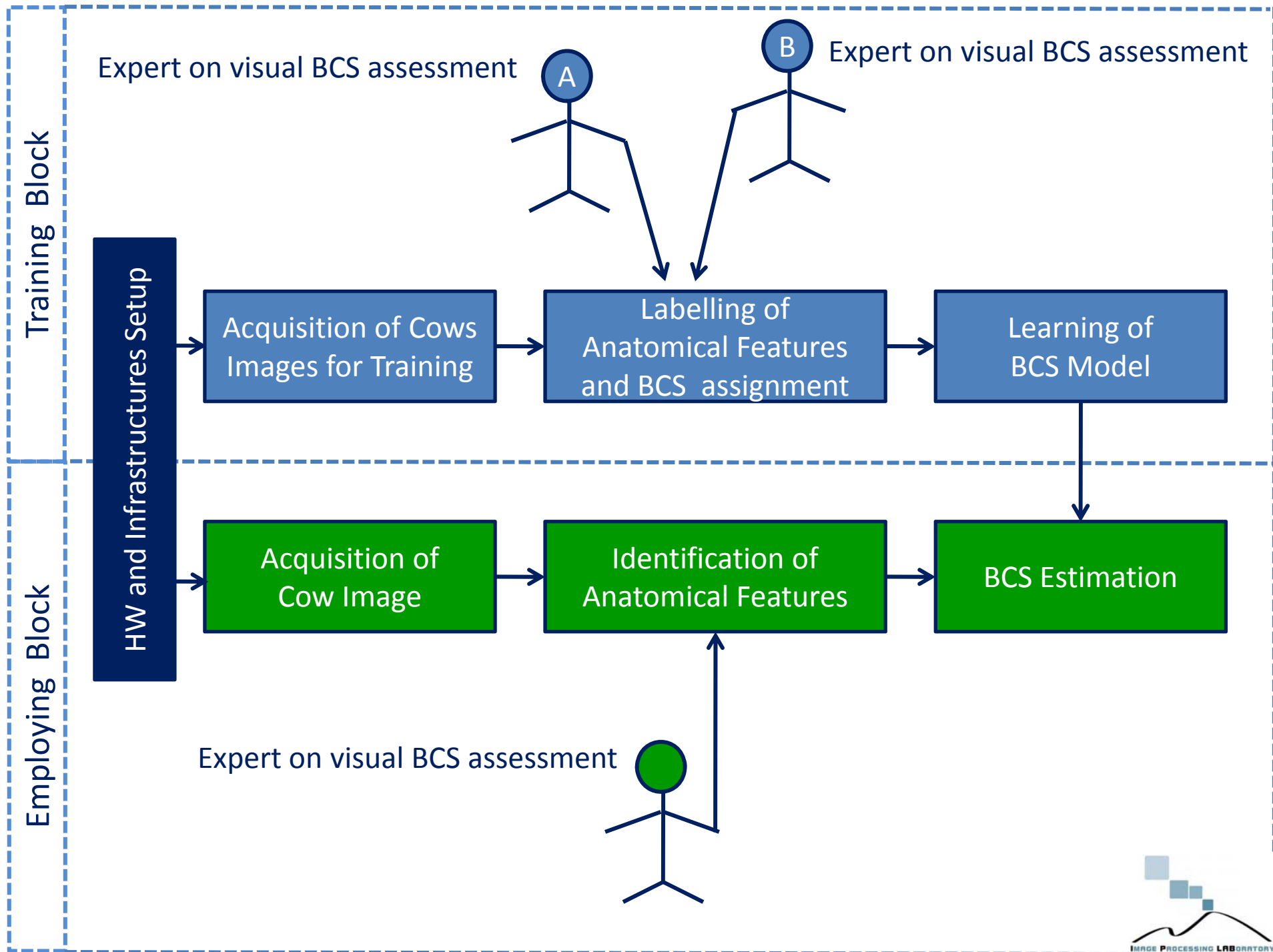


Body Condition Score (BCS)

- It is an important tool for management of dairy cattle
- Used to assess body energy reserves / body condition of dairy cows
- Body condition influences milk production, animal well-being, reproductive performance, and, more generally, farm productivity.
- BCS is estimated through visual or tactile inspection by trained technicians.
- Scale from 1 to 5 (0.25 increments)
- Few dairy farms use it as a feeding management tool: Subjective Estimation, lack of computerised reports, time consuming, costs for training technicians



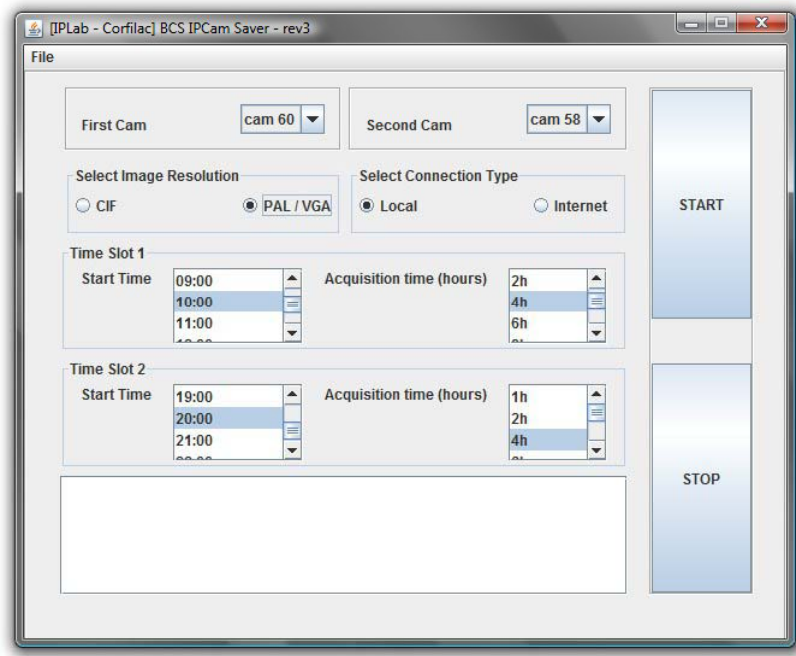
BCS using digital images has been recently investigated in Dairy /Animal Science:
Ferguson et al. 2006, Bewley et al. 2007, Halachmi et al. 2008.



Experimental Setup

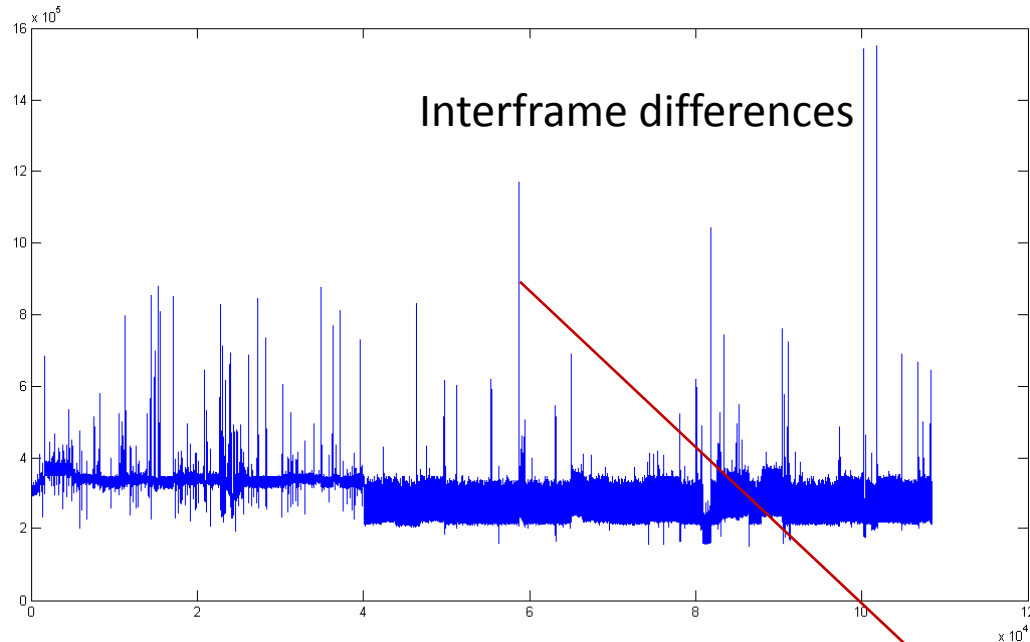


Orthogonal view

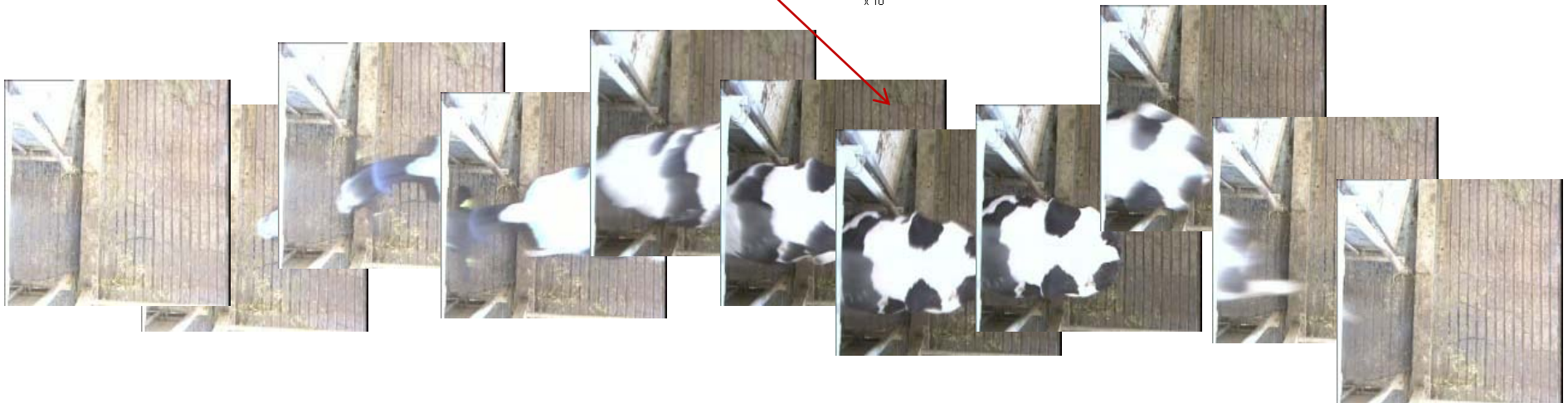


- Network camera specs:
 - Framerate 12 fps
 - Resolution 704x576
- Image acquisition
 - from April 4th through May 6th 2009
 - from 10 a.m. to 2 p.m.
 - from 8 p.m. to 00 a.m.
- The image acquisition system gathered approximately 172800 images for each acquisition interval of four hours to be analyzed.
- The useful information (i.e., the cow is in the scene) was contained in a very small subset (about 40).
- We developed an ad-hoc image processing pipeline to select images to be used in the experiments.

Automatic Selection of Cow images (1)

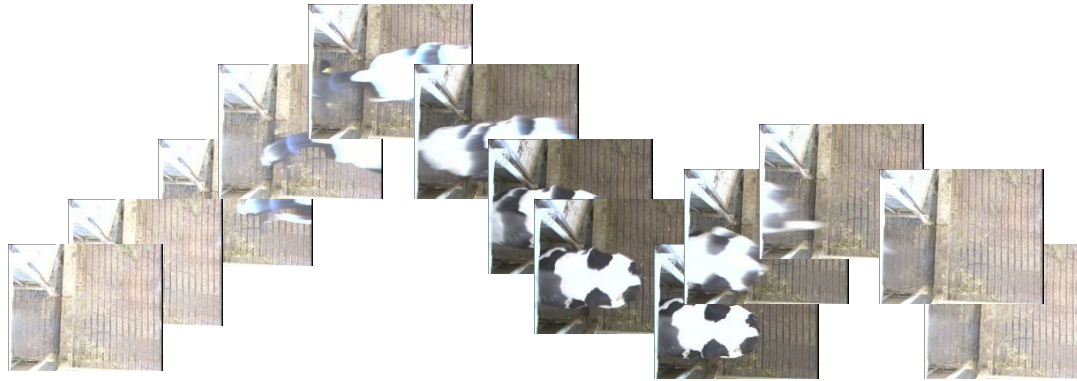


Peaks are strongly related to the presence of cows



Terabytes of data \rightarrow Megabytes of data

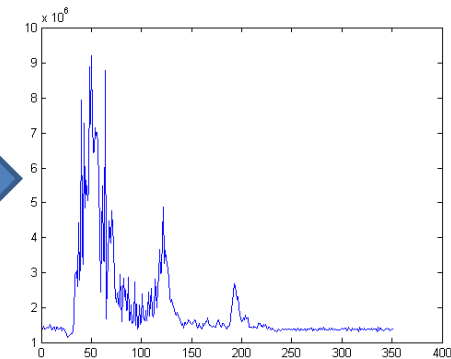
Selection of Cow Images



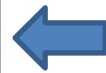
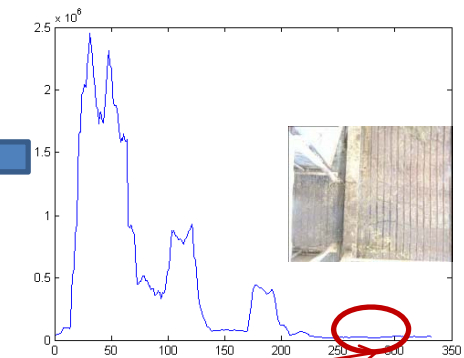
Acquisition interval of four hours

$$E = \sum_{i=1}^N \sum_{j=1}^M |I_t(i, j) - I_{t+1}(i, j)|$$

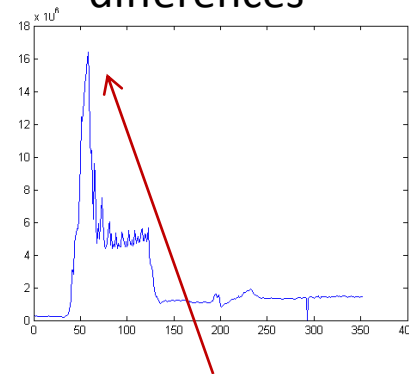
Interframe differences



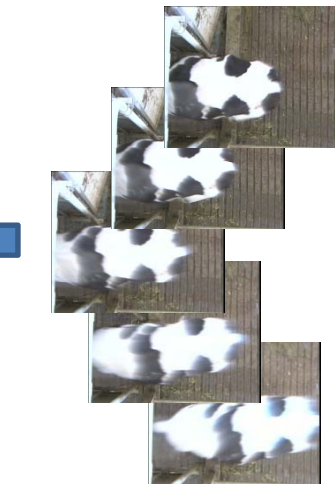
Local variance analysis



Frames-Background differences



Peak indicates the presence of the cow



Five frames per cow

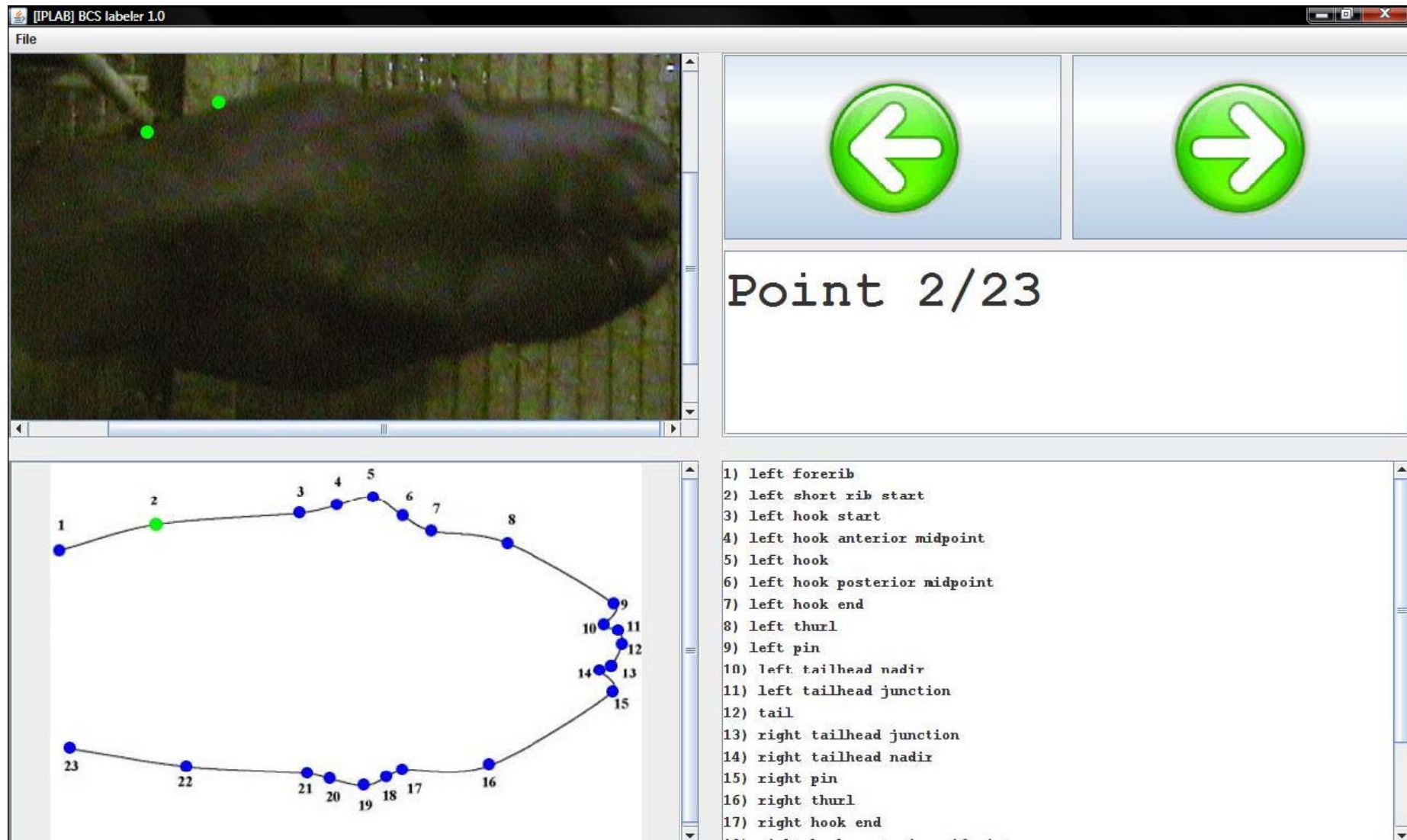


Best Frame
Manually selected



BCS Labeller: available for download

<http://iplab.dmi.unict.it/bcs/>



according to Bewley et al. 2008

Cow Images Identification and BCS Assignment

Cow	Time stamp	BCS Technician 1	BCS Technician 2
136	20/05/2009 11:18 AM	3,00	3,50
:	:	:	:
738	20/05/2009 11:23	3,00	2,75
132	21/05/2009 10:24	4,50	4,50
110	21/05/2009 22:01	3,50	3,50

Time



20_05_09_11_22_Frame245.jpg

20_05_09_11_22_Frame246.jpg

20_05_09_11_23_Frame1.jpg

20_05_09_11_23_Frame2.jpg

20_05_09_11_23_Frame3.jpg

20_05_09_11_23_Frame4.jpg

20_05_09_11_23_Frame5.jpg

20_05_09_11_23_Frame6.jpg

20_05_09_11_23_Frame7.jpg

20_05_09_11_23_Frame8.jpg

20_05_09_11_23_Frame9.jpg

20_05_09_11_23_Frame10.jpg

20_05_09_11_23_Frame11.jpg

20_05_09_11_23_Frame12.jpg




Ferguson et al. 1994

738_2.875_20_05_09_11_23_Frame5.jpg

Semi-Atomic Identification and Labeling Propagation (1)

Identified Images

Dataset with 29 different Cows



Labelled Shapes

Cow ID and corresponding BCS

Given a new unidentified image of the dataset, we have to recognize its ID in order to assign the estimated BCS.



32_3.5 21 05 09 10 12_Frame55.jpg

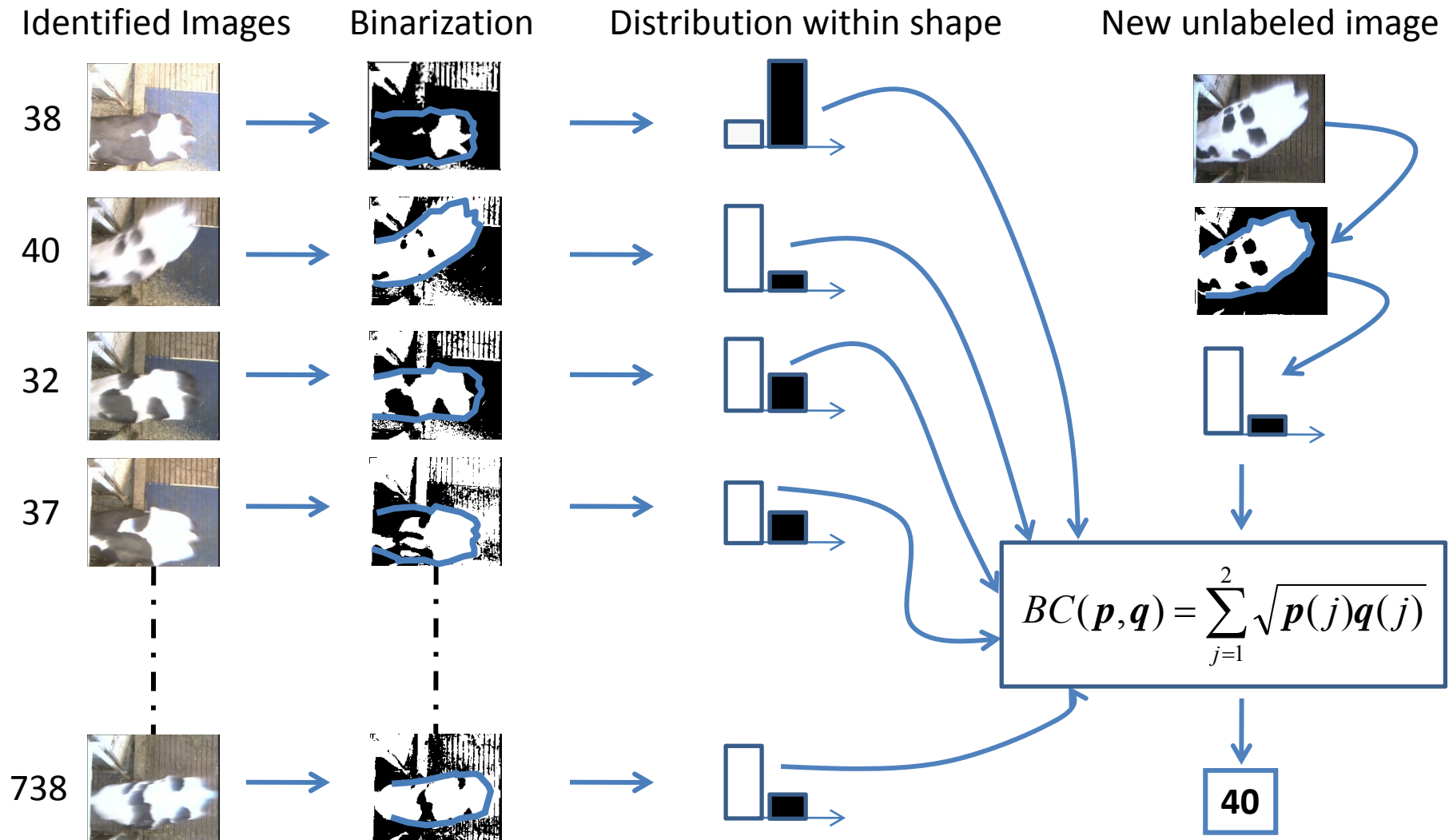


37_34_Frame153.jpg

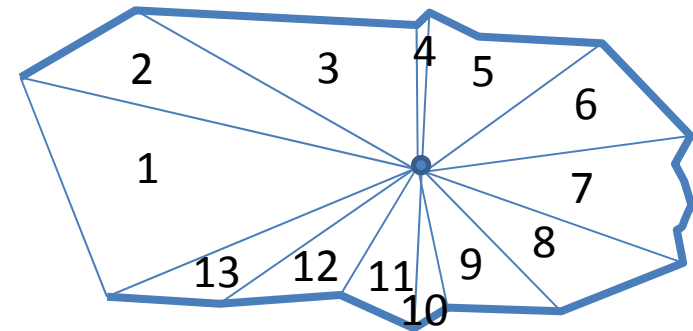


738_2.875_22_05_09_13_52_Frame15.jpg

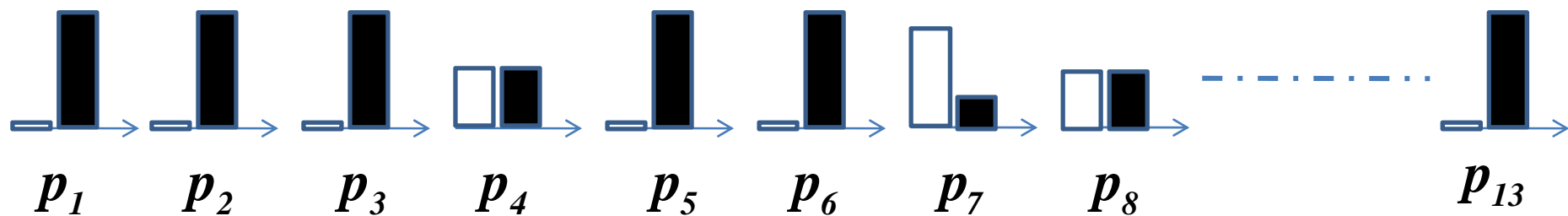
Semi-Atomic Identification and Labeling Propagation (2)



Semi-Atomic Identification and Labeling Propagation (3)



Different Cows with similar Color Distributions



The similarity is obtained by averaging the similarity of distributions of corresponding subregions measured with the Bhattacharyya coefficient.

$$BC(p, q) = \frac{1}{13} \sum_{i=1}^{13} \sum_{j=1}^2 \sqrt{p_i(j)q_i(j)}$$

Semi-Atomic Identification and Labeling Propagation (4)

Cow_g



Query labeled with ID and BCS

Top 10 retrieved

Selected by technicians and labeled by SW as the Query

Similar₁



Similar₂



Similar₃



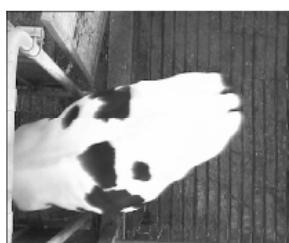
Similar₄



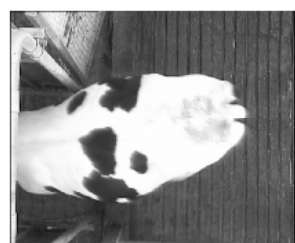
Similar₅



Similar₆



Similar₇



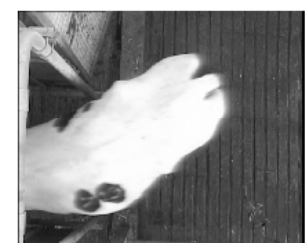
Similar₈

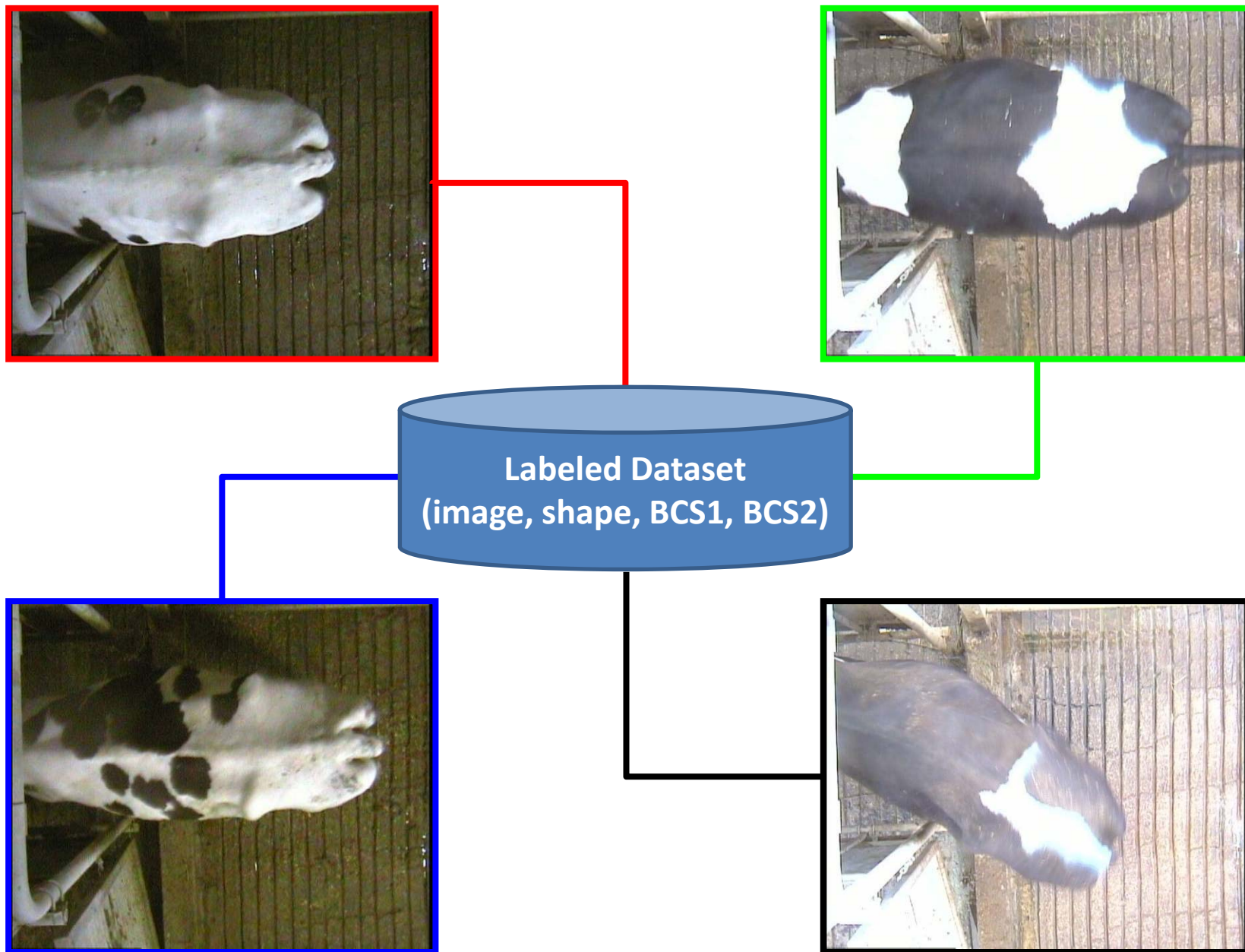


Similar₉



Similar₁₀





Examples of images within the experimental dataset

BCS Estimation: Proposed Approach

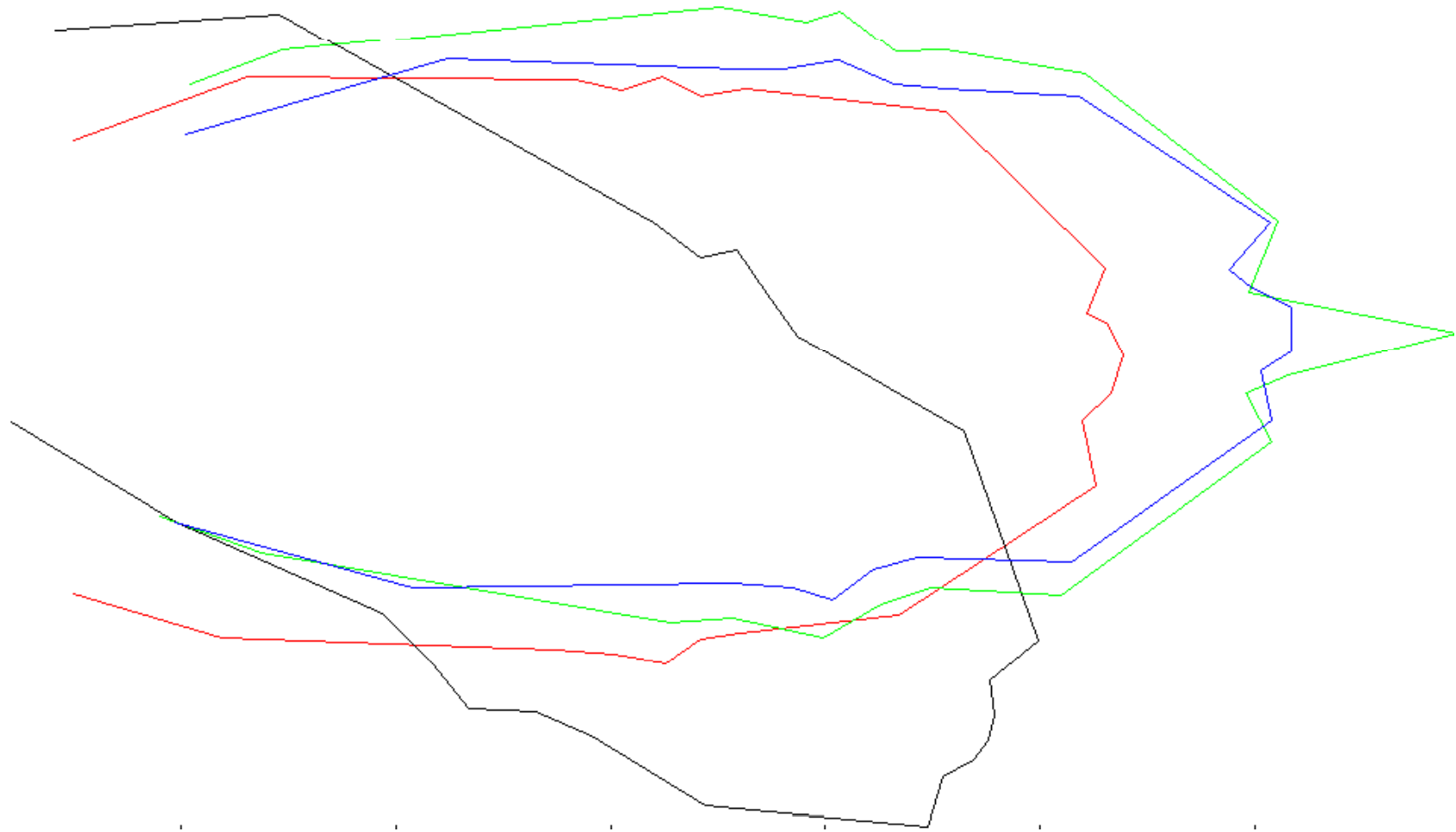
- Alignment of the cows' body shape to the same pose.
- Reconstruction of cows' body shape through a linear combination of basis shape obtained through statistical shape analysis from a dataset of examples.
- Description of the cows' body shape considering the variability in deviations from the mean shape.
- Kernel space to deal with the non-linearity of cows' body shape.
- Regression Machine used to infer BCS of new samples.

What is Shape?

Geometric information that remains when location, scale and rotational effects are removed

(Kendall, 1984)

Example of Shapes from Anatomical Points



Different Scale, Position and Orientation

Shape Alignment

- To establish the *pose* we used the left hook (5), right hook (19), tail (12), left forerib (1) and right forerib (23) as reference anatomical points.

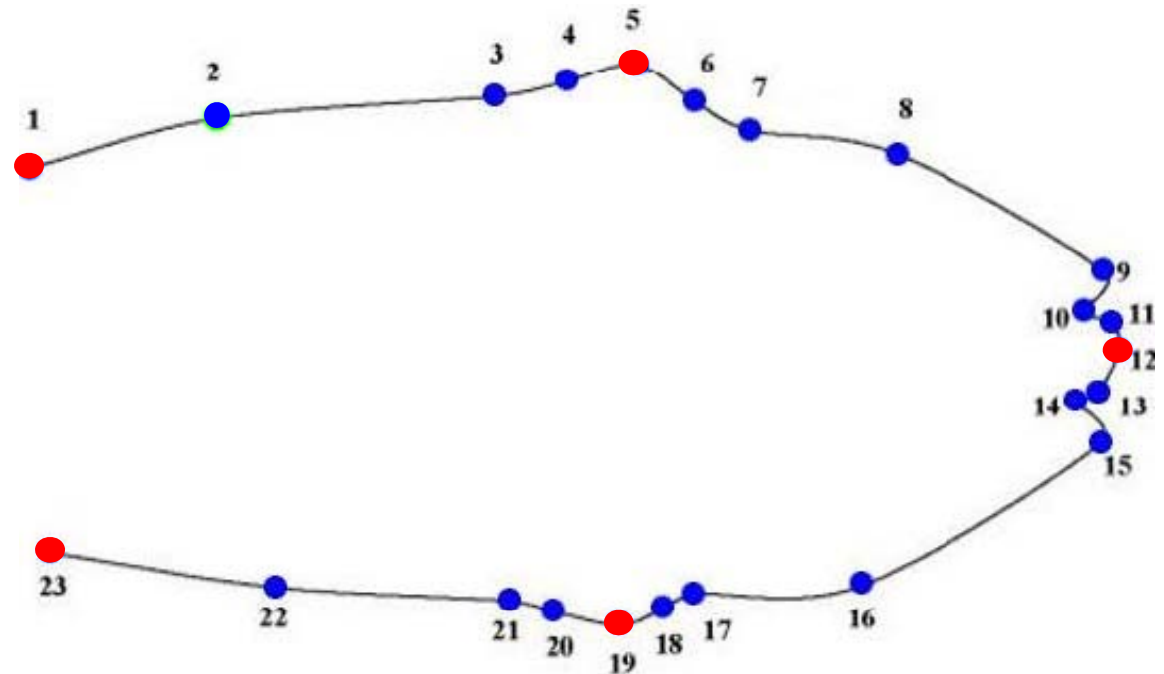
Alignment (shape)

1) Translation

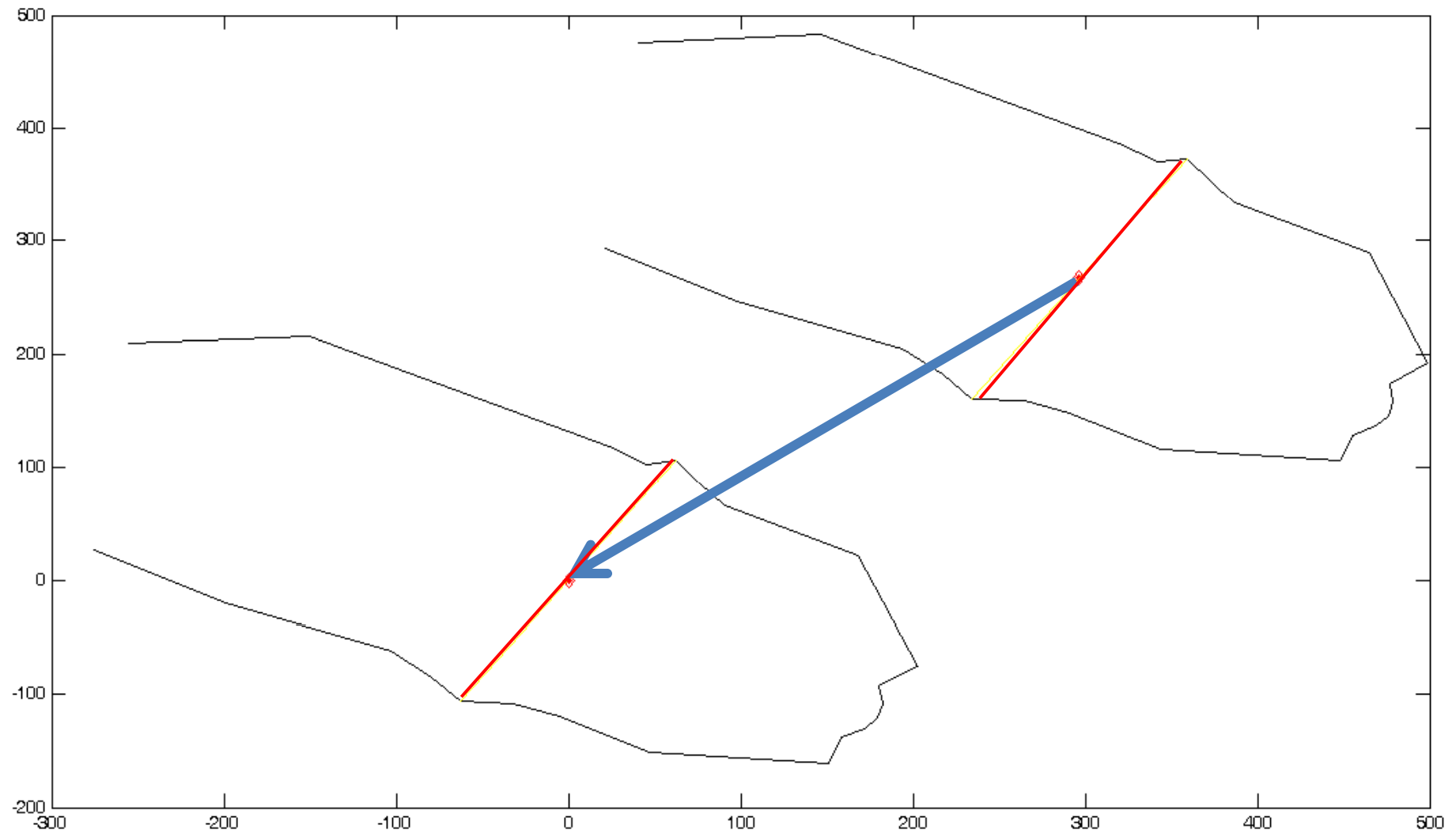
2) Rotation

3) Scaling

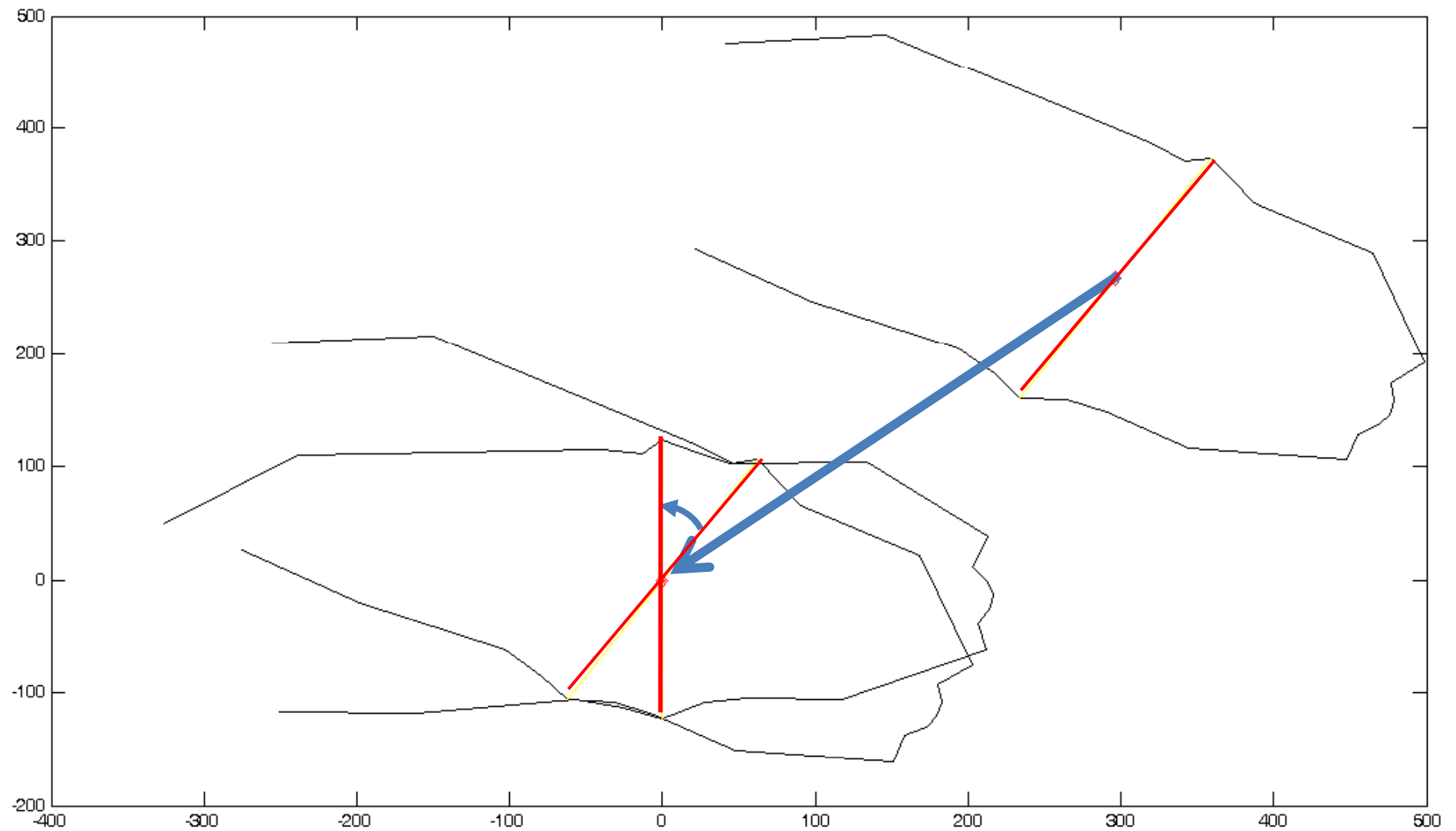
Return Aligned Shape



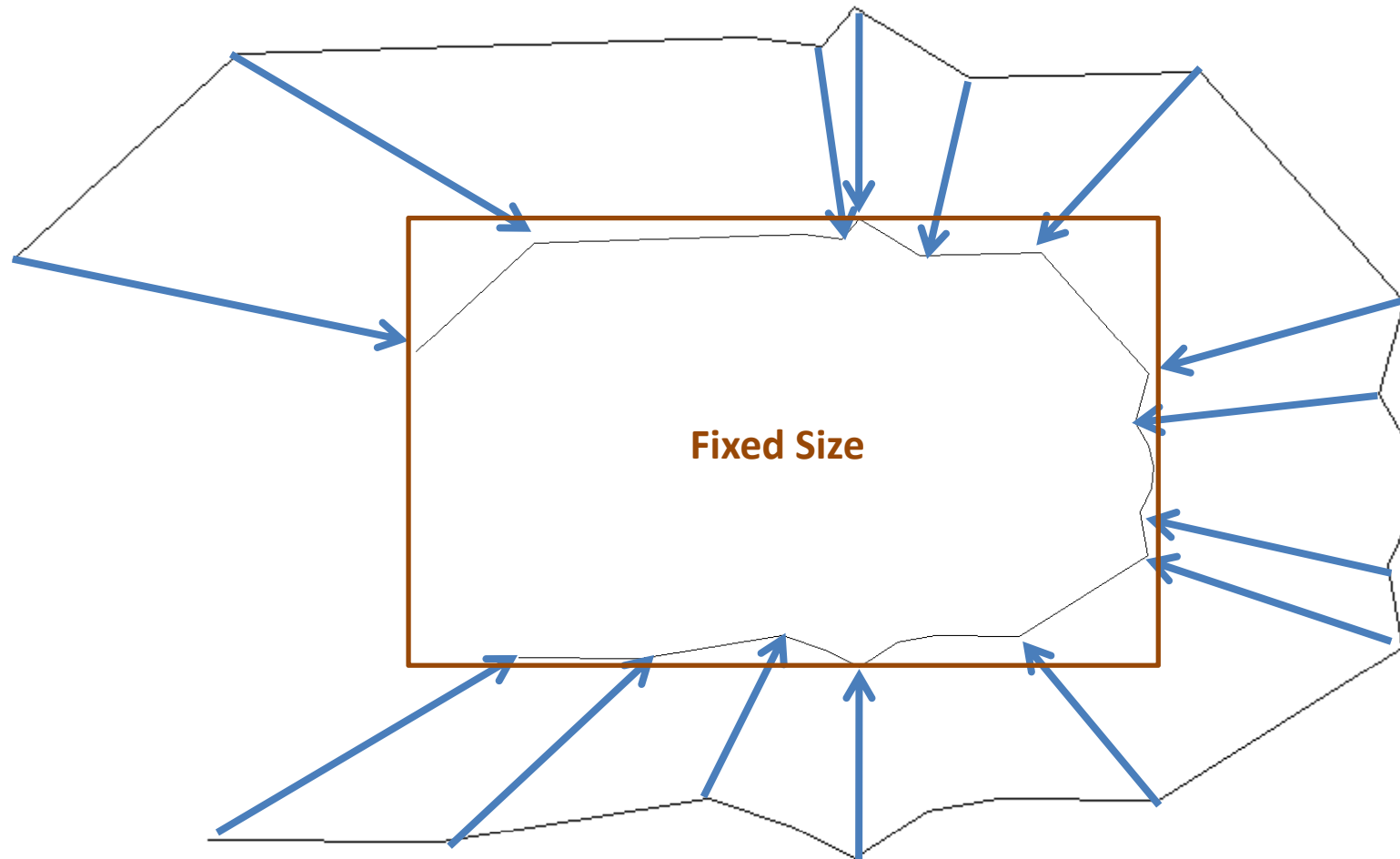
Translation Alignment



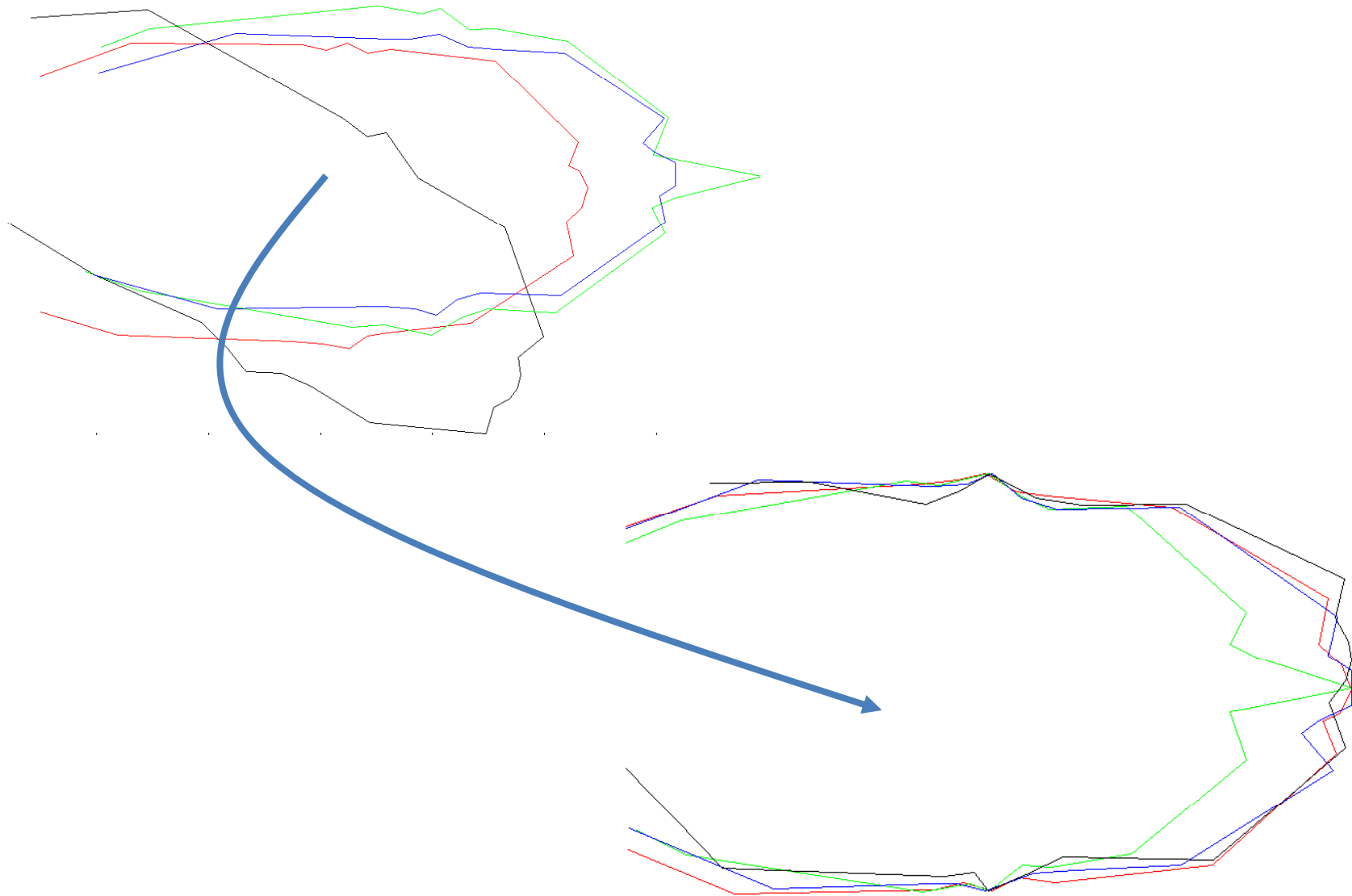
Rotation Alignment



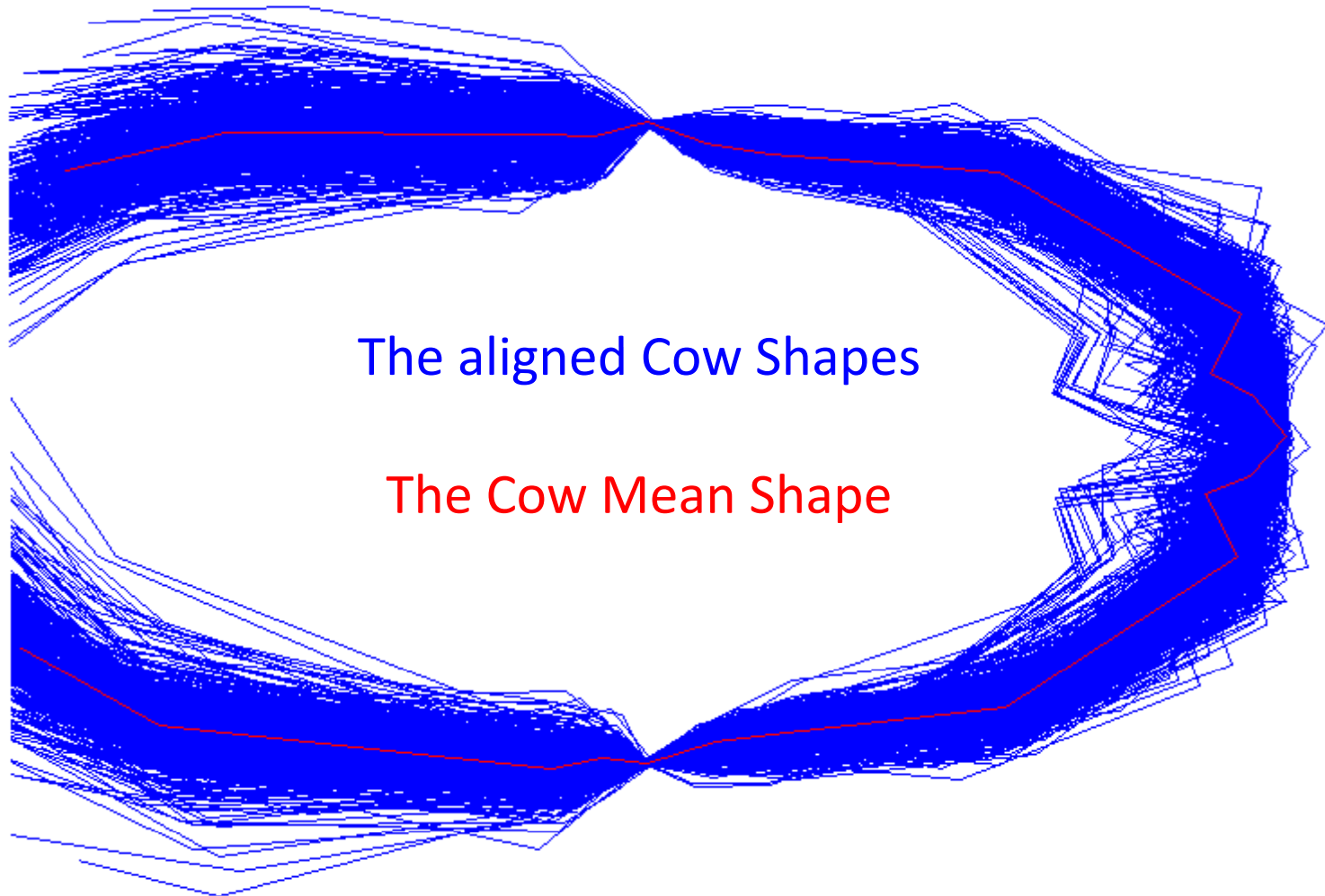
Scale Alignment



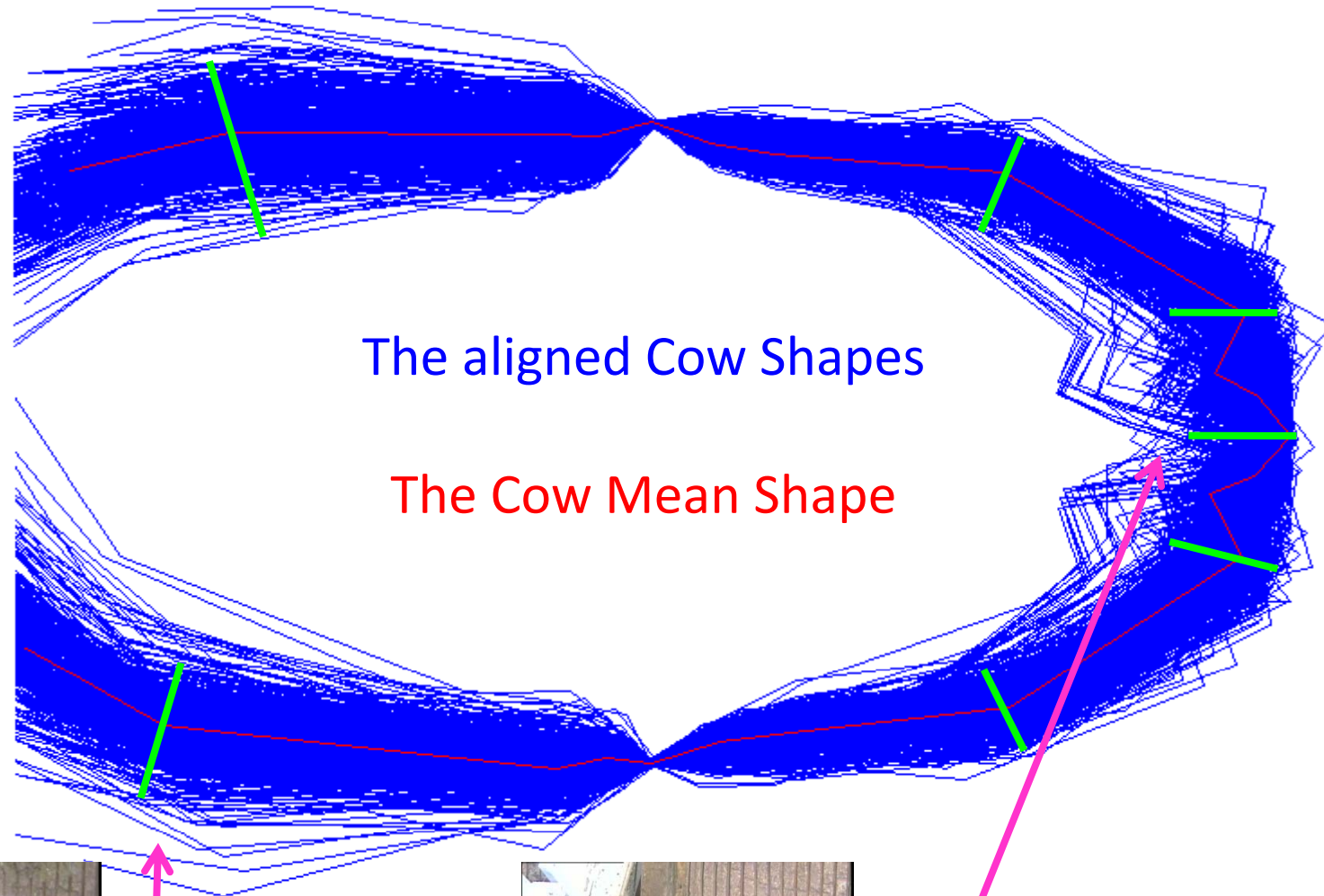
Examples of Aligned Shapes



Aligned Shapes



Variance with respect to the mean shape



Outliers:
Cow Body Distortion



Outliers:
Labelling Errors

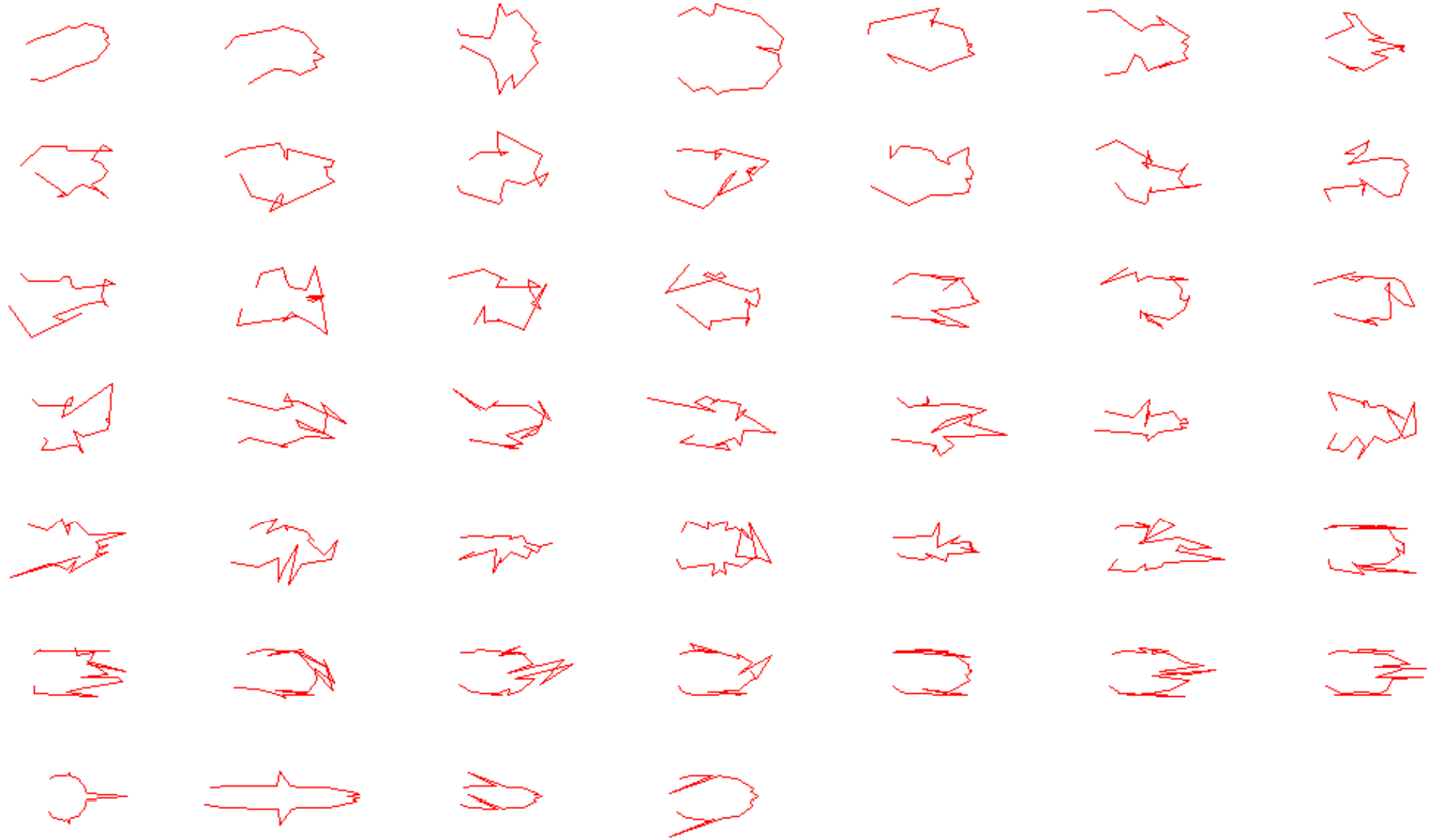
Main Idea

- Describe the shape of cows considering the different variability from the mean shape.
- Shapes of cows are reconstructed by using a linear combination of basis *shape*
- Basis shape are obtained through principal component analysis from a dataset of examples

$$\Phi_{Linear_Kernel}(\mathbf{x}) = \mathbf{x}$$

Basis Shape

$$\{\mathbf{e}_j^\Phi\}_{j=1}^{n_\Phi}$$

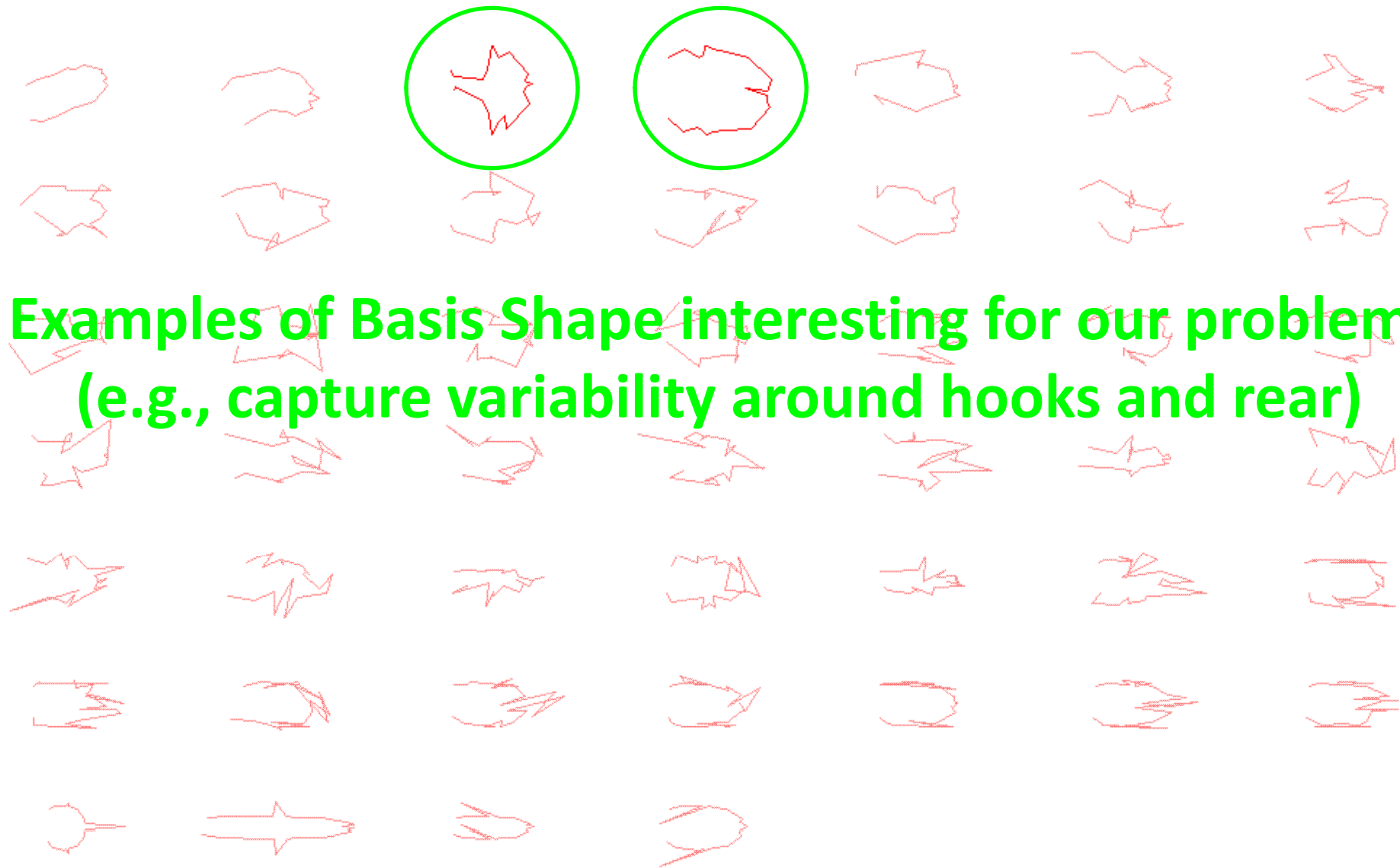


The Basis shape computed on the experimental dataset

$$\Phi_{Linear_Kernel}(\mathbf{x}) = \mathbf{x}$$

Basis Shape

$$\{\mathbf{e}_j^\Phi\}_{j=1}^{n_\Phi}$$



The Basis shape computed on the experimental dataset

Shape Reconstruction

$$\Phi_{Linear_Kernel}(\mathbf{x}) = \mathbf{x}$$

$$\{\mathbf{e}_j^\Phi\}_{j=1}^{n_\Phi}$$

$$\overline{\mathbf{s}}_\Phi' = \frac{1}{m} \sum_{i=1}^m \Phi(\mathbf{s}_i')$$

$$\Phi(\mathbf{s}_i') = \overline{\mathbf{s}}_\Phi' + \sum_{j=1}^{n_\Phi} a_{i,j}^\Phi \mathbf{e}_j^\Phi$$

$$a_{i,j}^\Phi = \mathbf{e}_j^{\Phi T} \left(\Phi(\mathbf{s}_i') - \overline{\mathbf{s}}_\Phi' \right)$$

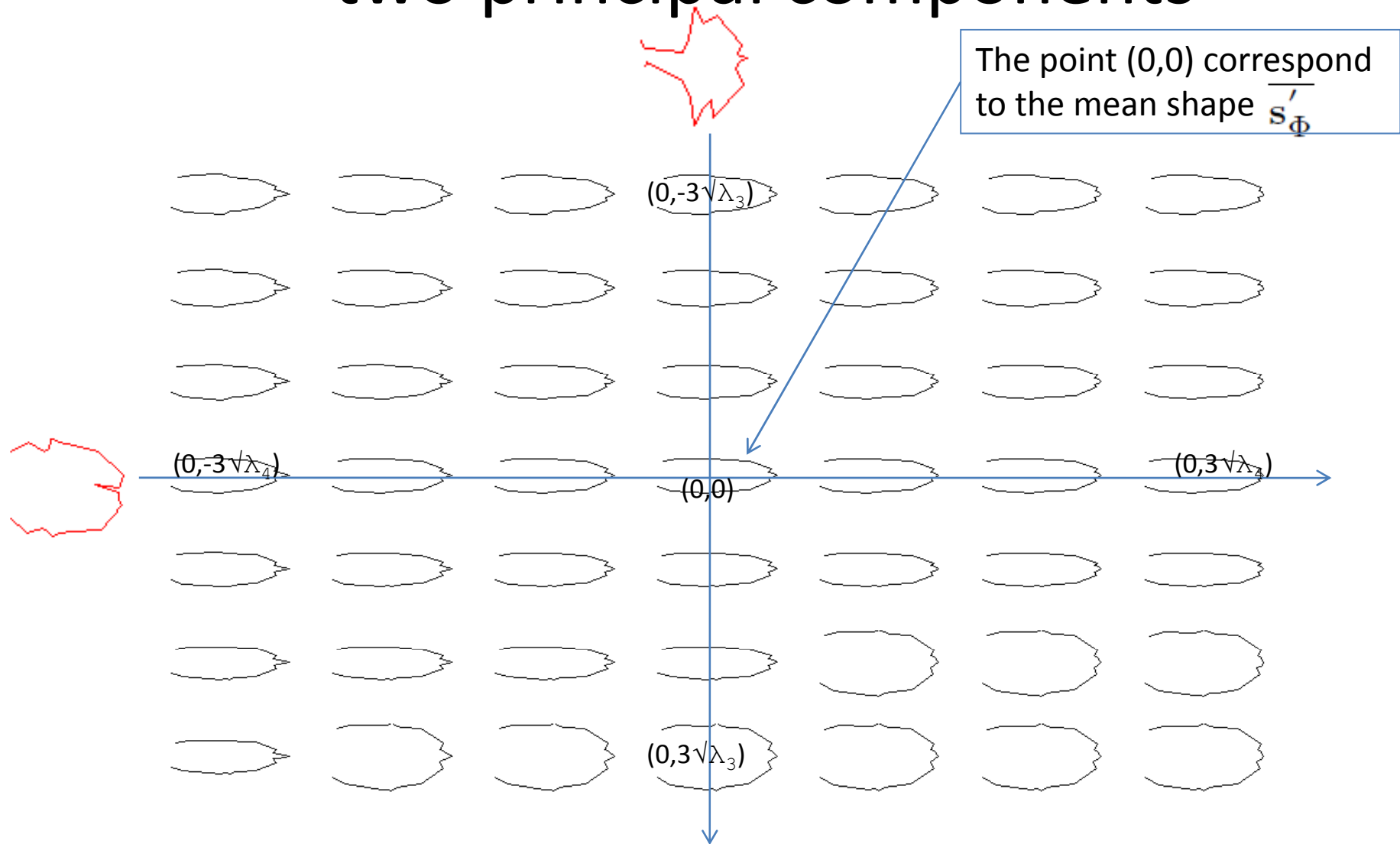
Reconstruction by considering two elements of the basis shape:

$$a_{i,3}^\Phi * \mathbf{e}_3^\Phi + a_{i,4}^\Phi * \mathbf{e}_4^\Phi + \text{Mean Shape } \overline{\mathbf{s}}_\Phi' = \text{RECONSTRUCTED (APPROXIMATED SHAPES)}$$

The vector $[a_{i,1}^\Phi, a_{i,2}^\Phi, \dots, a_{i,n_\Phi}^\Phi]$ is used to describe aspects of the shape $\Phi(\mathbf{s}_i')$

$\Phi_{Linear_Kernel}(\mathbf{x}) = \mathbf{x}$ Reconstruction considering two principal components

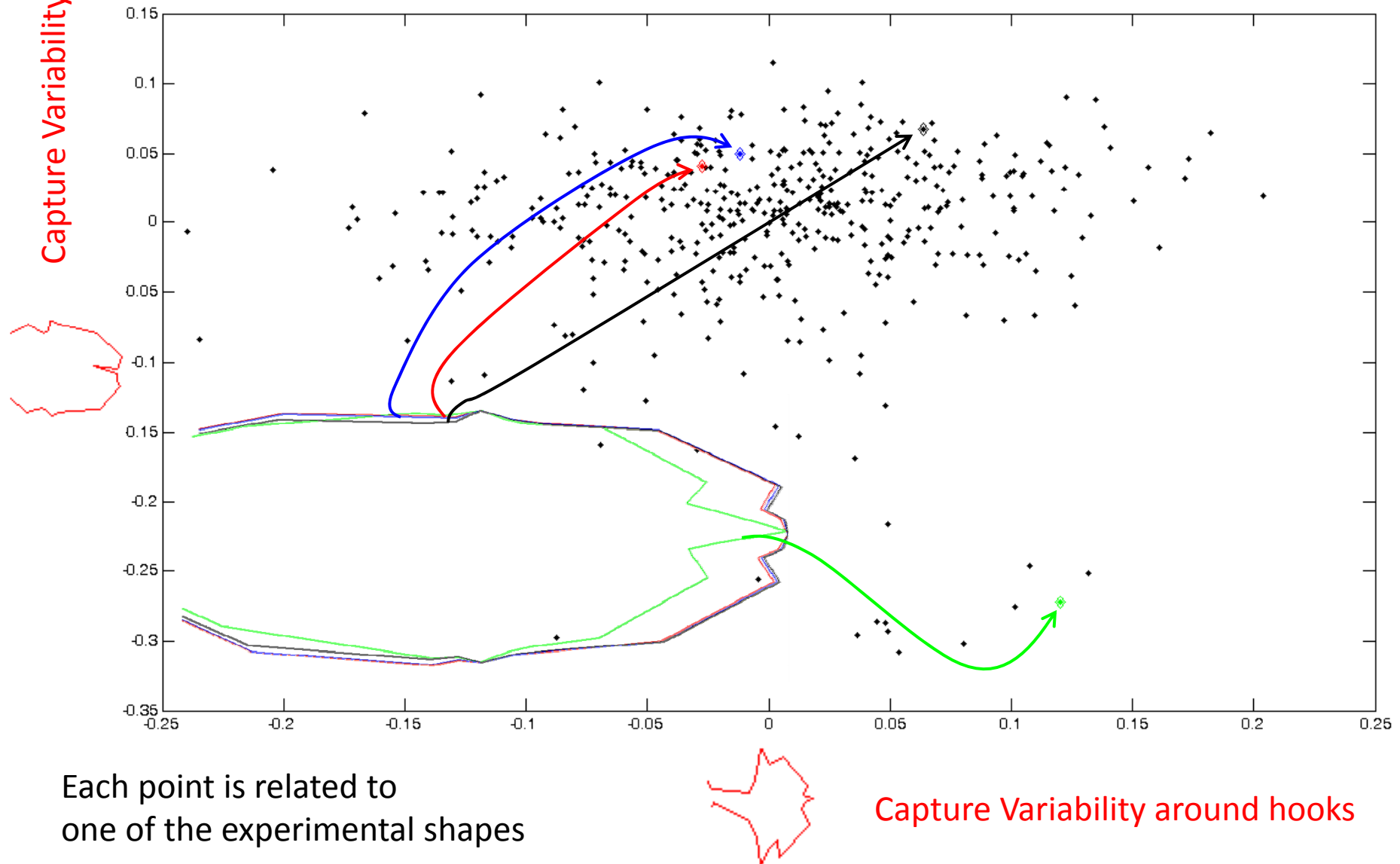
$$a_{i,3}^{\Phi} * \mathcal{E}_3^{\Phi} + a_{i,4}^{\Phi} * \mathcal{E}_4^{\Phi} + \text{Mean Shape } \bar{s}_{\Phi} = \text{RECONSTRUCTED (APPROXIMATED SHAPES)}$$



Points sampled over a 7x7 grid

$\Phi_{Linear_Kernel}(\mathbf{x}) = \mathbf{x}$ Shape Descriptor Space

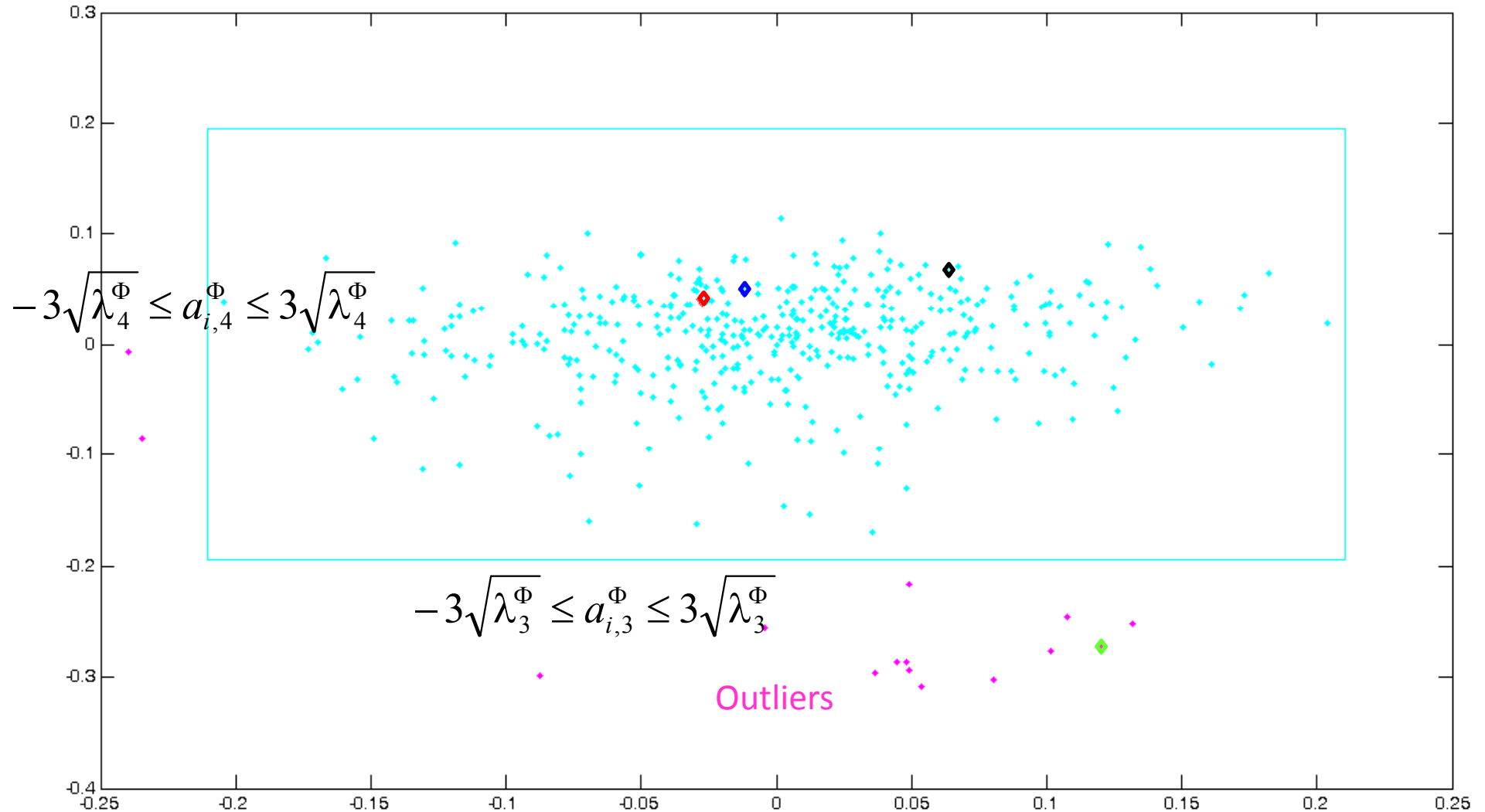
- 1) Red, Blue and Black Shapes are “similar” → Corresponding Points are close
- 2) Green Shape differ from Red, Blue, and Black shapes → Green Point is far



$$\Phi_{\text{Linear_Kernel}}(\mathbf{x}) = \mathbf{x}$$

Remove Outliers

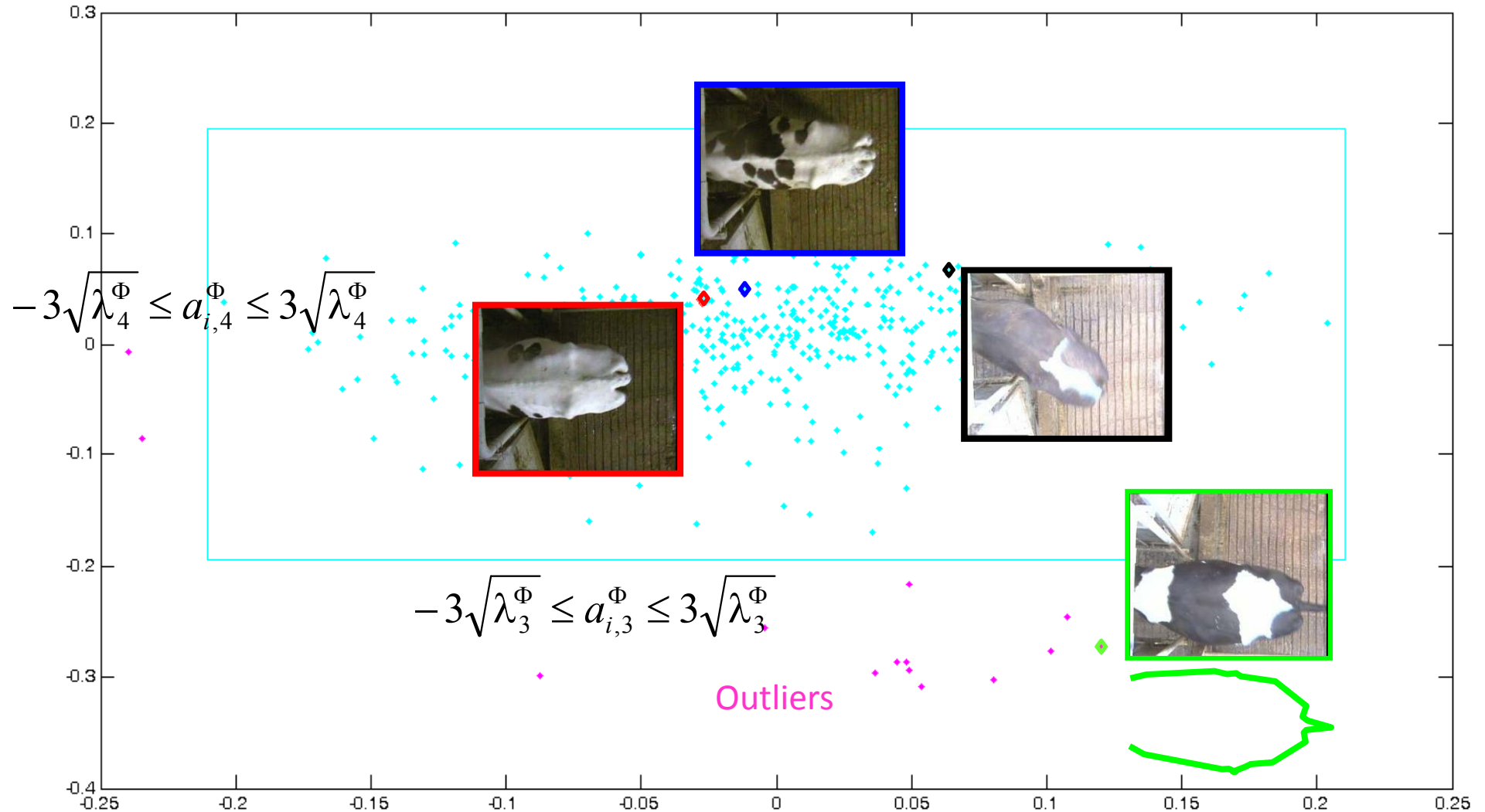
- Suitable space limits:



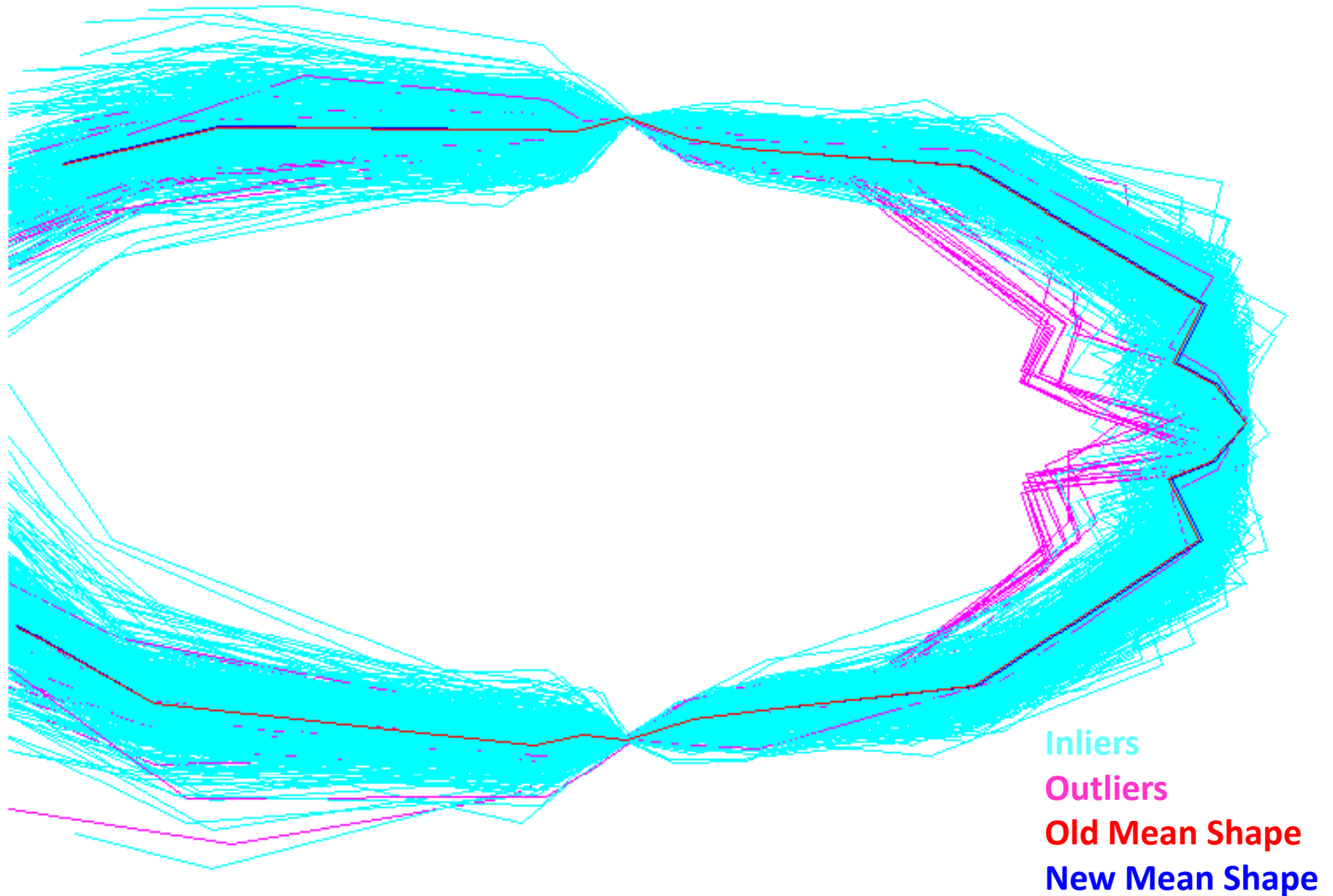
$$\Phi_{Linear_Kernel}(\mathbf{x}) = \mathbf{x}$$

Remove Outliers

- Suitable space limits:



Identified Outliers




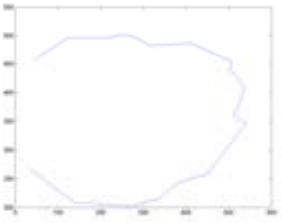

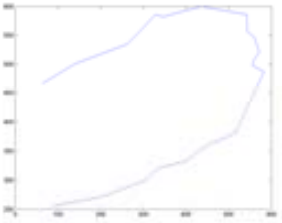

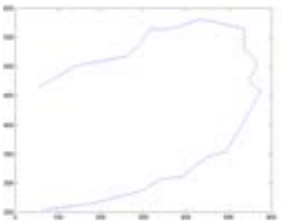
BODY CONDITION SCORE DATABASE

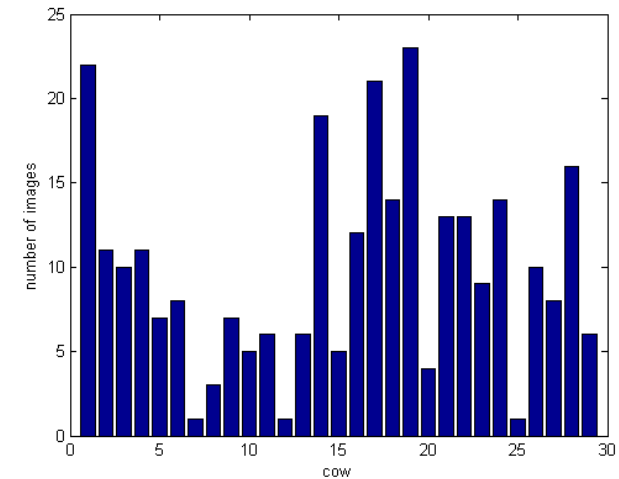


<http://iplab.dmi.unict.it/bcs/>

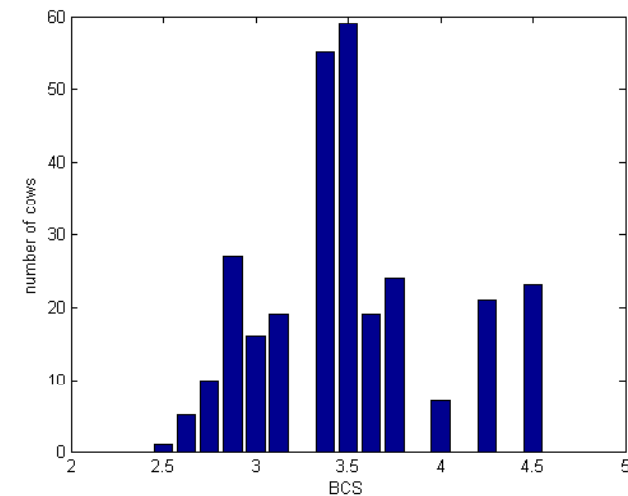
The Final BCS Database contains 286 labelled images with the related ground truth (anatomical points and BCS).

Overview Dataset Publications Software People

		Download 3 3
Original Image	Shape from Anatomical Points	Date BCS ₁ BCS ₂
		Download 3.25 3.5
Original Image	Shape from Anatomical Points	Date BCS ₁ BCS ₂
		Download 3.25 3.5
Original Image	Shape from Anatomical Points	Date BCS ₁ BCS ₂



Number of images per cow



BCS distribution

BCS Estimation

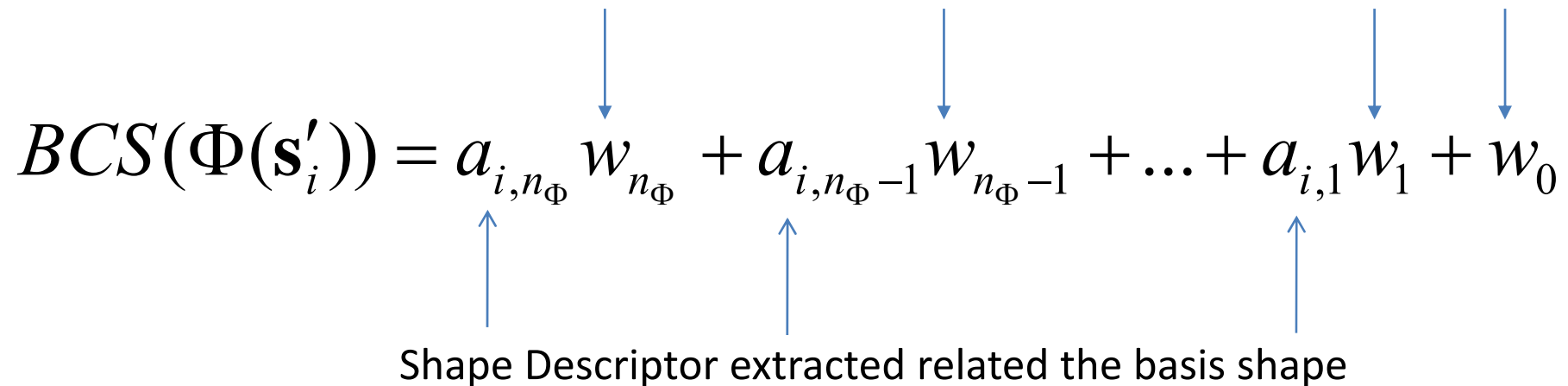
We used the coefficients corresponding to the principal components in order to describe the shape of cows.

A regression approach was employed to infer the BCS.

The parameters are learned using the BCS labels estimated by technicians

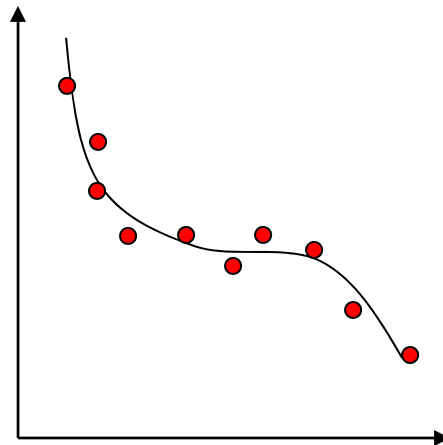
$$BCS(\Phi(\mathbf{s}'_i)) = a_{i,n_\Phi} w_{n_\Phi} + a_{i,n_\Phi-1} w_{n_\Phi-1} + \dots + a_{i,1} w_1 + w_0$$

Shape Descriptor extracted related the basis shape

The diagram illustrates the BCS estimation process. At the top, the text "The parameters are learned using the BCS labels estimated by technicians" has four blue arrows pointing down to the coefficients a_{i,n_Φ} , $a_{i,n_\Phi-1}$, $a_{i,1}$, and w_0 in the equation below. The equation is $BCS(\Phi(\mathbf{s}'_i)) = a_{i,n_\Phi} w_{n_\Phi} + a_{i,n_\Phi-1} w_{n_\Phi-1} + \dots + a_{i,1} w_1 + w_0$. Below the equation, the text "Shape Descriptor extracted related the basis shape" has three blue arrows pointing up to the coefficients a_{i,n_Φ} , $a_{i,n_\Phi-1}$, and $a_{i,1}$.

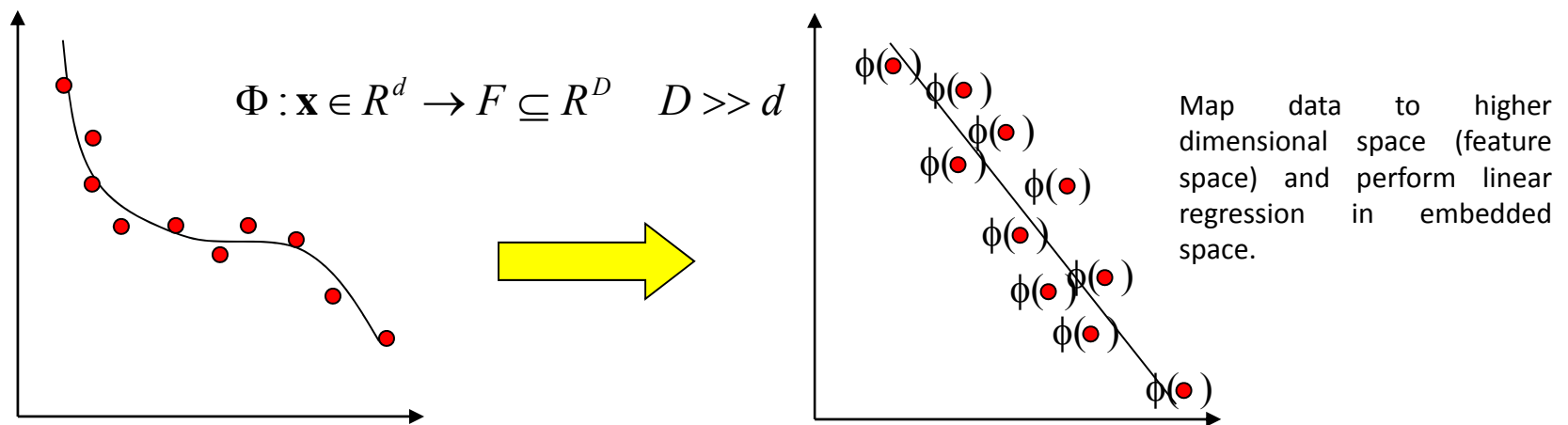
Kernel Principal Component Regression

- Linear Regression: address the problem of identifying linear relations between one selected variable and the input variables where the relation is assumed linear
- Often, however, the relations that are sought are nonlinear



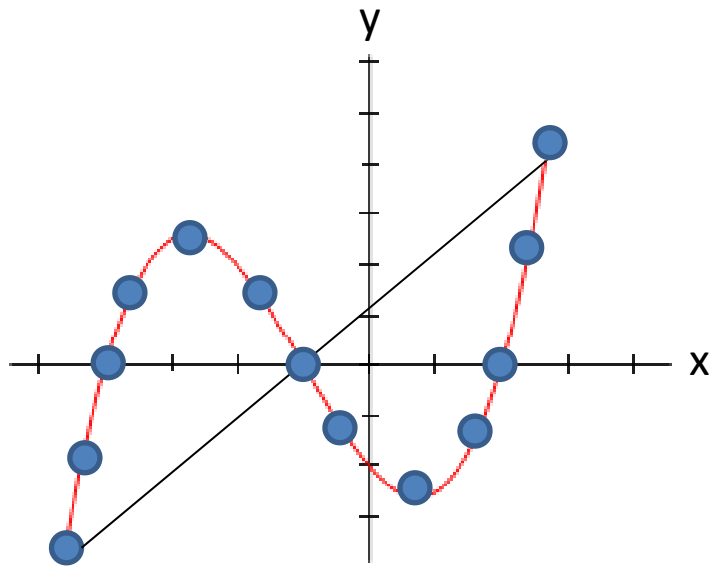
Kernel Principal Component Regression

- Key Idea: Map the input space into a new features space in such way that the sought relations can be represented in a linear form and hence the linear regression algorithm described above will be able to detect them.



The choice of the map ϕ aims to convert the nonlinear relations into linear ones. Hence the map reflects our expectations about the relation $y=f(x)$

Example: Polynomial features space



$$\begin{array}{c} x \\ \downarrow \\ \Phi(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix} \end{array}$$

$$f(x) = w_1x + b$$

$$f(x) = \langle \mathbf{w}, \phi(x) \rangle + b = w_1x + w_2x^2 + w_3x^3 + b$$

$$w_1 = 1/4$$

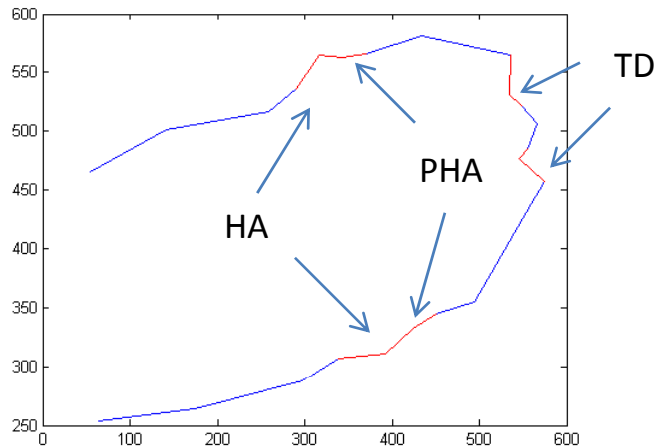
$$w_2 = 1/4$$

$$w_3 = -1/2$$

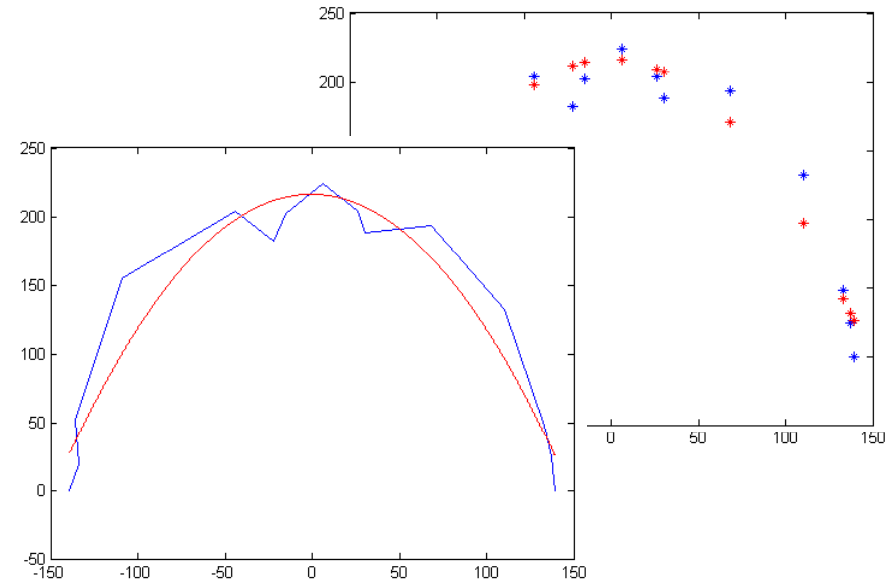
$$b = -2$$

Related works taken into account for comparison

Bewley et al. 2008



Halachmi et al. 2008



Model 1

$$y_{ij} = \mu + \beta_1 HA_{ij} + \beta_2 PHA_{ij} + \beta_3 (HA \times PHA)_{ij} + e_{ij}$$

$$TBCS = 5 \times 9 \times \frac{1}{MAE}$$

Model 2

$$y_{ij} = \mu + \beta_1 HA_{ij} + \beta_2 PHA_{ij} + \beta_3 (HA \times PHA)_{ij} + \beta_4 TD_{ij} + \beta_5 (PHA \times TD)_{ij} + e_{ij}$$

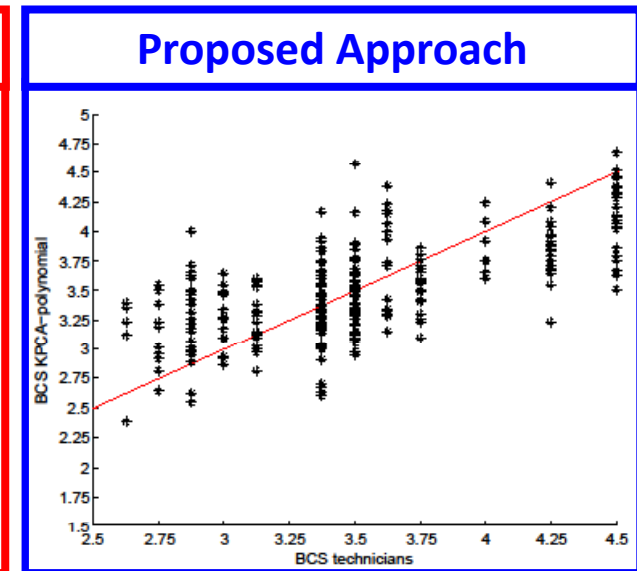
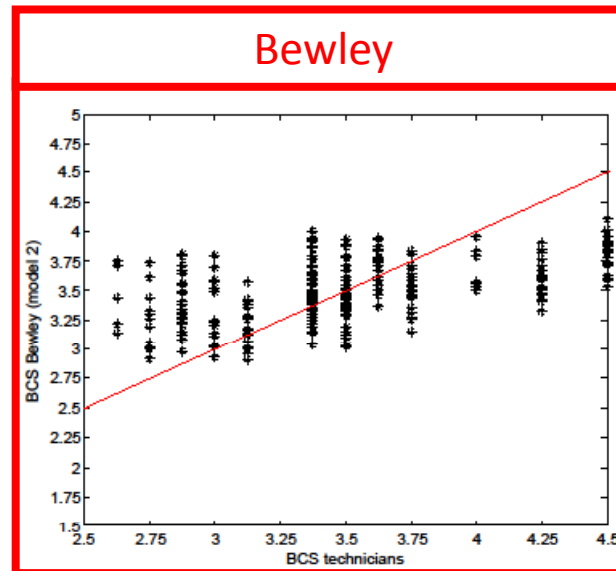
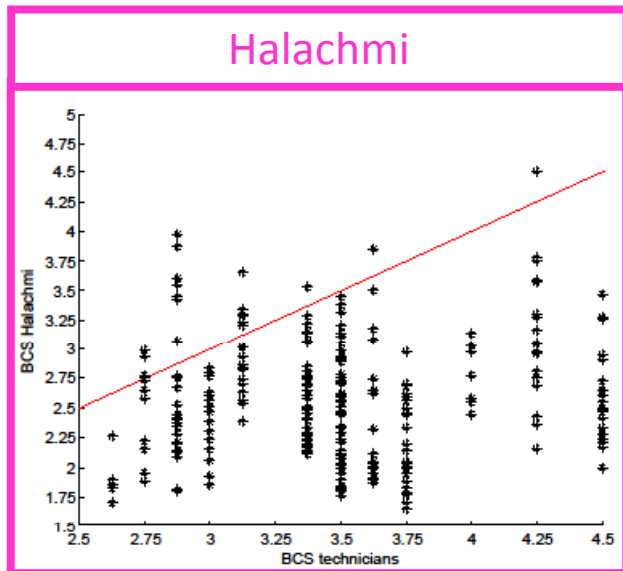
MAE (Mean Absolute Error)

Evaluation



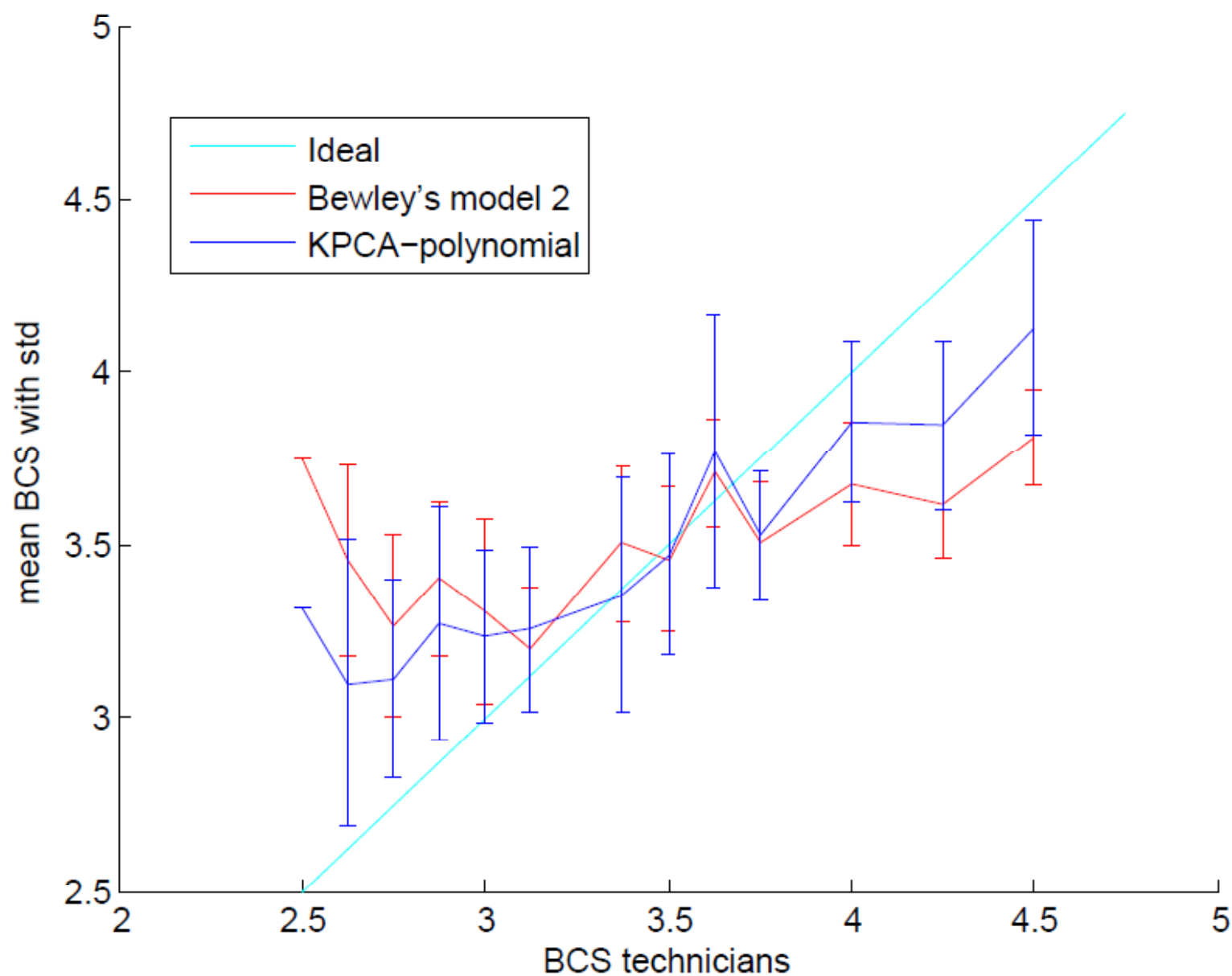
- Leave One Out Cross Validation
 - Each run of LOOCV involved a single observation of the dataset as test, and the remaining samples as training data. The average error rate was computed taking into account all runs.
- Regression Error Characteristic Curves
 - It is essentially the cumulative distribution function of the error. The area over the curve is a biased estimation of the expected error of an employed regression model.

Results (1)



Method	Mean BCS Error
Halachmi et al. 2007 (Using Anatomical Points)	0.9837
Bewley et al. 2008 (Model 2)	0.3289
Proposed Approach (KPCA - Polynomial Kernel)	0.3059

Results (2)



Results (3)

Admissible Error: 0.25

