MRC:
The Maximum Rejection Classifier for Pattern Detection

With

Michael Elad, Renato Keshet

The Problem

- **Pattern Detection**: Given a pattern that is subjected to a particular type of variation, detect occurrences of this pattern in an image.

- Detection should be:
  - Accurate (small number of mis-detections/false-alarms).
  - As fast as possible.

Example

Face detection in images

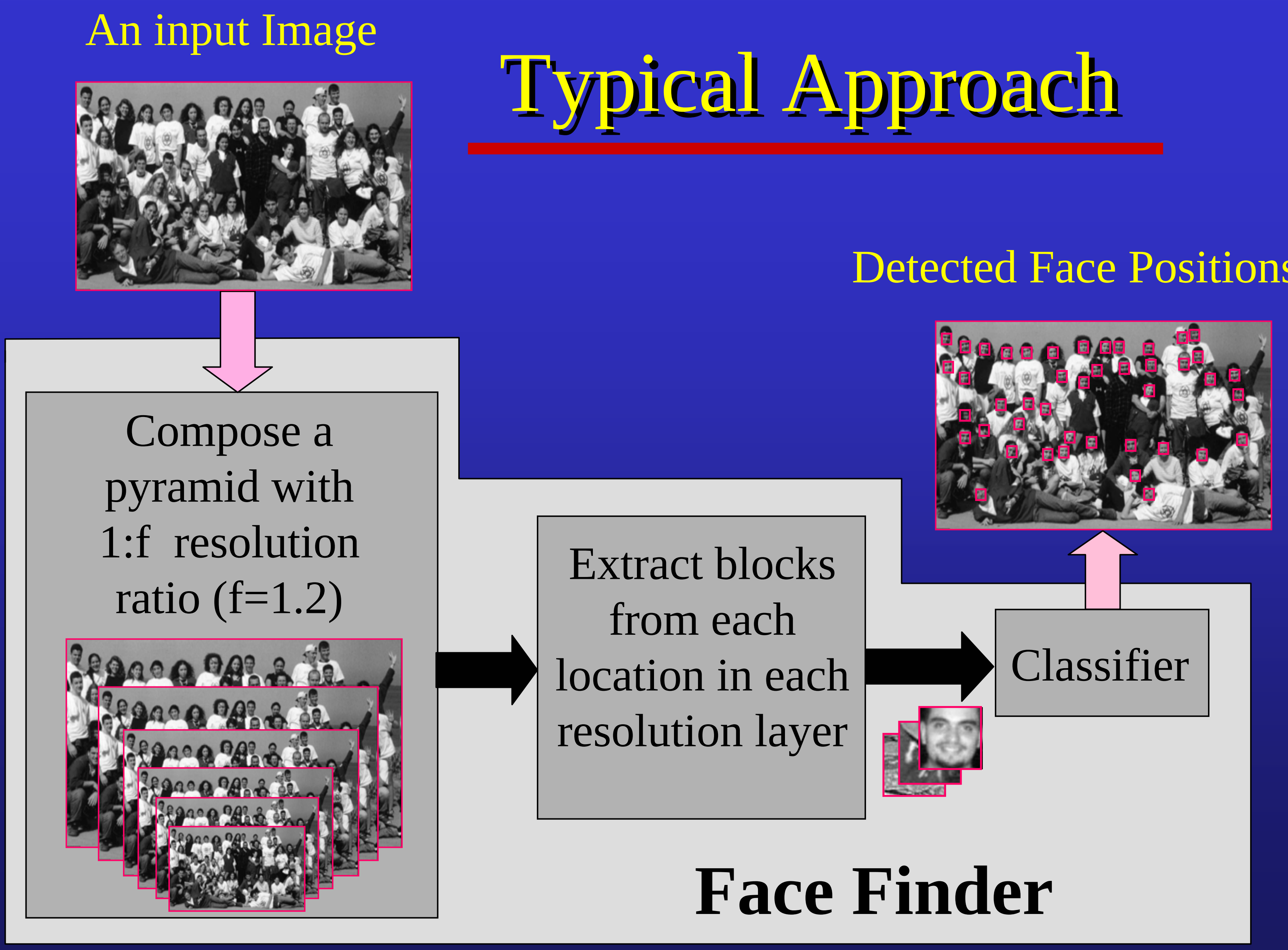
Face Examples

Variations in Faces

Faces may vary in their appearance:

- Scale
- Location
- Illumination condition
- Pose
- Identity
- Facial expression

Typical Approach



- The above type of algorithm is capable of finding faces:
  - In various scales
  - In various locations

Brute Force Search

  - In various illumination conditions
  - In various pose
  - Of various identities
  - In various facial expressions

Trained Classifier

Complexity

Searching for faces in a 1000x1000 image, the classifier is applied 1e6 times

The algorithm' complexity is dominated by the classifier

Pattern Detection as a Classification Problem

- Pattern detection requires a separation between two classes:
  a. The Target class.
  b. The Clutter class.
- Given an input pattern $\underline{Z}$, we would like to classify it as Target or Clutter.

Definition:

- A classifier is a non-linear parametric function $C(\underline{Z},\underline{\theta})$ of the form:

$$C(Z,\theta):\ \Re^n \rightarrow \{+1,-1\}$$

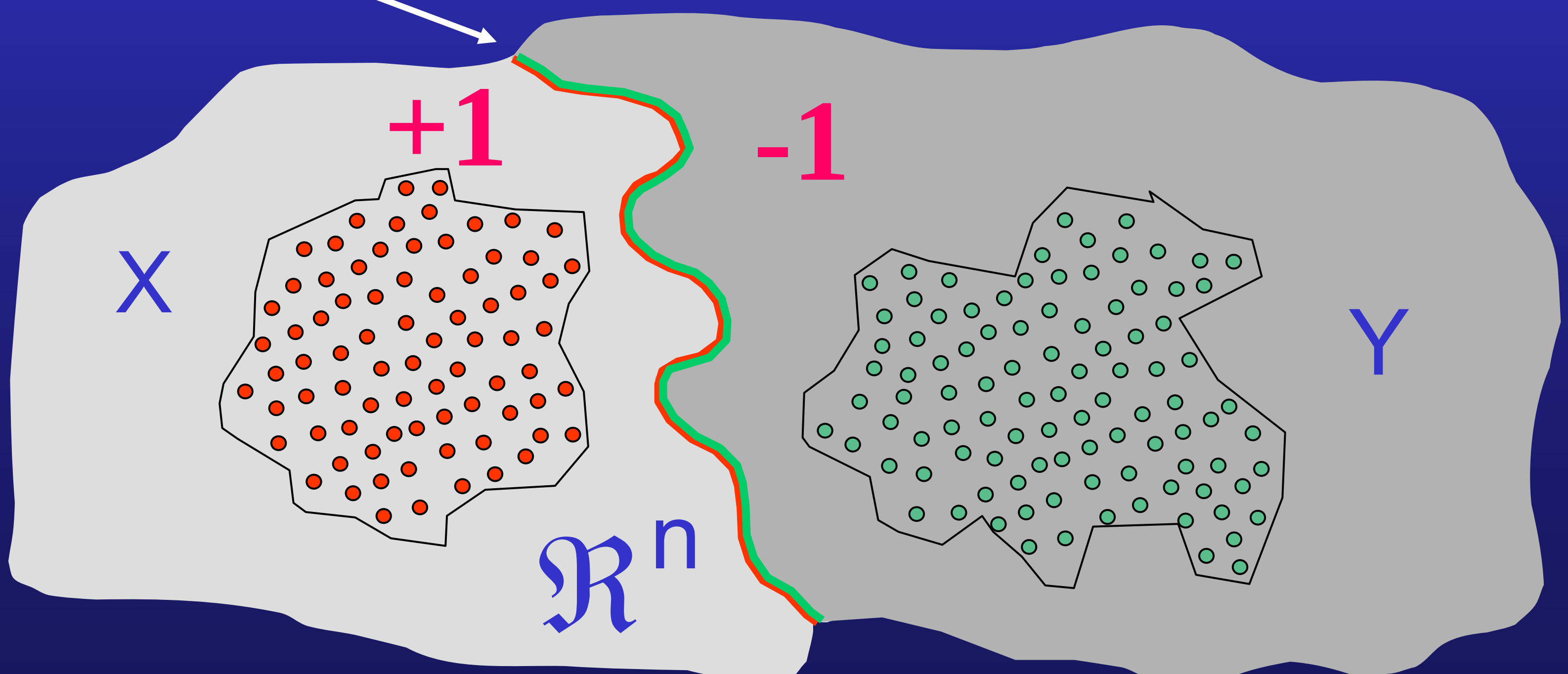
- A simple example: For blocks of 4 pixels $[z_1,z_2,z_3,z_4]$ we can define $C\{Z\}$ as:

$$C(Z,\theta)= \mathrm{sign}(\theta_0+ \theta_1 z_1+ \theta_2 z_2+ \theta_3 z_3+ \theta_4 z_4)$$

Geometric View

C($\underline{Z}$) draws a separating manifold between the two classes

C($\underline{Z}$)=0



Supervised Learning:

- In order to obtain a precise classifier we must find a good choice of parameters $\underline{\theta}$:

$$C\{Z,\theta\}: \quad \Re^n \rightarrow \{+1,-1\}$$

Use examples with known labeling to find a good set of parameters

Example Based Classifiers

- Given two training sets:

$$\left\{\underline{X}_k\right\}_{k=1}^{N_X} \in X \qquad \left\{\underline{Y}_k\right\}_{k=1}^{N_Y} \in Y$$

- We want to find a set of parameters $\underline{\theta}$ such that:

$$C\left\{X_k, \theta\right\} = +1 \qquad C\left\{Y_k, \theta\right\} = -1$$

- We "hope" that the generalization is correct.

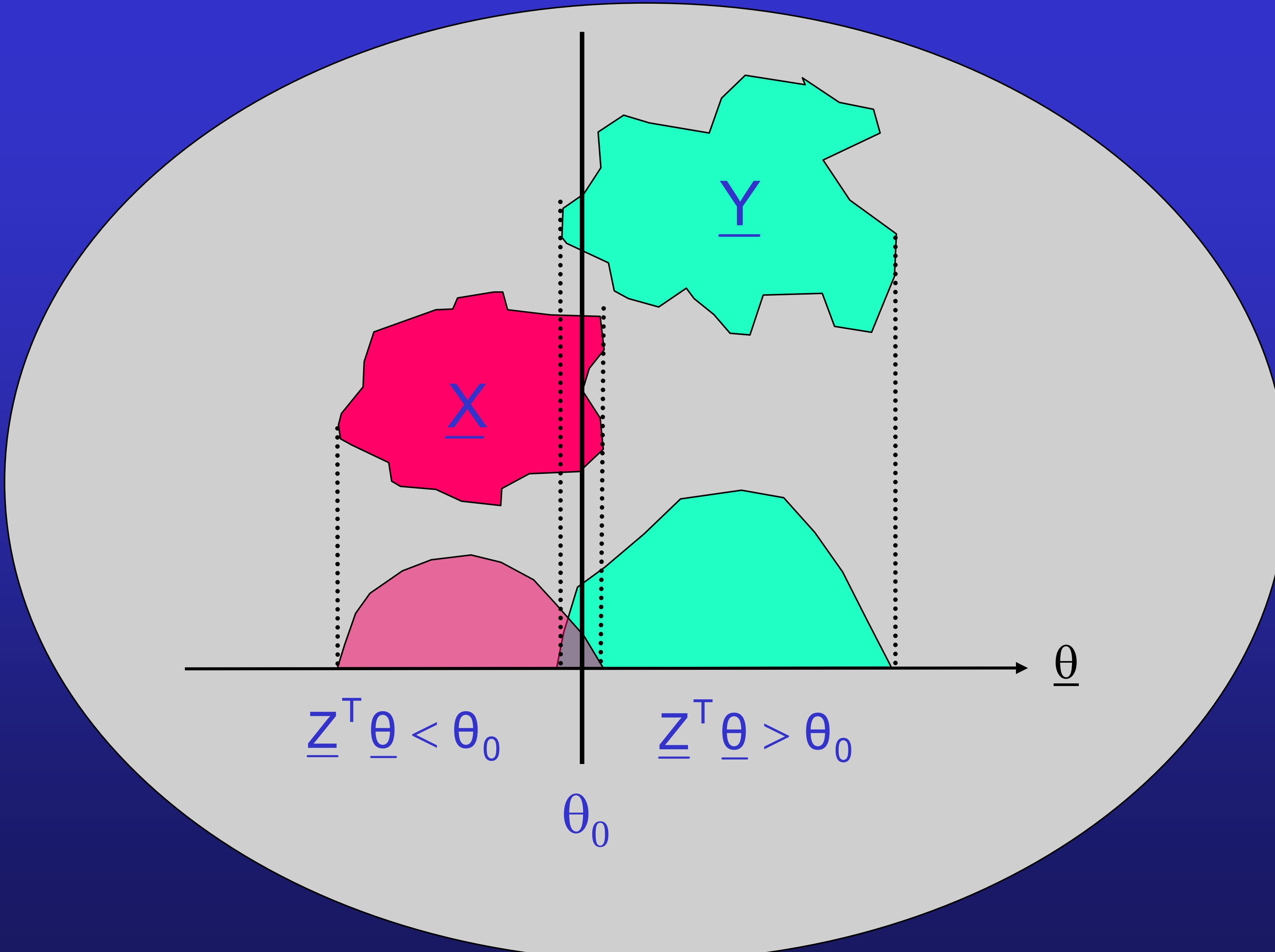
Linear Classifiers

- The *Linear Classifiers* are the simplest ones.

- The decision is based on the projection of an input signal $\underline{Z}$ onto a kernel $\underline{\theta}$.

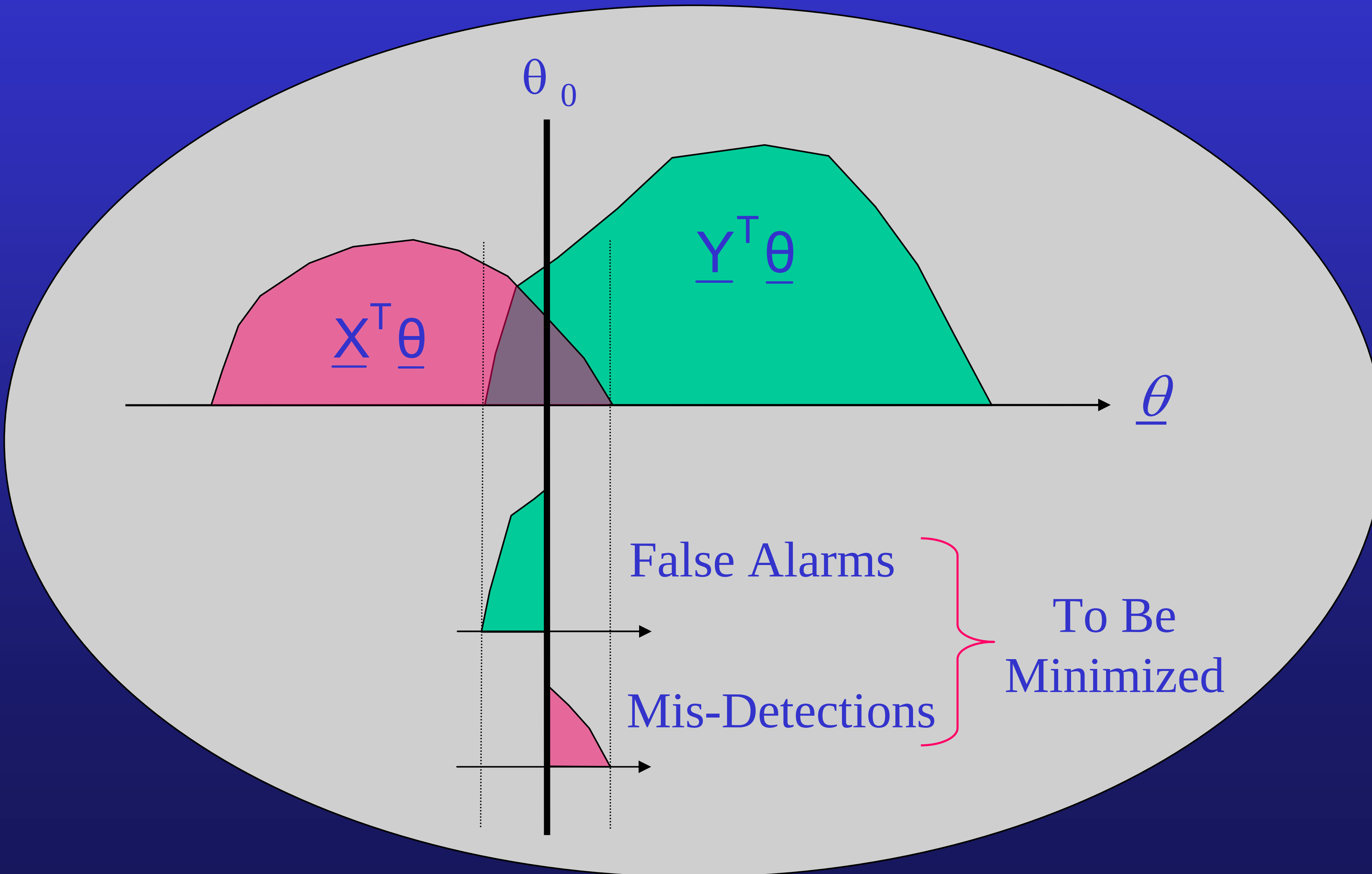
$$C(\underline{Z},\underline{\theta}) = \text{sign}\left\{\underline{Z}^T\underline{\theta} - \theta_0\right\}$$

- The parameters $\{\underline{\theta},\theta_0\}$ define a separating hyper-plane.



$\underline{X}$
$\underline{Y}$
$\underline{\theta}$
$\underline{Z}^T\underline{\theta} < \theta_0$
$\underline{Z}^T\underline{\theta} > \theta_0$
θ_0

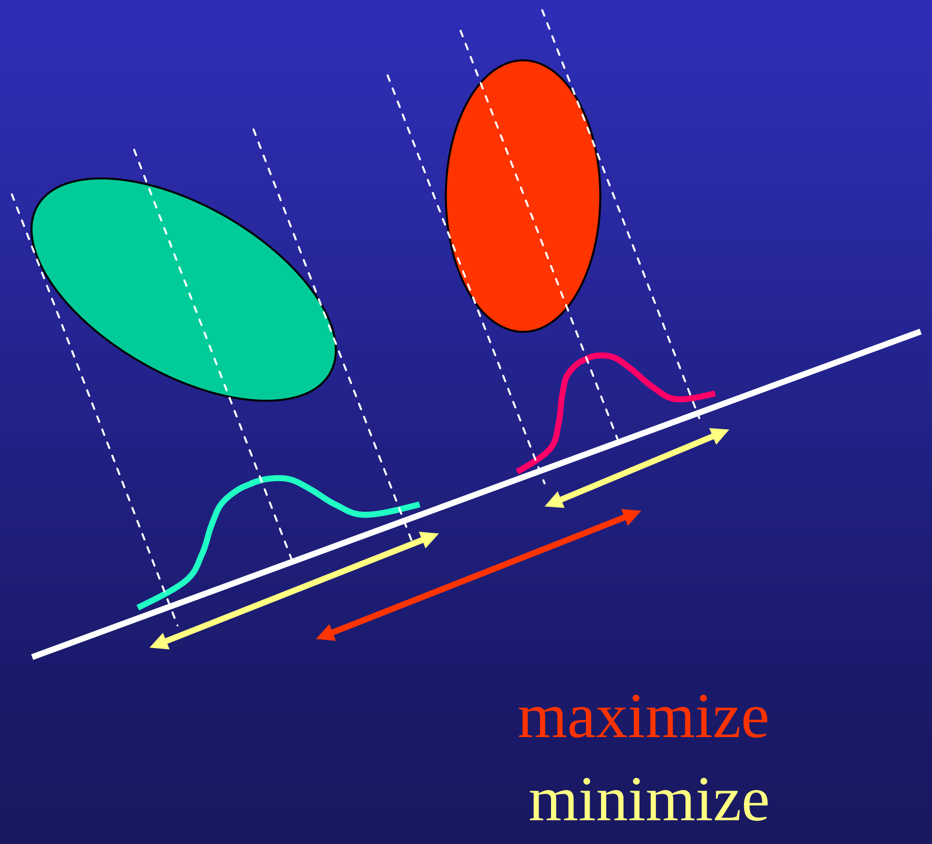
Designing Optimal Linear Classifiers



The Fisher Linear Classifiers

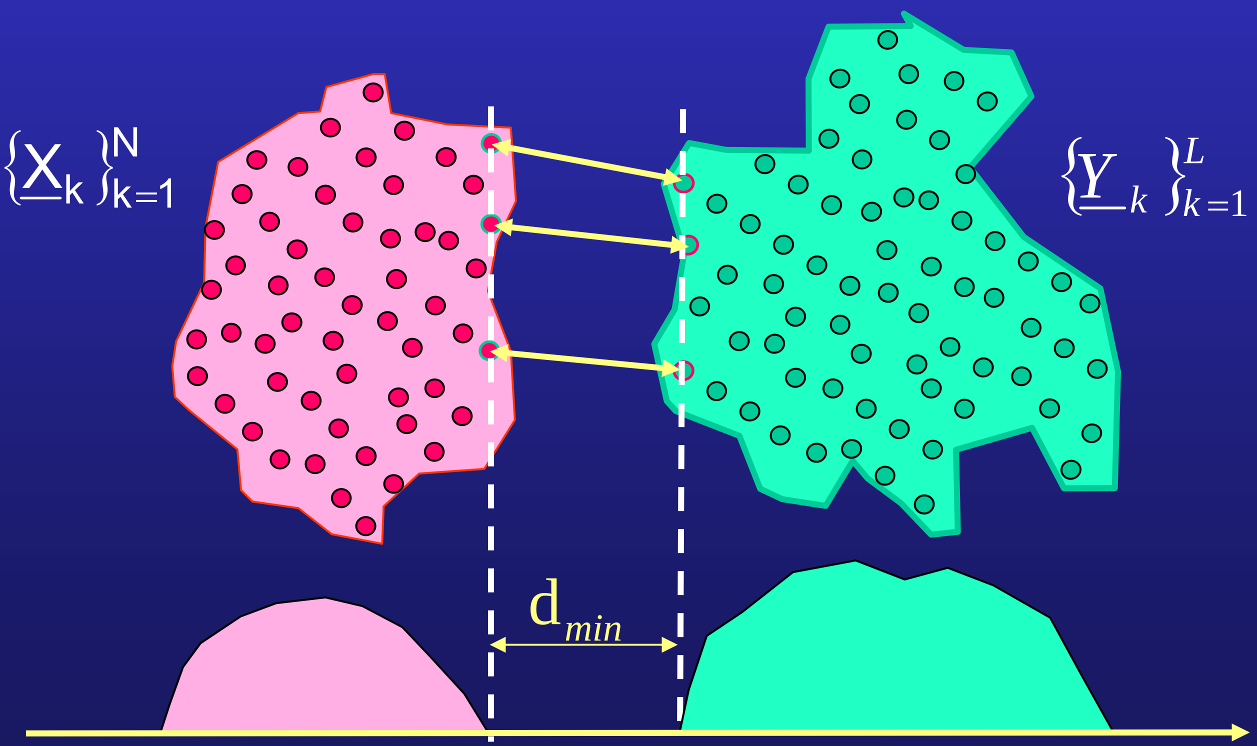
- Choose the projection kernel $\underline{\theta}$ that maximizes the Mahalanobis-distance between $X^T\underline{\theta}$ and $Y^T\underline{\theta}$

$$d_M(i,j) = \frac{\left\|\mu_i - \mu_j\right\|^2}{\sigma_i^2 + \sigma_j^2}$$



The Support Vector Machine (SVM)

Choose the projection kernel $\underline{\theta}$ that maximizes the classes margin d_{min}. (Vapnik-Chernoveskis 82)

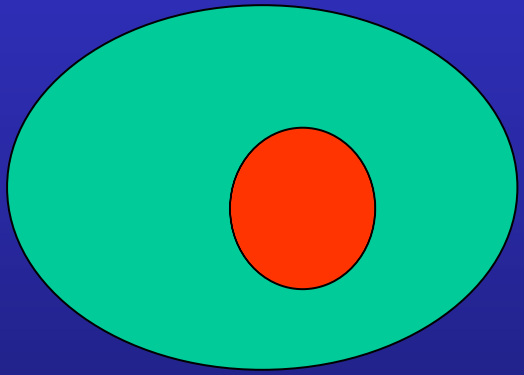


The Support Vector Machine - Continue

- The support vectors are those examples realizing the minimal distance. The decision function is composed of a linear combination of these vectors.
- The optimal projection that emerges turns out to be the solution of a QP problem.
- Generalization error is bounded, and the SVM acheives the tightest bound.

The Limitations of Linear Classifiers

- The above classifiers are suitable for linearly seperable classes (or close to this).
- In other cases:



  - Generalize to non-linear classifiers.
  - Map into higher dimensional feature space.

Complicated !!

Face Detection - Previous Works

- Rowley & Kanade (98), Juel & March (96):

  Neural Network approach - Non linear classifier.

- Sung & Poggio (98):

  Clustering the faces/non-faces into sub-groups, and RBF classifier- Non Linear.

- Osuna, Freund, & Girosi (97):

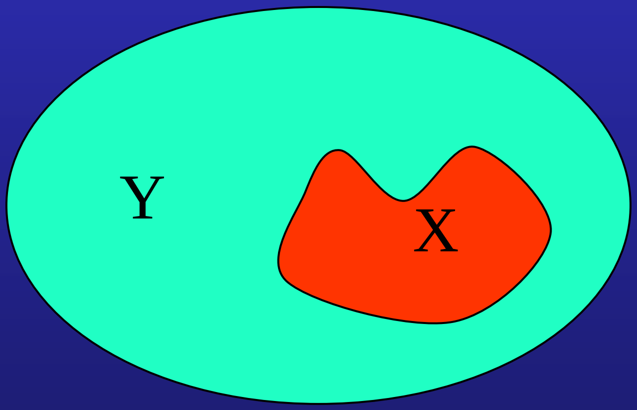
  Support Vector Machine - Classification in high dimensional feature space.

- Keren & Gotsman (98):

  Anti-Faces method - finding a linear kernel that is orthogonal to faces and smooth.

Our Approach - Two Steps Back

- Observations:
  - A typical configuration in Pattern Detection is that the Target class is surrounded by the Clutter class.
  - P{Target}<<P{Clutter}

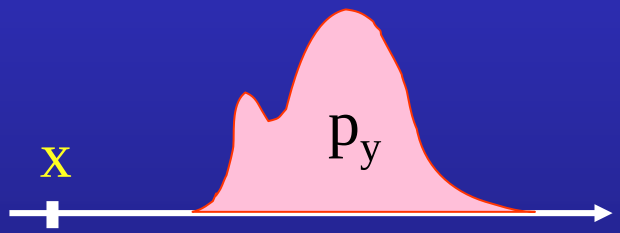


- Conclusions:
  - A pdf separation is not appropriate.
  - Clutter labeling should be performed fast.

Distance Definition:

- Define a distance of a point x from a pdf $p_Y(y)$:

$$D(x, p_Y) = \int_y \frac{(x-y)^2 p_Y(y)}{\sigma_y^2} dy = \frac{(x-\mu_Y)^2 + \sigma_Y^2}{\sigma_Y^2}$$



- Consequally, we define the distance of $p_X(x)$ from $p_Y(y)$:

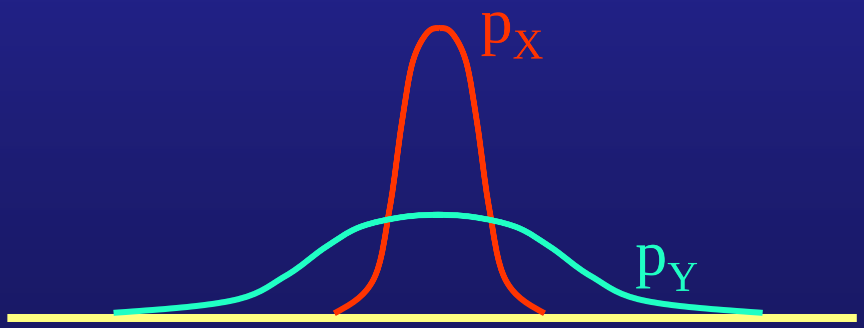
$$D(p_X, p_Y) = \int_x \frac{(x-\mu_Y)^2 + \sigma_Y^2}{\sigma_Y^2} p_X(x) dx = \boxed{\frac{(\mu_X-\mu_Y)^2 + \sigma_X^2 + \sigma_Y^2}{\sigma_Y^2}}$$

$$D(p_X, p_Y) = \frac{(\mu_X - \mu_Y)^2 + \sigma_X^2 + \sigma_Y^2}{\sigma_Y^2}$$

- Alternatively, we can define the proximity of p_x to p_y:

$$Prox(p_X, p_Y) = \frac{\sigma_Y^2}{(\mu_X - \mu_Y)^2 + \sigma_X^2 + \sigma_Y^2}$$

- Note, that the distance is asymmetric.



$D(p_X,p_Y) < D(p_Y,p_X)$:

Optimal Classifier for Pattern Detection

- We would like to find a projection kernel $\underline{\theta}$ which minimizes the overlap between $p_x=p(X^T\underline{\theta})$ and $p_y=p(Y^T\underline{\theta})$:

$$E(\underline{\theta}) = P(X)\frac{\sigma_y^2}{(\mu_x-\mu_y)^2+\sigma_x^2+\sigma_y^2} + P(Y)\frac{\sigma_x^2}{(\mu_x-\mu_y)^2+\sigma_x^2+\sigma_y^2}$$

- If P(X)=P(Y) we obtain the Fisher Linear Classifier.
- In Pattern Detection P(X)<<P(Y), hence we get:

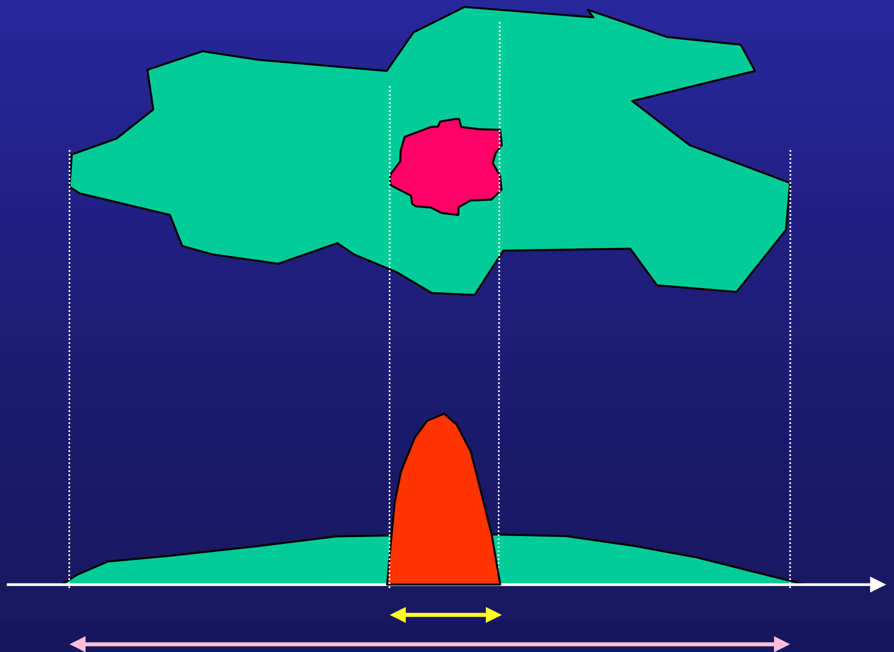
$$\boxed{\underline{\theta} = \arg\min \frac{\sigma_x^2}{(\mu_x-\mu_y)^2+\sigma_x^2+\sigma_y^2}}$$

$$\underline{\theta} = \operatorname{argmin} \frac{\sigma_x^2}{(\mu_x - \mu_y)^2 + \sigma_x^2 + \sigma_y^2}$$

- The penalty term can be minimized by two alternatives:
  - – Maximize the “between class” distance.
  - – Minimize σ_x while maximizing σ_y.
- The second alternative is more common in Pattern Detection.

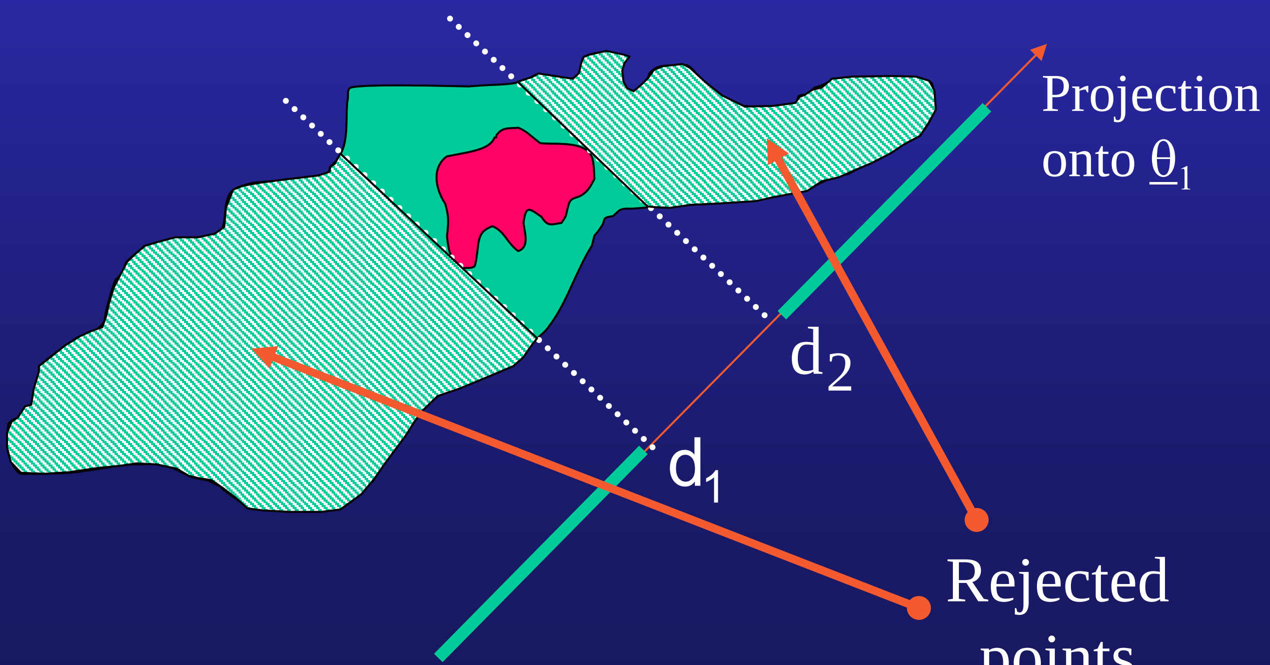
Minimize

Maximize



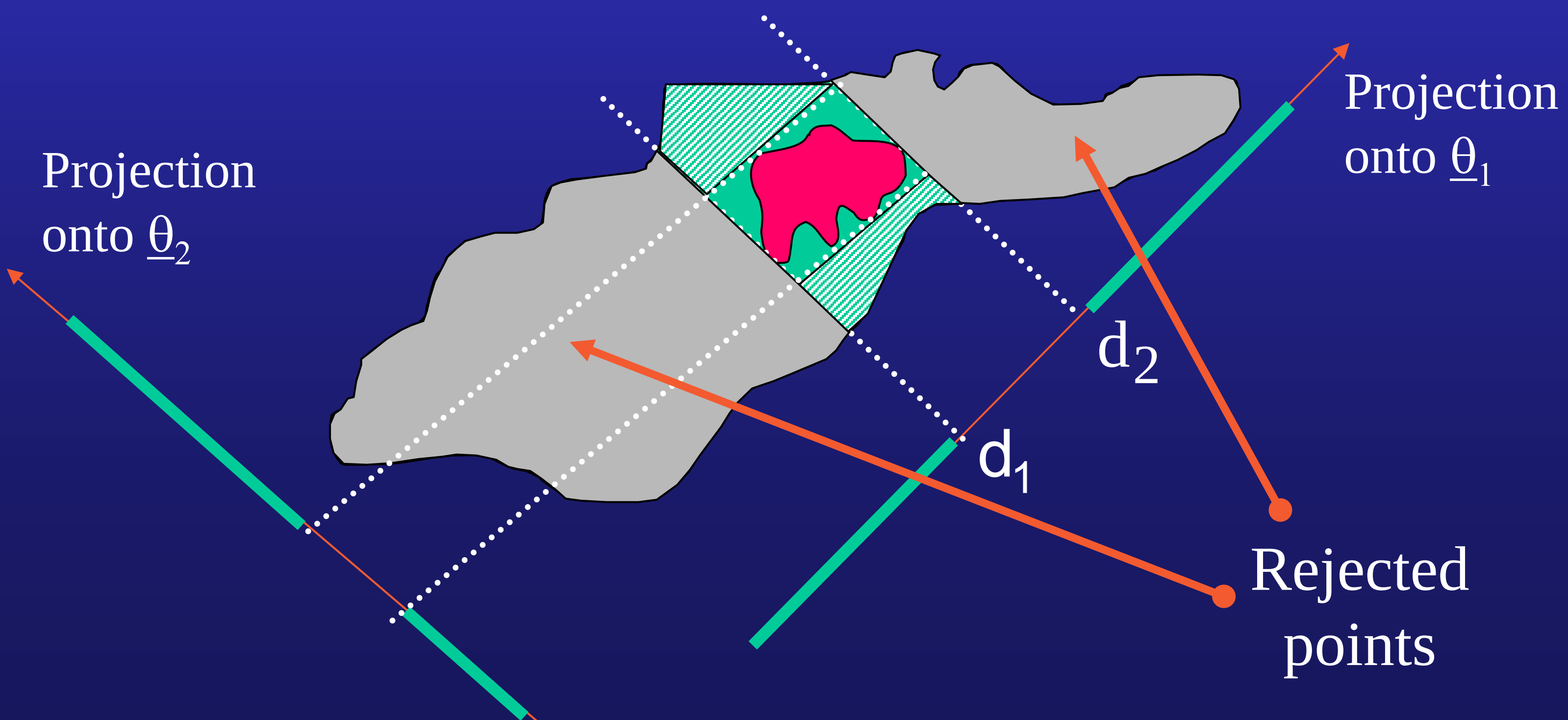
Maximal Rejection

- The optimal $\underline{\theta}$ assures that most $Z \in Y$ are distant from X.
- **Rejection**: two decision levels $[d_1, d_2]$ such that the number of rejected clutters is maximized while finding all targets.



Successive Rejection

- Following the first rejection as many as possible clutters were classified, while targets remain unclassified.
- In order to further reject clutter, we apply the maximal rejection technique to the remaining classes.



Formal Derivation

- In practice we have only samples from P_X and P_Y .
- The class means are estimated:

$$M_x = \frac{1}{L_x}\sum_k X_k \qquad M_y = \frac{1}{L_y}\sum_k Y_k$$

- The class covariances are estimated:

$$S_X^2 = \frac{1}{L_x}\sum_k (X_k - M_x)(X_k - M_x)^T \qquad S_Y^2 = \frac{1}{L_Y}\sum_k (Y_k - M_Y)(Y_k - M_Y)^T$$

- The means and variances after projection onto $\underline{\theta}$ are:

$$\mu_x = \underline{\theta}^T M_x \quad \text{and} \quad \mu_Y = \underline{\theta}^T M_Y$$

$$\sigma_X^2 = \underline{\theta}^T M_x \underline{\theta} \quad \text{and} \quad \sigma_Y^2 = \underline{\theta}^T M_Y \underline{\theta}$$

Formal Derivation - Cont.

- The optimal $\underline{\theta}$ minimizes the following term:

$$E(\underline{\theta}) = \frac{\sigma_x^2}{(\mu_x - \mu_y)^2 + \sigma_x^2 + \sigma_y^2} =$$

$$\boxed{= \frac{\underline{\theta}^T S_x \underline{\theta}}{\underline{\theta}^T \left[(M_Y - M_X)(M_Y - M_X)^T + S_x + S_Y \right] \underline{\theta}} = \frac{\underline{\theta}^T A \underline{\theta}}{\underline{\theta}^T B \underline{\theta}}}$$

- This term can be rewritten as a generalized eigenvalue problem:

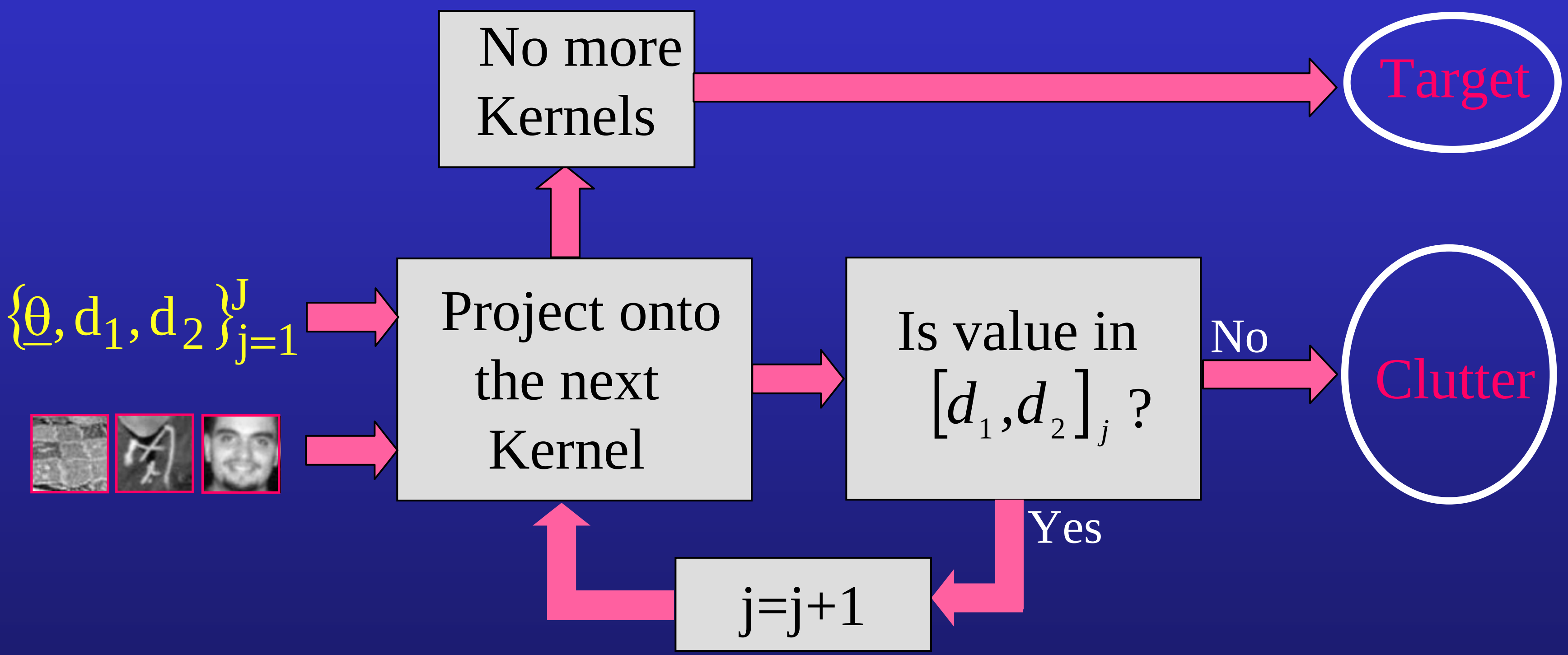
$$A\underline{\theta} = \lambda B \underline{\theta}$$

- The solution is the eigenvector corresponding to the smallest eigenvalue.

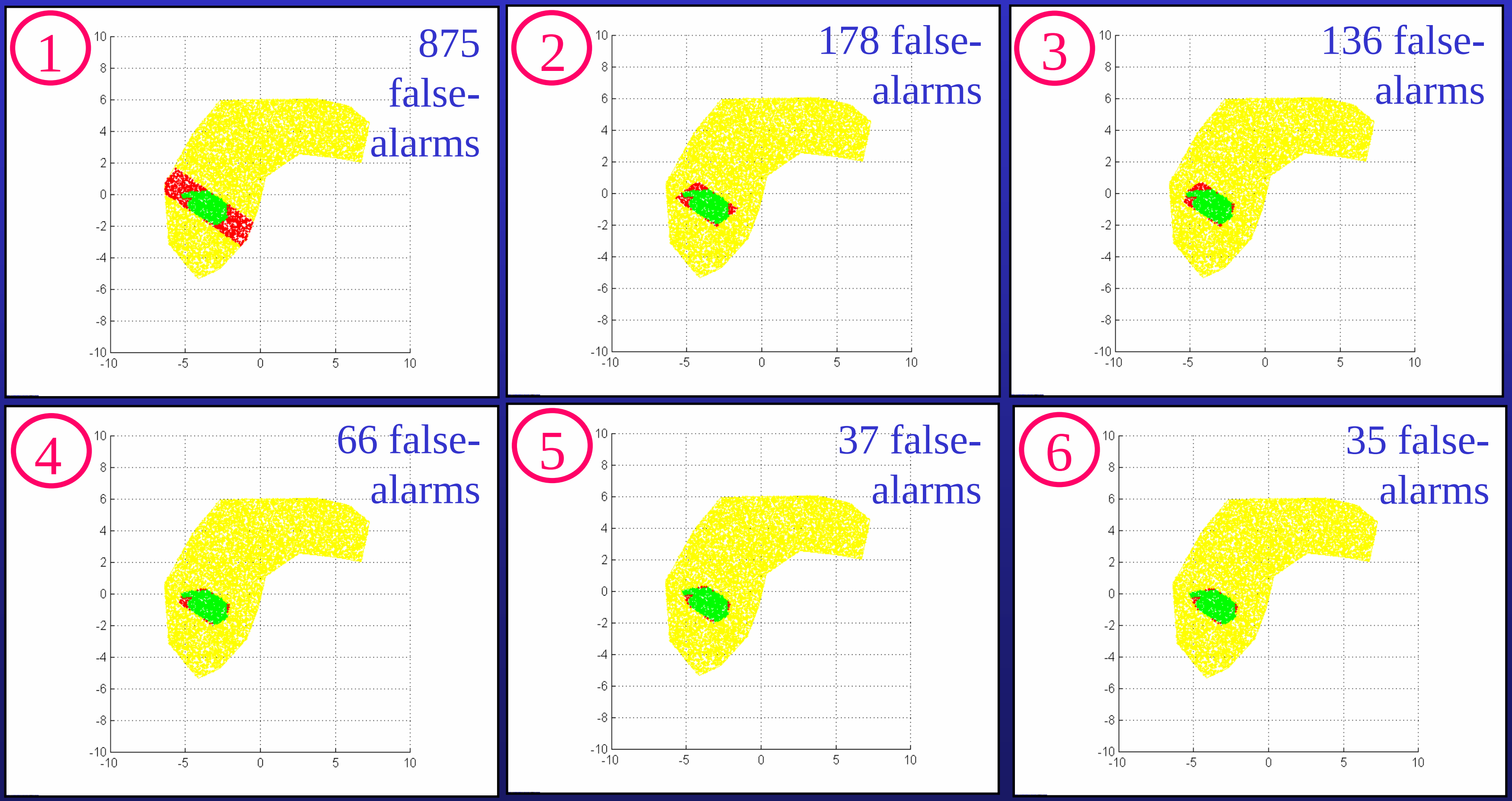
Proposed Algorithm

- There are two phases to the algorithm:
  - Training: Compute the projection kernels, and their thresholds. This process is performed ONCE and off line.
  - Testing: Given an image, find targets using the above found kernels and thresholds.

Testing Stage



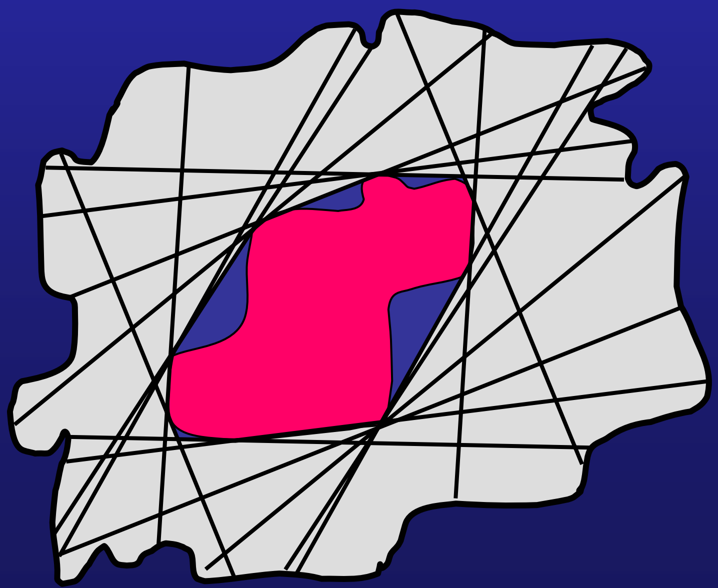
2D example



1
875 false-alarms
2
178 false-alarms
3
136 false-alarms
4
66 false-alarms
5
37 false-alarms
6
35 false-alarms

Limitations

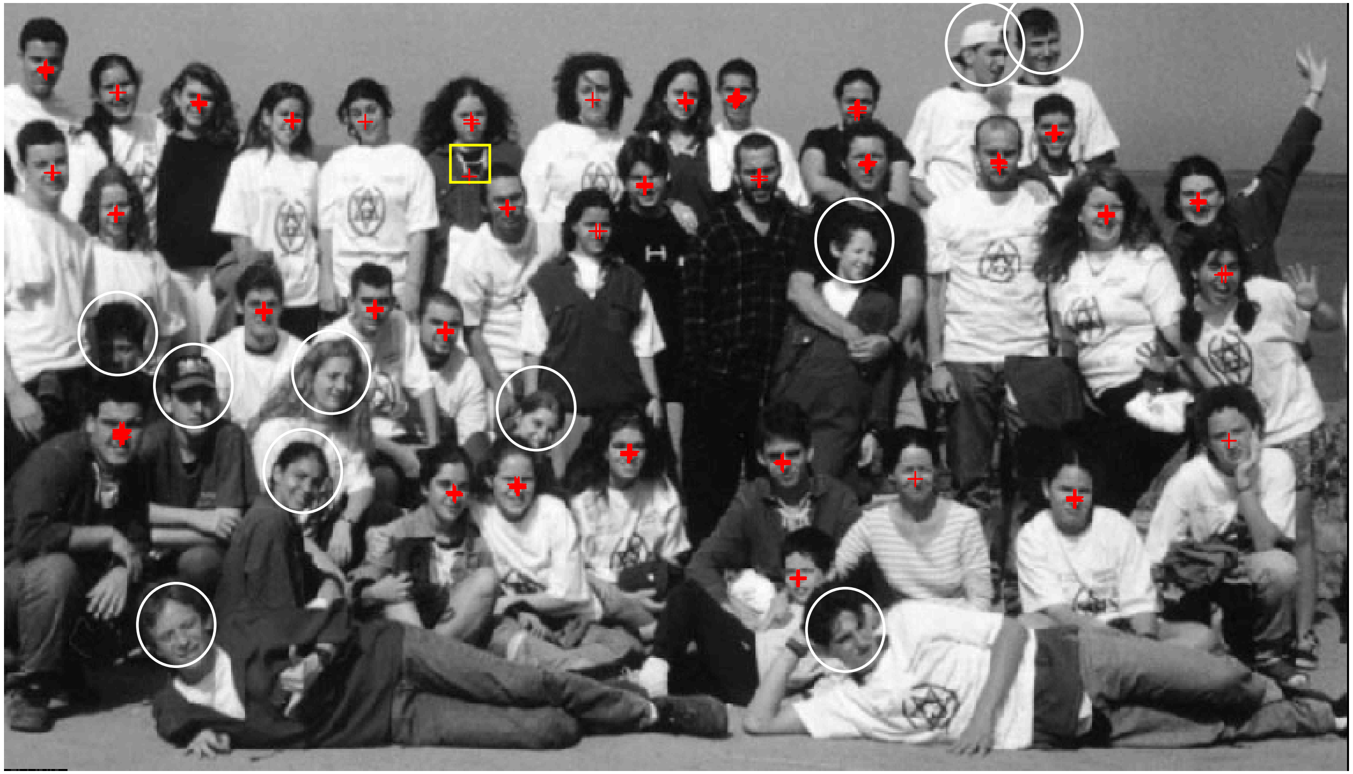
- The discriminated zone is a parallelogram polytope. Thus, if the target set is non-convex, zero false alarm discrimination is impossible!!
- Even if the target-set is convex, convergence to zero false-alarms is not guaranteed.



Face Detection Results

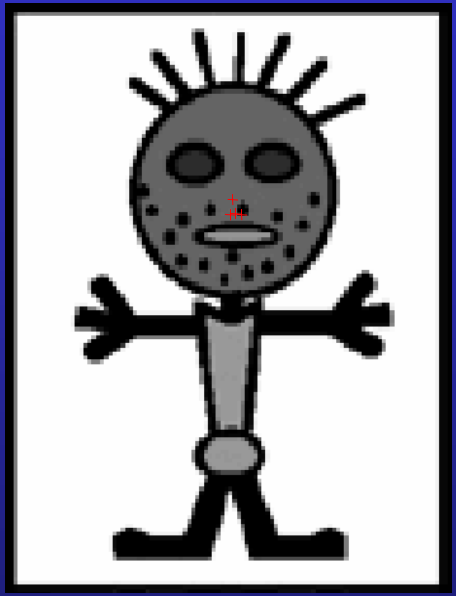
- Kernels for finding *faces* (15x15) and *eyes* (7x15).
- Searching for eyes and faces sequentially - very efficient!
- Face DB: 204 images of 40 people (ORL-DB). Each image is also rotated ±5° and vertically flipped. This produced 1224 Face images.
- Non-Face DB: 54 images - All the possible positions in all resolution layers and vertically flipped - about 40E6 non-face images.

Results



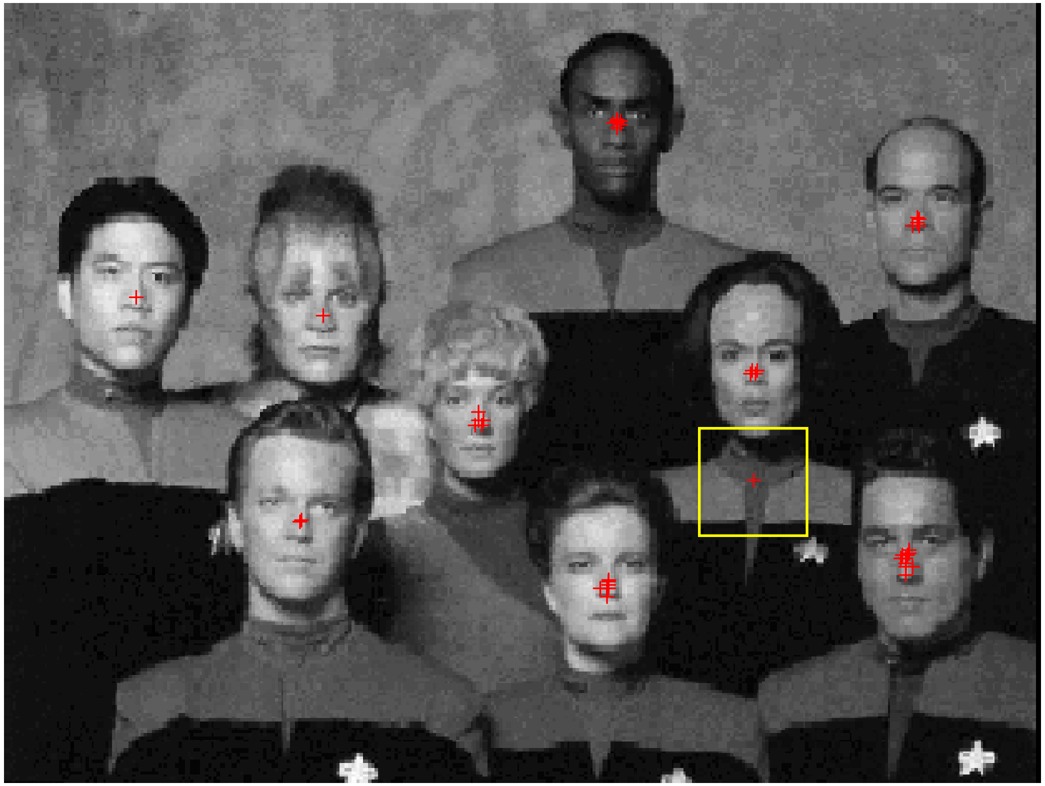
Out of 44 faces, 10 faces are undetected, and 1 false alarm
(the undetected faces are circled - they are either rotated or shadowed)

Results



All faces detected with no false alarms

Results



All faces detected with 1 false alarm(looking closer, this false alarm can be considered as a face)

Complexity

- For a set of 15 kernels (with appropriate decision levels), the first kernel typically removes about 90% of the pixels from further consideration. Other kernels give typically a rejection factor of 50%.
- Thus, the algorithm requires slightly more that one convolution of the image with a kernel (per each resolution layer).
- The algorithm gives very good results (probability of detection and false alarm rate) for the tests we did on frontal faces in images.

Relation to Anti-Faces (Keren & Gotsman 98)

- Projection kernels are in Null(X) and smooth.
- This can be seen as a case where the pdf of the Clutter class is defined parametrically

$$P_Y(Z) \approx e^{-Z^T \left(D^T D\right) Z}$$

where D is a derivative operator.

- In this case $S_Y=(D^TD)^{-1}$, where in the Fourier basis

$$(D^TD)^{-1} \Rightarrow \ \text{diag}(1\ ,\ 1/2^2\ ,\ 1/3^2\ ,\ \ldots)$$

- If $M_x \approx M_y$, minimizing the following term tends to give $\underline{\theta} \in \text{Null}(S_x)$, and smooth:

$$E(\underline{\theta}) = \frac{\underline{\theta}^T S_x \underline{\theta}}{\underline{\theta}^T \left[(M_Y - M_X)(M_Y - M_X)^T + S_x + S_Y\right] \underline{\theta}}$$

Conclusions

- MRC: projection onto pre-trained kernels, and thresholding. The process is a rejection based classification.
- Appropriate for pattern detection where pdf separation is impossible.
- Exploits the fact that P(clutter)>>P(target)
- Gives a very fast clutter labeling at the expense of slow target labeling.
- Can also deal with non linearly separable classes (convexly separable).
- Simple to apply (linear), with promising results for face-detection in images.
- A generalization of the Fisher Linear Classifier.

END