Rejection Based Face Detection

Michael Elad*

Scientific Computing and Computational Mathematics

Stanford University

(Gates 282, phone: 723-7685, email: elad@sccm.stanford.edu)

May 13th, 2002

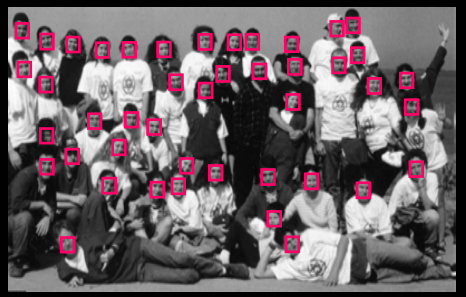
* Collaboration with Y. Hel-Or and R. Keshet

Chapter 1

Problem Definition

1.1 Requirements

Face Finder



Detect FRONTAL & VERTICAL faces:

- All spatial position, all scales
- any person, any expression
- Robust to illumination conditions
- Old/young, man/woman, hair, glasses.

Design Goals:

- Fast algorithm
- Accurate (False alarms/ mis-detections)

1.2 Face Examples

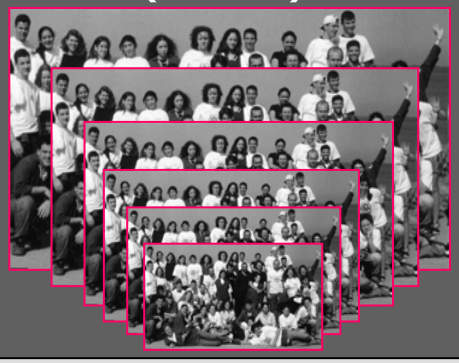
Taken from the ORL Database

1.3 Position & Scale

Input Image

Suspected Face Positions

Compose a pyramid with 1:f resolution ratio (f=1.2)



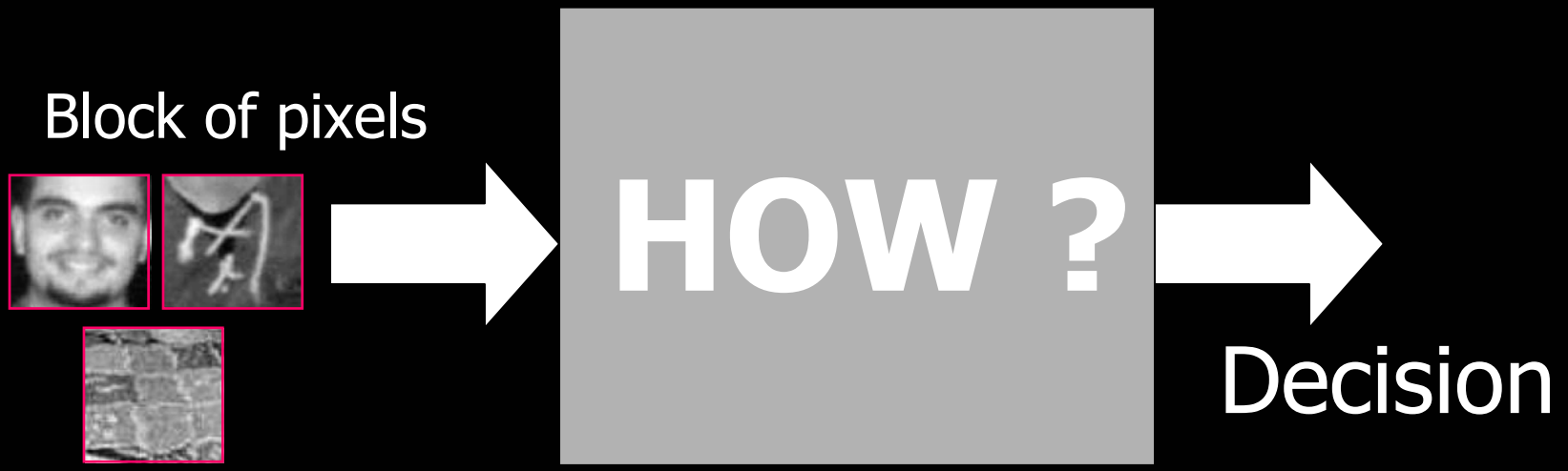
Draw L*L blocks from each location and in each resolution layer

Classifier

Face Finder

1.4 Classifier?

The classifier gets blocks of fixed size L^2 (say 15^2) pixels, and returns a decision (Face/Non-Face)



Chapter 2

Example-Based Classification

2.1 Definition

A classifier is a parametric (J parameters) function $C(\underline{Z},\underline{\theta})$ of the form

$$C\{\underline{Z},\underline{\theta}\}:\ \Re^{L^2}\times\Re^{J}\rightarrow\{-1,+1\}$$

Example: For blocks of 4 pixels $\underline{Z}=[z_1,z_2,z_3,z_4]$,

we can assume that $C(\underline{Z})$ is obtained by

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

2.2 Classifier Design

$$C\{\underline{Z},\underline{\theta}\}:\ \Re^{L^2}\times\Re^{J}\rightarrow\{-1,+1\}$$

In order to get a good quality classifier, we have to answer two questions:

1. What parametric form to use? Linear or non-linear? What kind of non-linear? Etc.
2. Having chosen the parametric form, how do we find appropriate $\underline{\theta}$?

2.3 Complexity

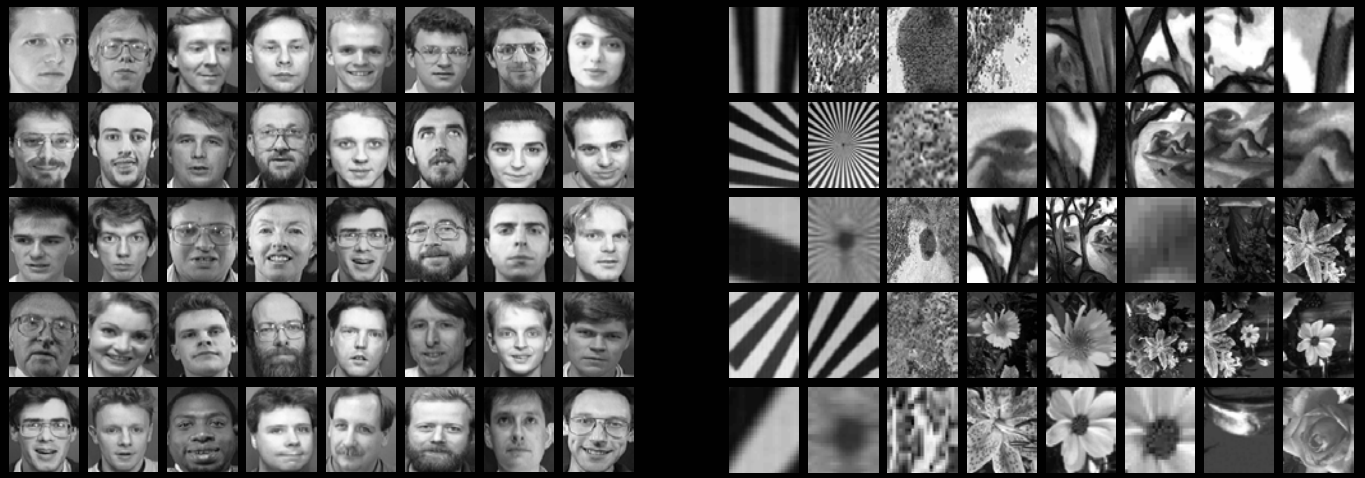
Searching faces in a given scale, for a 1000 by 2000 pixels image, the classifier is applied 2e6 times

THE ALGORITHMS' COMPLEXITY IS GOVERNED BY THE CLASSIFIER PARAMETRIC FORM

2.4 Examples

Collect examples of Faces and Non-Faces

Faces: $\{\underline{X}_k\}_{k=1}^{N_X}$, Non-Faces: $\{\underline{Y}_k\}_{k=1}^{N_Y}$



Obviously, we should have $N_X << N_Y$

2.5 Training

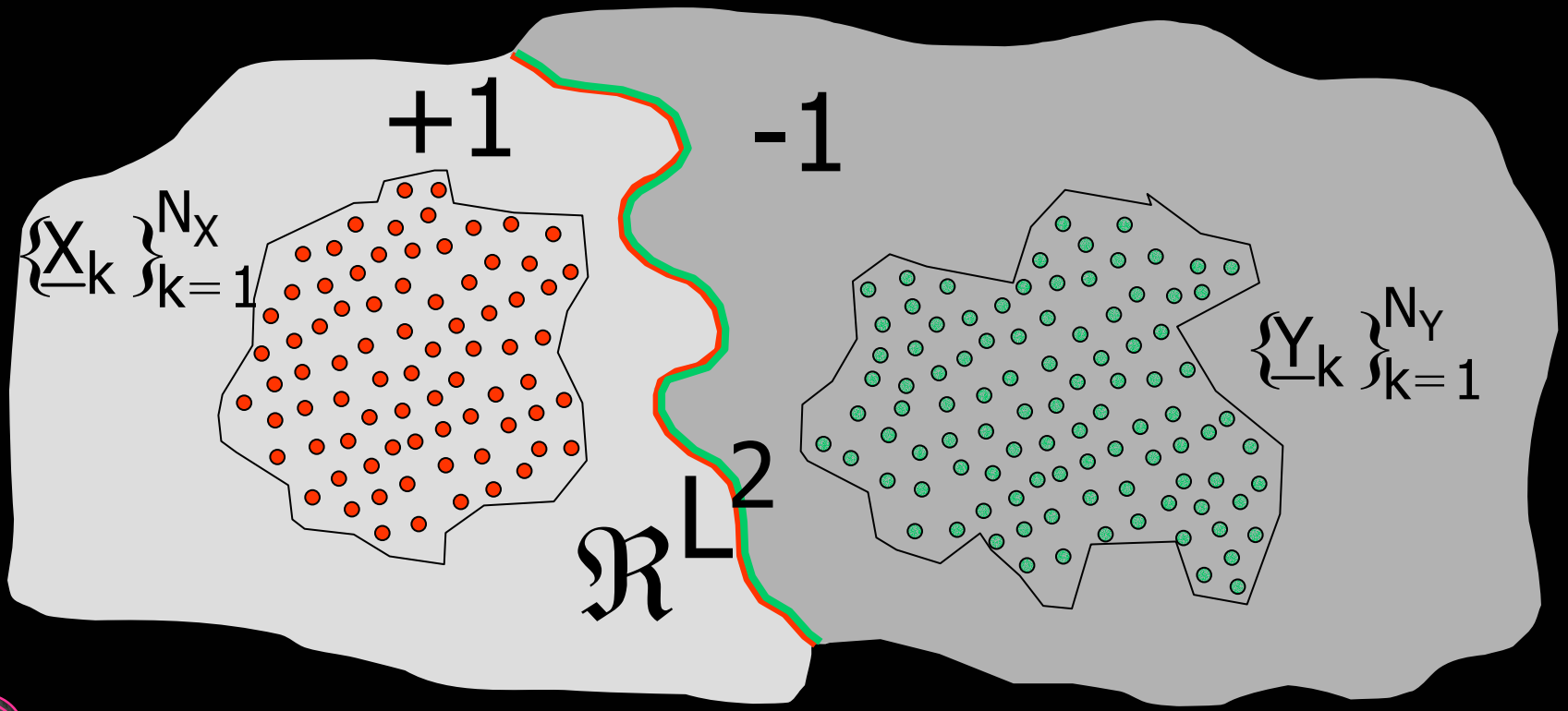
The basic idea is to find $\underline{\theta}$ such that

$$\forall 1 \le k \le N_X,\ C\left\{\underline{X}_k, \underline{\theta}\right\} = +1$$
$$\forall 1 \le k \le N_Y,\ C\left\{\underline{Y}_k, \underline{\theta}\right\} = -1$$

or with few errors only, and with good
GENERALIZATION ERROR

2.6 Geometric View

C($\underline{Z}$,$\underline{\theta}$) is to drawing a separating manifold between the two classes



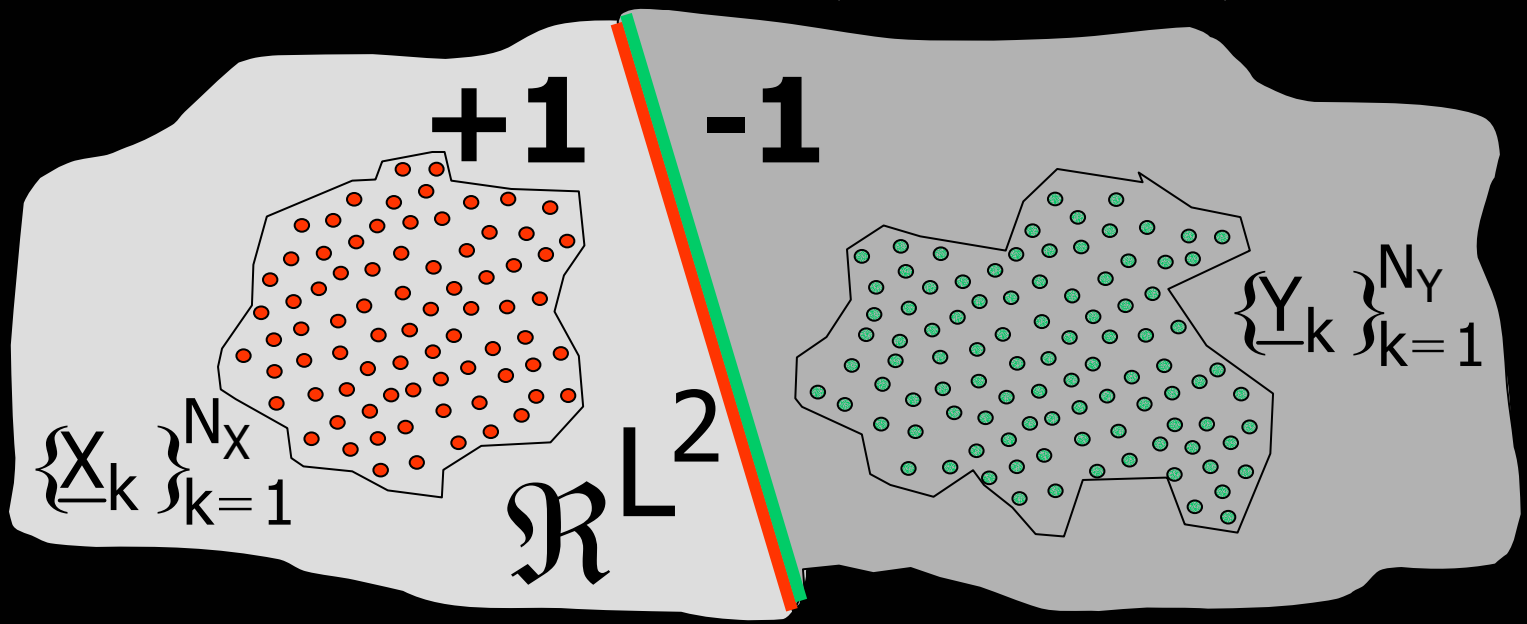
Chapter 3

Linear Classification

3.1 Linear Separation

The Separating manifold is a Hyper-plane

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

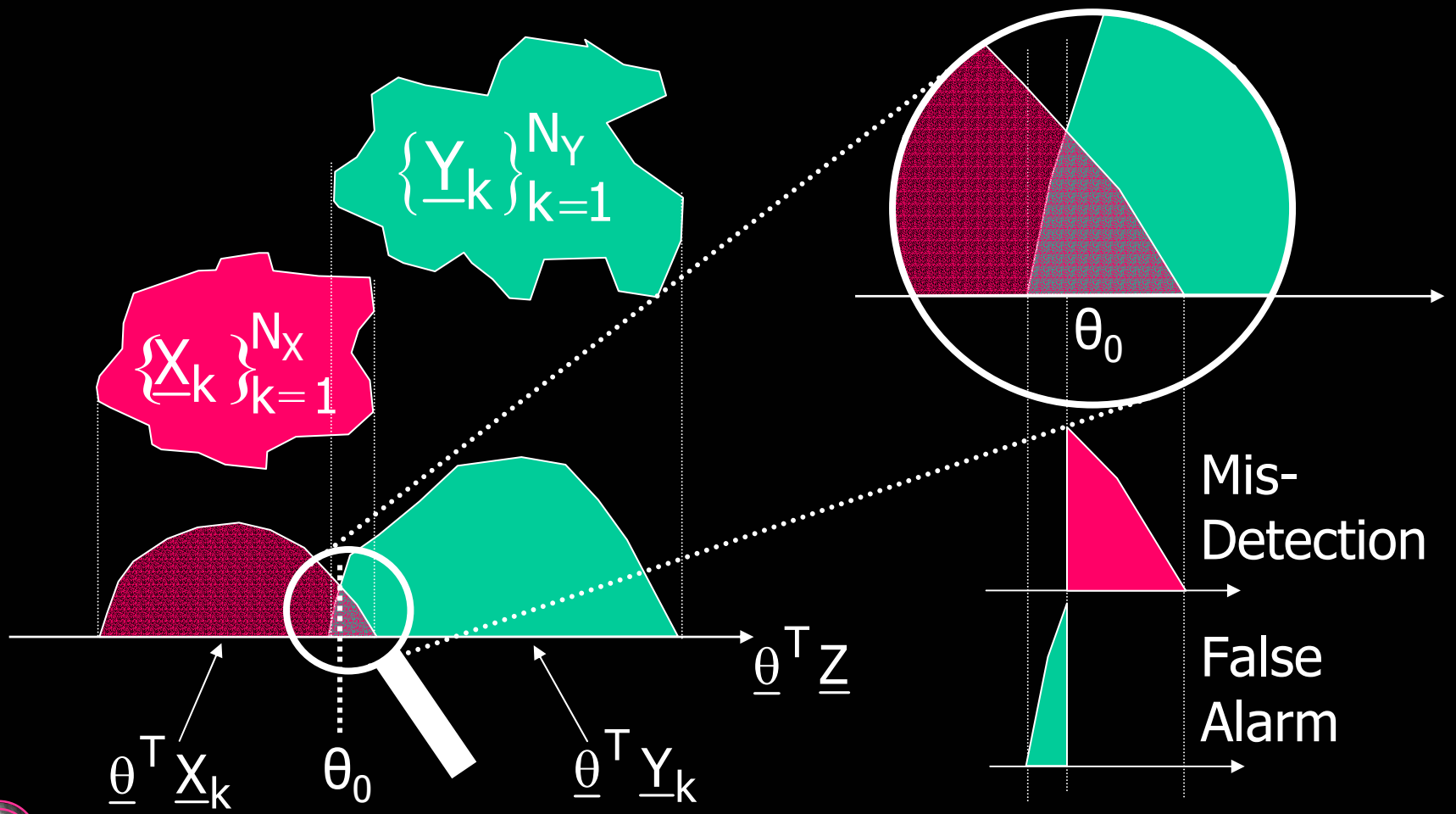


3.2 Projection

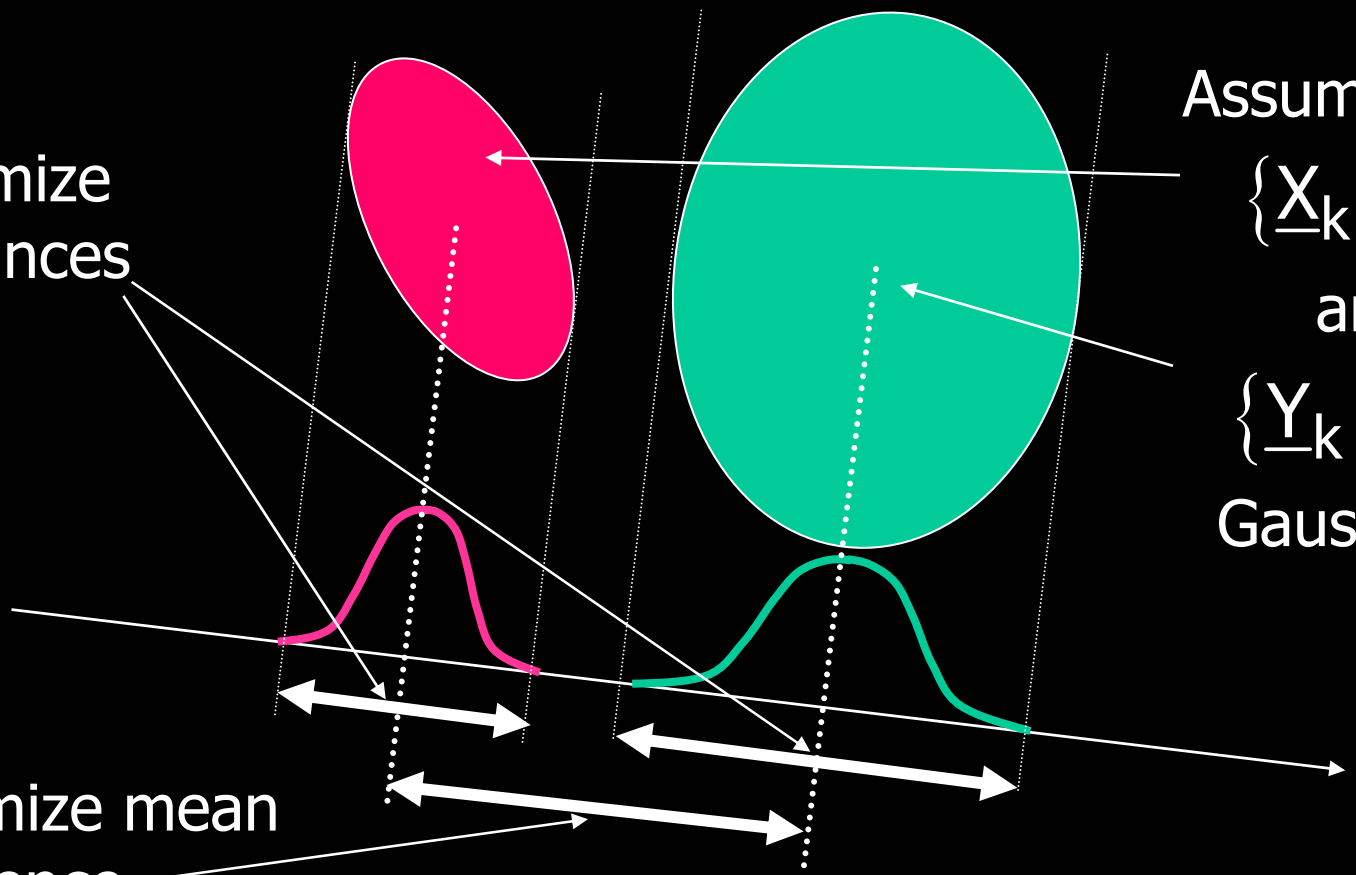
Projecting every block in this image onto a kernel $\underline{\theta}$ is a

CONVOLUTION

3.3 Linear Operation



3.4 FLD*



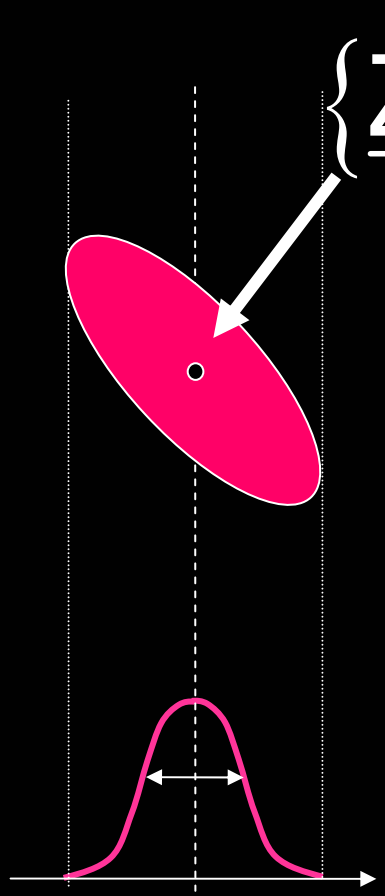
Assume that $\{\underline{X}_k\}_{k=1}^{N_X}$ and $\{\underline{Y}_k\}_{k=1}^{N_Y}$ Gaussians

Minimize variances

Maximize mean difference

*FLD - Fisher Linear Discriminant

3.5 Projected Moments



$$\{\underline{Z}_k\}_{k=1}^{N}$$

$$\underline{M} = \frac{1}{N}\sum_{k=1}^{N}\underline{Z}_k$$

$$\mathbf{R} = \frac{1}{N}\sum_{k=1}^{N}\left[\underline{Z}_k - \underline{M}\right]\left[\underline{Z}_k - \underline{M}\right]^T$$

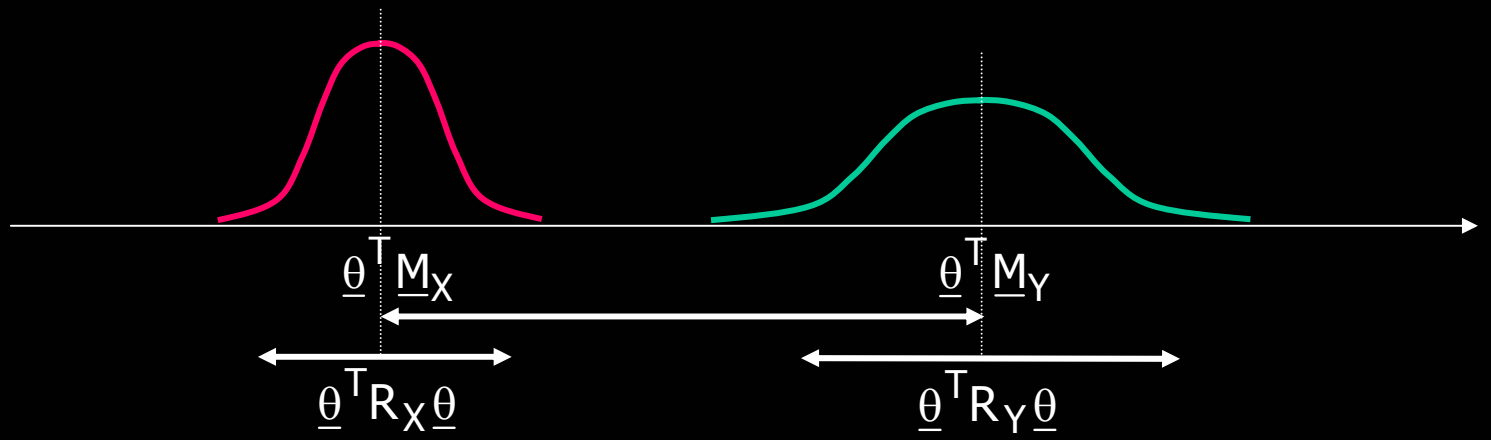
$$z_k = \underline{\theta}^T\underline{Z}_k \Rightarrow m = \underline{\theta}^T\underline{M},\ r = \underline{\theta}^T\mathbf{R}\underline{\theta}$$

3.6 FLD Formally

Maximize

$$f\{\underline{\theta}\} = \frac{\left[\underline{\theta}^T \underline{M}_X - \underline{\theta}^T \underline{M}_Y\right]^2}{\underline{\theta}^T R_X \underline{\theta} + \underline{\theta}^T R_Y \underline{\theta}} = \frac{\underline{\theta}^T \left[\underline{M}_X - \underline{M}_Y\right]\left[\underline{M}_X - \underline{M}_Y\right]^T \underline{\theta}}{\underline{\theta}^T \left[R_X + R_Y\right] \underline{\theta}}$$

Minimize



3.7 Solution

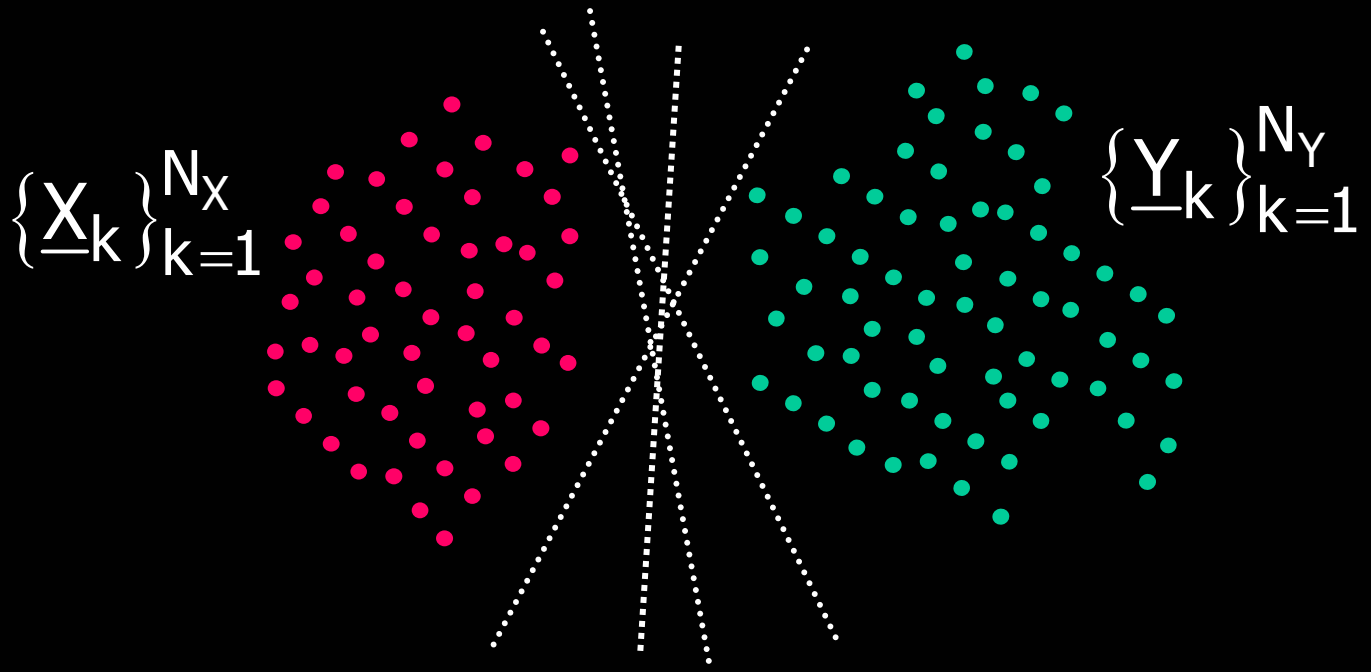
$$\underset{\|\underline{\theta}\|^2=1}{\text{Maximize}} \quad \varepsilon^2(\underline{\theta}) = \frac{\underline{\theta}^T \mathbf{R} \underline{\theta}}{\underline{\theta}^T \mathbf{Q} \underline{\theta}}$$

$$\frac{\partial \varepsilon^2(\underline{\theta})}{\partial \underline{\theta}} = \frac{\left(\underline{\theta}^T \mathbf{Q} \underline{\theta}\right) \mathbf{R} \underline{\theta} - \left(\underline{\theta}^T \mathbf{R} \underline{\theta}\right) \mathbf{Q} \underline{\theta}}{\left(\underline{\theta}^T \mathbf{Q} \underline{\theta}\right)^2} = 0 \quad \Rightarrow \quad \mathbf{R} \underline{\theta} = \frac{\left(\underline{\theta}^T \mathbf{R} \underline{\theta}\right)}{\left(\underline{\theta}^T \mathbf{Q} \underline{\theta}\right)} \mathbf{Q} \underline{\theta}$$

Generalized Eigenvalue Problem: $R\underline{v}=\lambda Q\underline{v}$

⇒ The solution is the eigenvector which corresponds to the largest eigenvalue

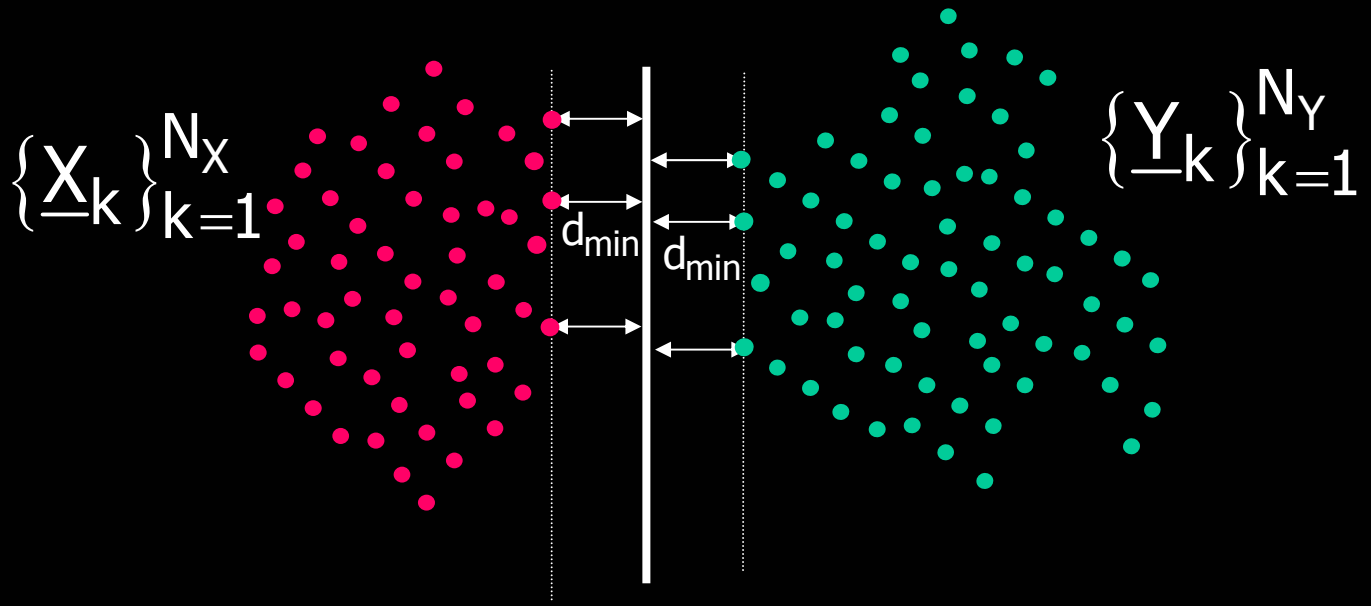
3.8 SVM*



Among the many possible solutions, choose the one which Maximizes the minimal distance to the two example-sets

*SVM - Support Vector Machine

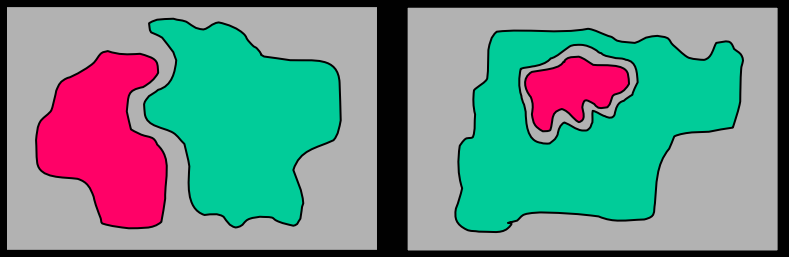
3.9 Support Vectors



1. The optimal $\underline{\theta}$ turns out to emerge as the solution of a Quadratic-Programming problem
2. The support vectors are those realizing the minimal distance. Those vectors define the decision function

3.10 Drawback

Linear methods are not suitable



1. Generalize to non-linear Discriminator by either mapping into higher dimensional space, or using kernel functions
2. Apply complex pre-processing on the block

Complicated Classifier!!

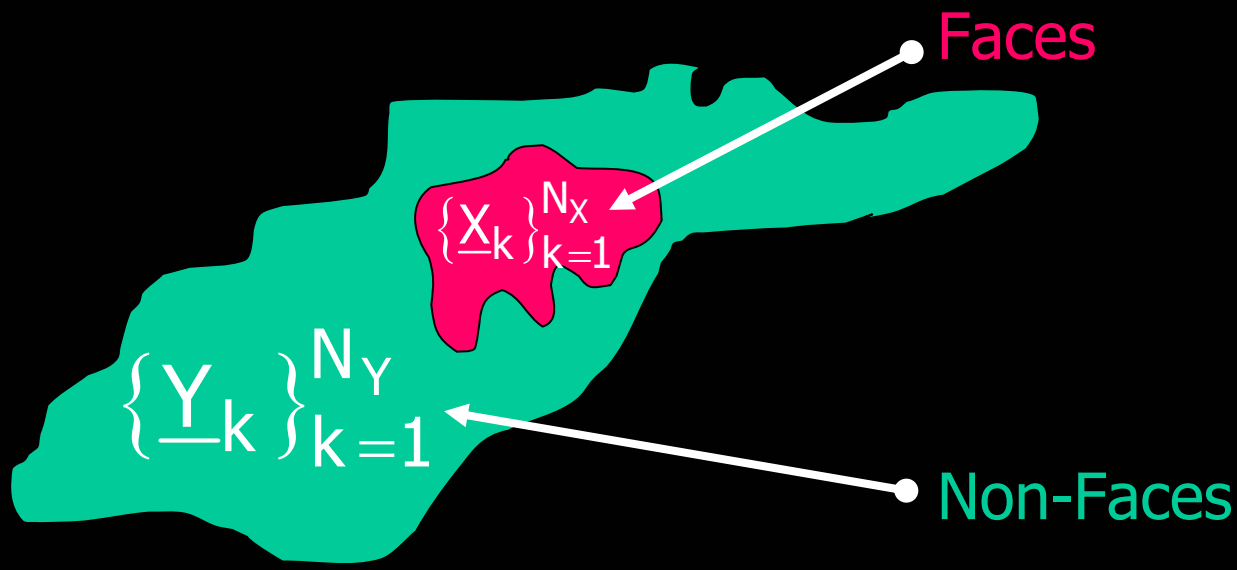
3.11 Previous Work

- Rowley & Kanade (1998), Juel & March (96):
  Neural Network approach - NL
- Sung & Poggio (1998):
  Clustering into sub-groups, and RBF - NL
- Osuna, Freund, & Girosi (1997):
  Support Vector Machine & Kernel functions - NL
- Osdachi, Gotsman & Keren (2001):
  Anti-Faces method - particular case of our work
- Viola & Jones (2001):
  Similar to us (rejection, simplified linear class., features)

Chapter 4

Maximal Rejection Classification

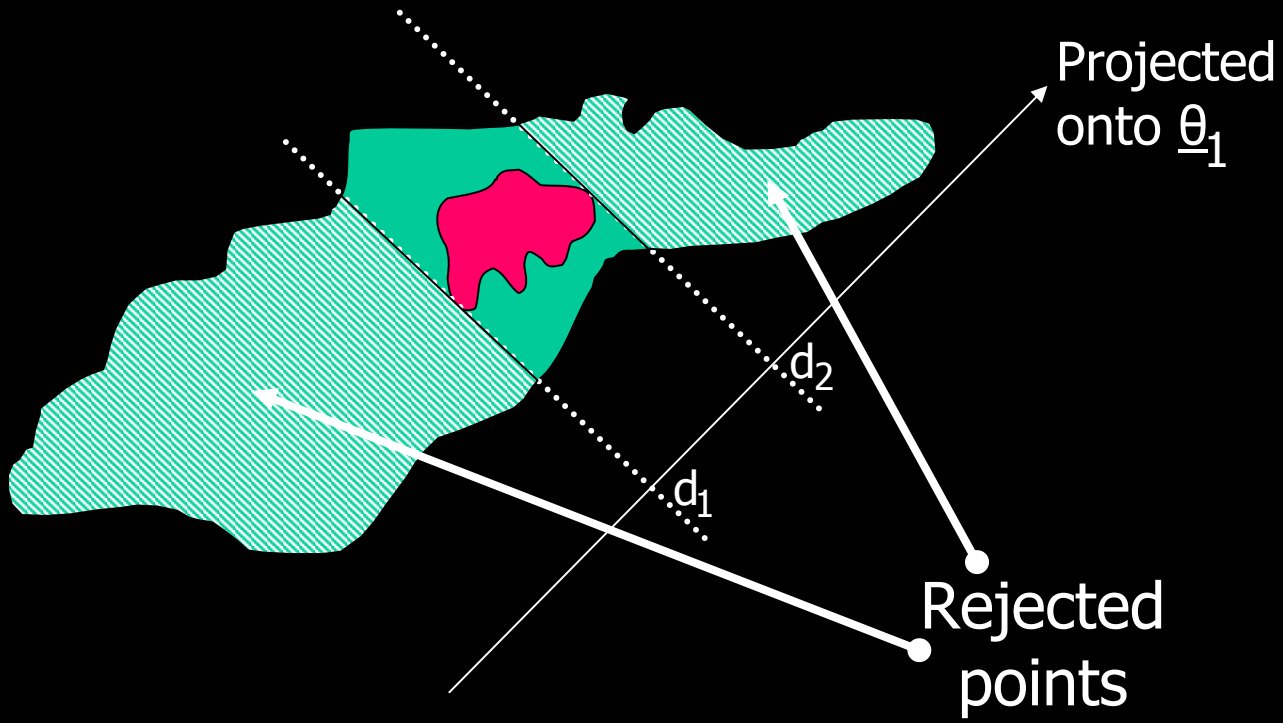
4.1 Model Assumption



We think of the non-faces as a much richer set and more probable, which may even surround the faces set

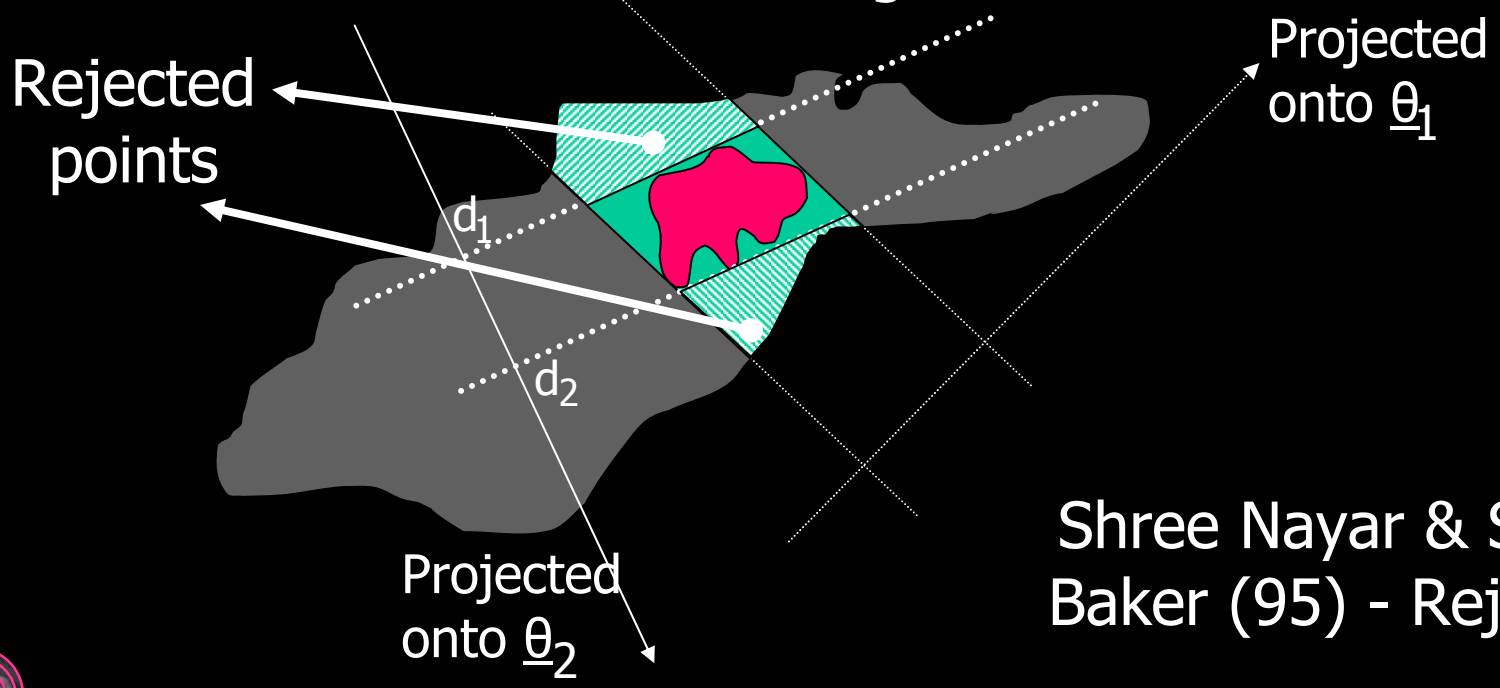
4.2 Maximal Rejection

Find $\underline{\theta}_1$ and two decision levels $[d_1, d_2]_1$ such that the number of rejected non-faces is maximized while finding all faces



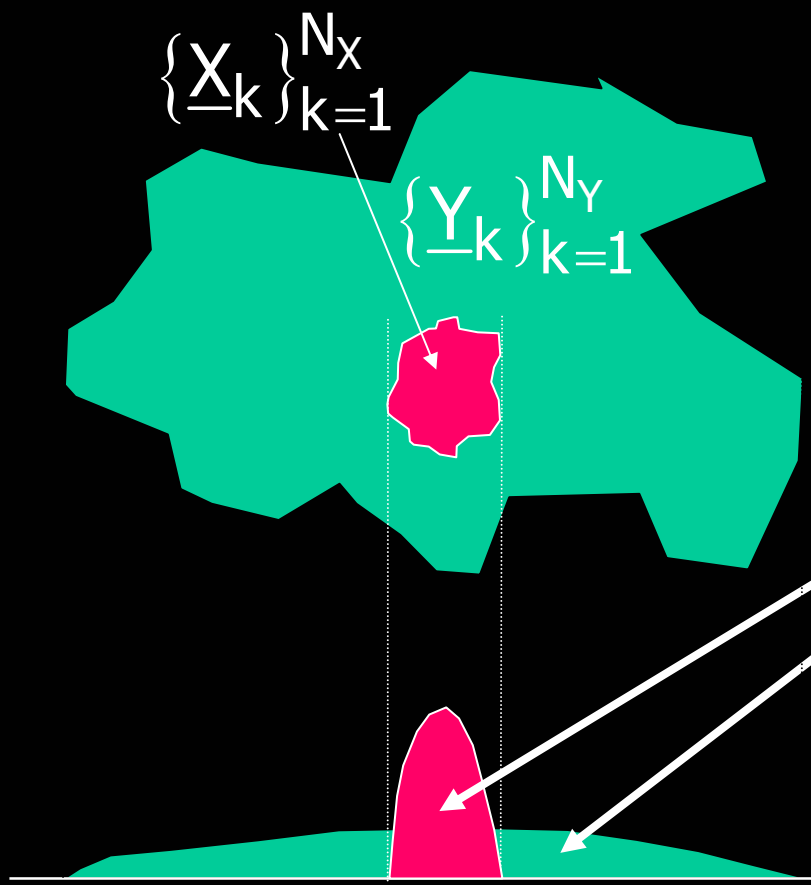
4.3 Iterations

Taking ONLY the remaining non-faces:
Find $\underline{\theta}_2$ and two decision levels $\left[d_1, d_2\right]_2$ such that the number of rejected non-faces is maximized while finding all faces



Shree Nayar & Simon Baker (95) - Rejection

4.4 Formal Derivation



Maximal Rejection

⇕

Maximal distance between these two PDF's

⇓

We need a measure for this distance which will be appropriate and easy to use

4.5 Design Goal

Maximize the following function:

$$f\{\underline{\theta}\} = \frac{\sum_{j=1}^{N_Y}\sum_{k=1}^{N_X}\left[\underline{\theta}^T\underline{X}_k - \underline{\theta}^T\underline{Y}_j\right]^2}{\sum_{j=1}^{N_X}\sum_{k=1}^{N_X}\left[\underline{\theta}^T\underline{X}_k - \underline{\theta}^T\underline{X}_j\right]^2} = C\cdot\frac{\underline{\theta}^T\mathbf{R}\underline{\theta}}{\underline{\theta}^T\mathbf{Q}\underline{\theta}} =$$

A Reighly Quotient

Maximize the distance between all the pairs of [face, non-face]

Minimize the distance between all the pairs of [face, face]

4.6 Objective Function

$$f\{\underline{\theta}\} = \frac{\underline{\theta}^T \left\{ \left[\underline{M}_X - \underline{M}_Y\right]\left[\underline{M}_X - \underline{M}_Y\right]^T + R_X + R_Y \right\} \underline{\theta}}{\underline{\theta}^T R_X \underline{\theta}}$$

- The expression we got finds the optimal kernel $\underline{\theta}$ in terms of the 1st and the 2nd moments only.
- The solution effectively project all the X examples to a near-constant value, while spreading the Y's away - $\underline{\theta}$ plays a role of an approximated invariant of the faces
- As in the FLD, the solution is obtained problem is a Generalized Eigenvalue Problem.

Chapter 5

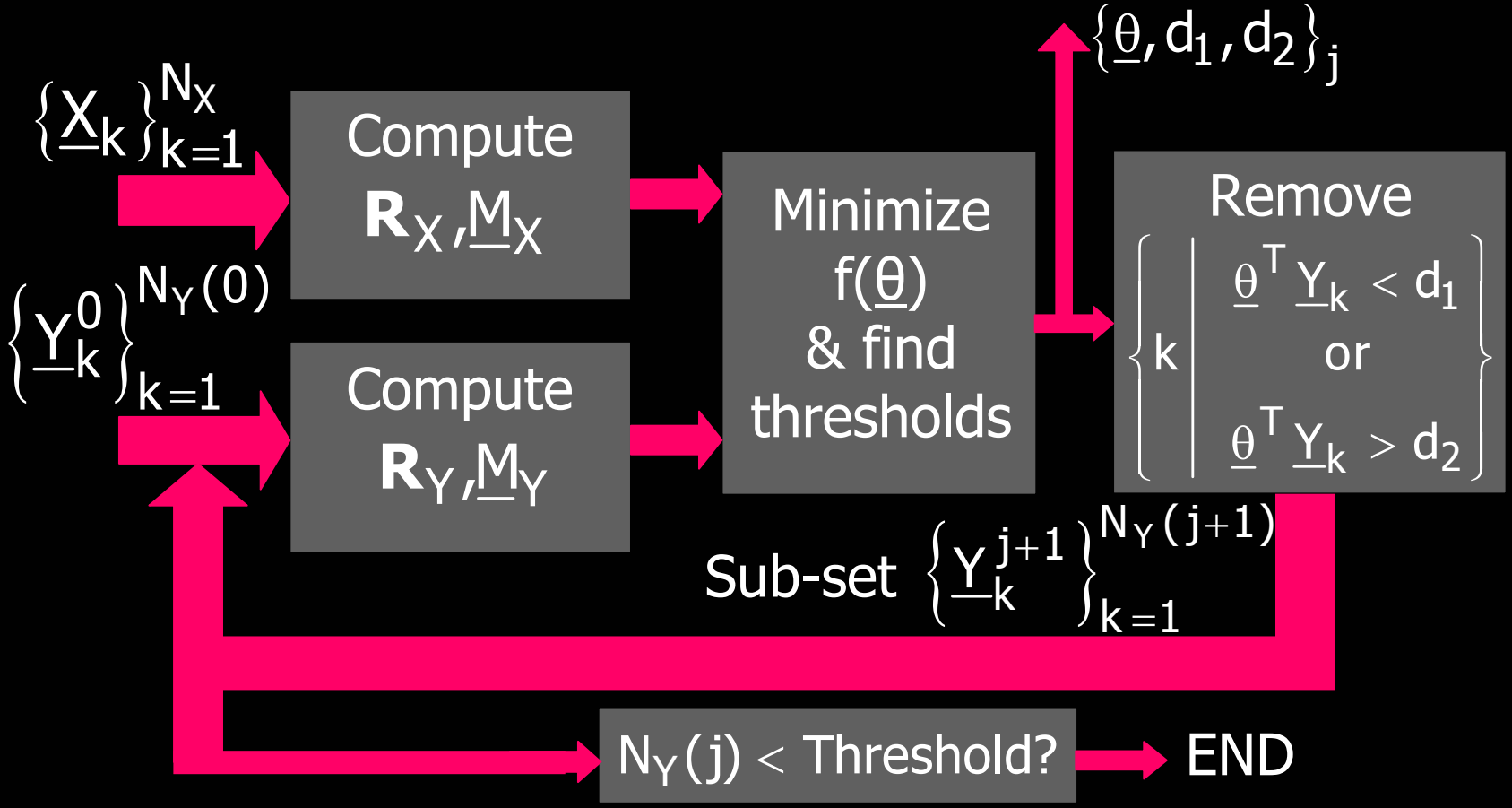
MRC in Practice

5.1 General

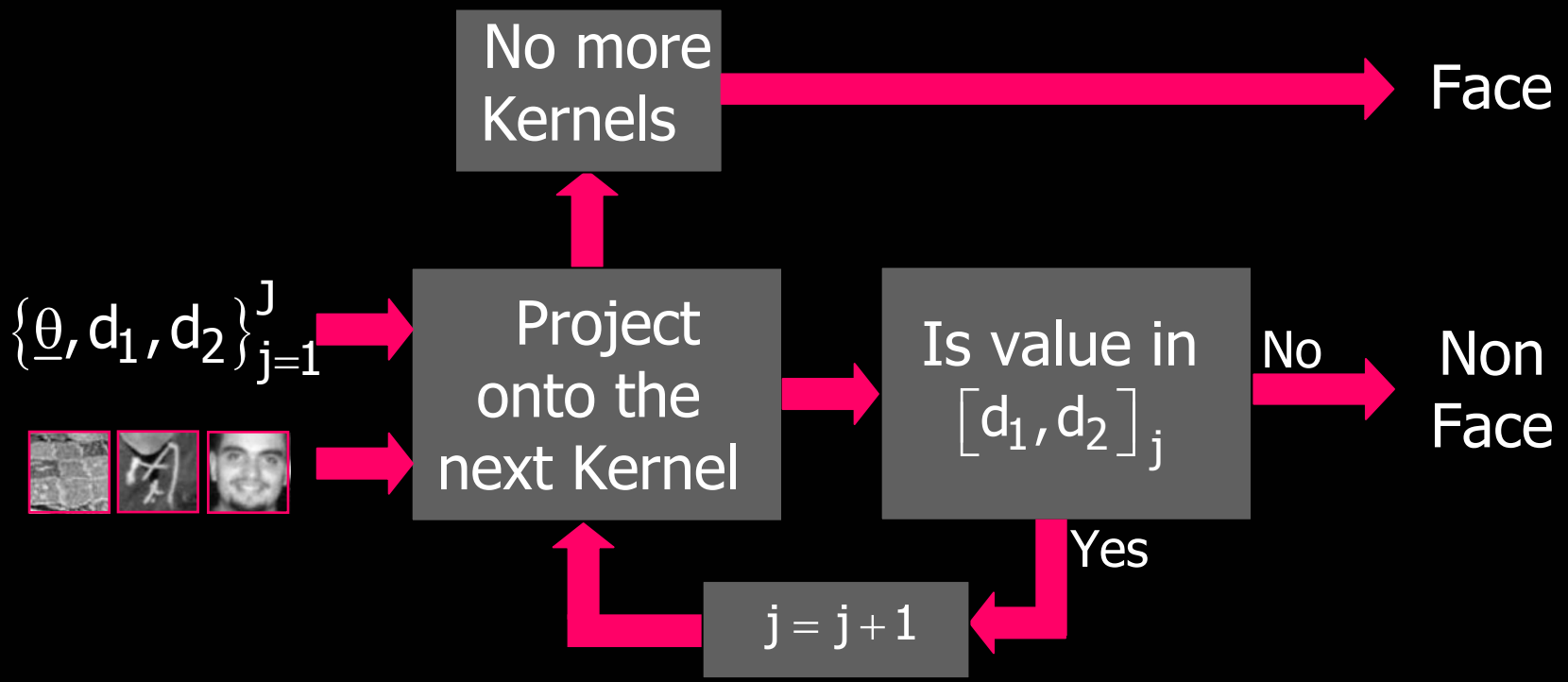
There are two phases to the algorithm:

1. Training: Computing the projection kernels, and their thresholds. This process is done ONCE and off line
2. Testing: Given an image, finding faces in it using the above found kernels and thresholds.

5.2 Training Stage



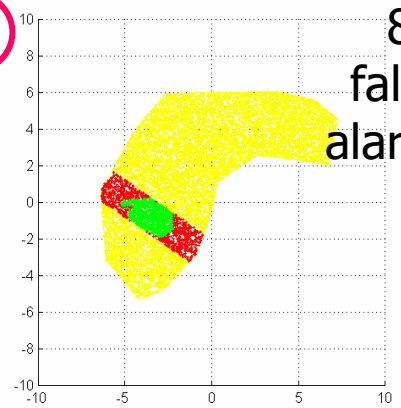
5.3 Testing Stage



5.5 2D Example

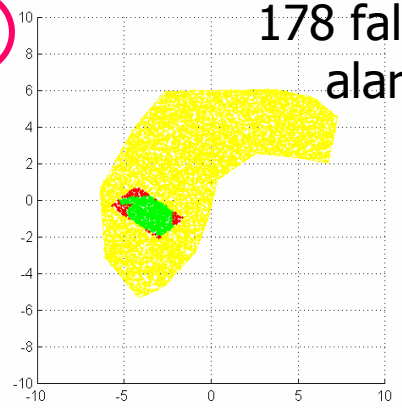
Original yellow set contains 10000 points

1



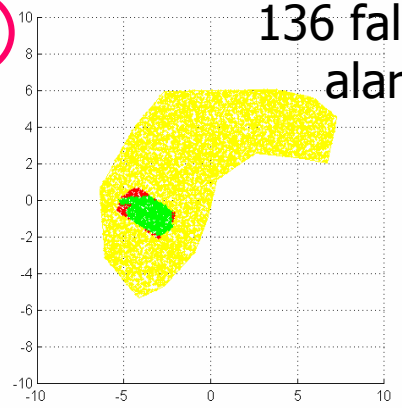
875 false-alarms

2



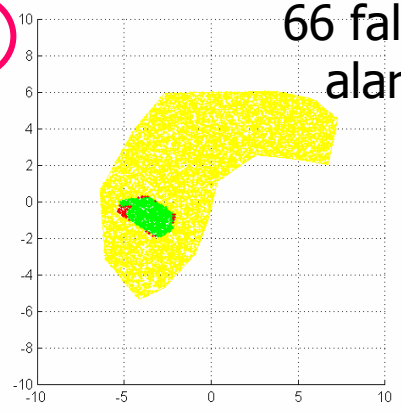
178 false-alarms

3



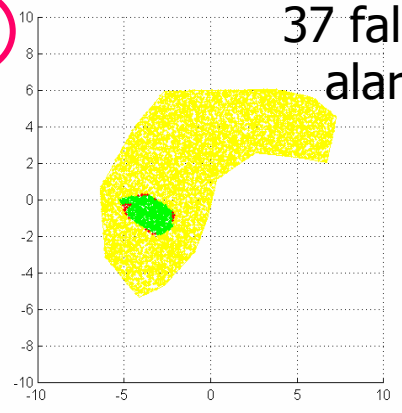
136 false-alarms

4



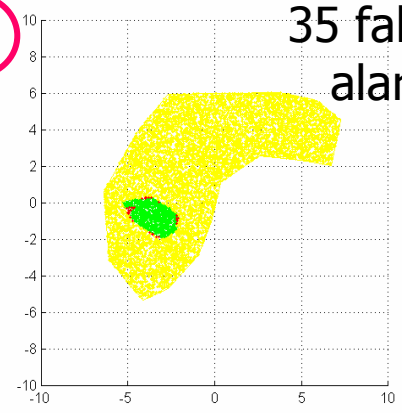
66 false-alarms

5



37 false-alarms

6

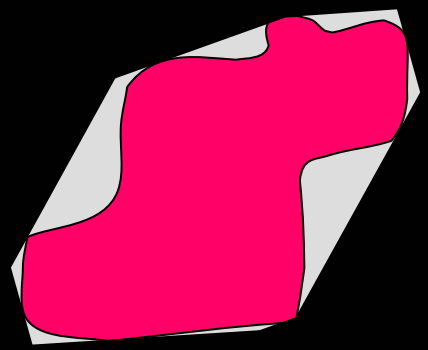


35 false-alarms

5.6 Limitations

- The discriminated zone is a parallelogram. Thus, if the faces set is non-convex, zero false alarm is impossible!!

  Solution: Second Layer



- Even if the faces-set is convex, convergence to false-alarms is not guaranteed.

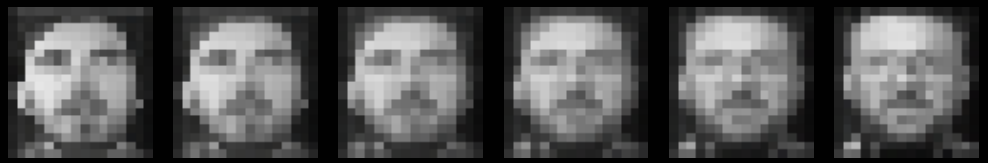
  Solution: Clustering

5.7 Convexity?

Can we assume that the Faces set is convex?

- We are dealing with frontal and vertical faces only

- We are dealing with a low-resolution representation of the faces



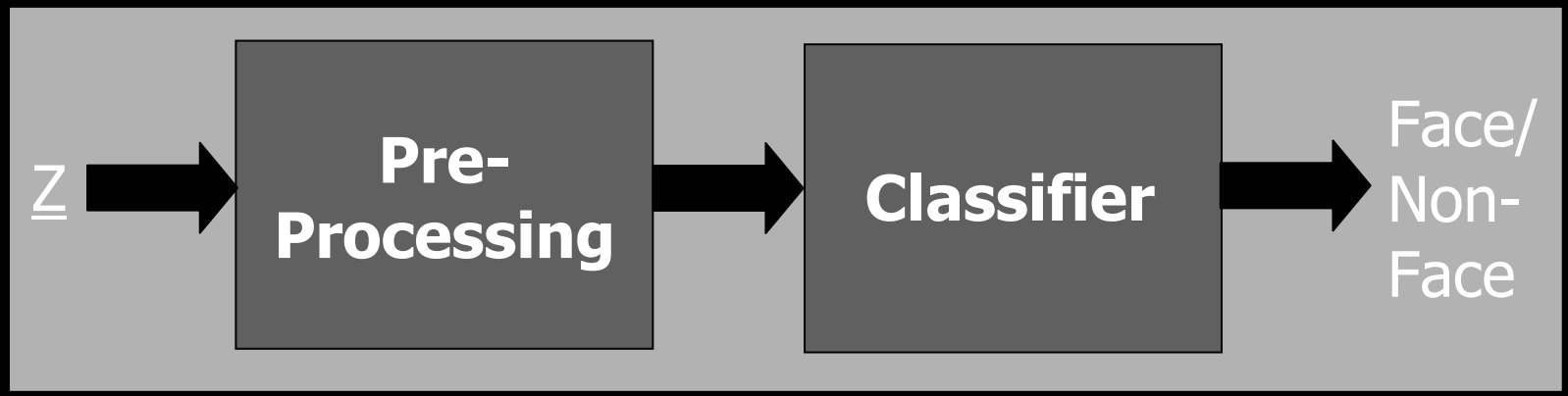
Chapter 6

Pre-Processing

No time ? Press here 

6.1 Pre-Process

Typically, pre-processing is an operation that is done to each block Z before applying the classifier, in order to improve performance



Problem: Additional computations !!

6.2 Why Pre-Process

Example: Assume that we would like our detection algorithm to be robust to the illumination conditions system

Option: Acquire many faces with varying types of illuminations, and use these in order to train the system

Alternative: Remove (somehow) the effect of the varying illumination, prior to the application of the detection algorithm – Pre-process!

6.3 Linear Pre-Process

If the pre-processing is linear ($P \cdot \underline{Z}$):

$$C\left\{P \cdot \underline{Z}\right\} = \left\{ \begin{array}{ll} d_1 \le \underline{\theta}^T P \underline{Z} \le d_2 & \Rightarrow \quad \text{Face} \\ \text{Otherwise} & \Rightarrow \quad \text{Non-Face} \end{array} \right\}$$

$$= \left\{ \begin{array}{ll} d_1 \le \left[P^T \underline{\theta}\right]^T \underline{Z} \le d_2 & \Rightarrow \quad \text{Face} \\ \text{Otherwise} & \Rightarrow \quad \text{Non-Face} \end{array} \right\}$$

The pre-processing is performed on the Kernels ONCE, before the testing !!

6.4 Possibilities

1. Masking to disregard irrelevant pixels.

2. Subtracting the mean to remove sensitivity to brightness changes.

Can do more complex things, such as masking some of the DCT coefficients, which results with reducing sensitivity to lighting conditions, noise, and more ...

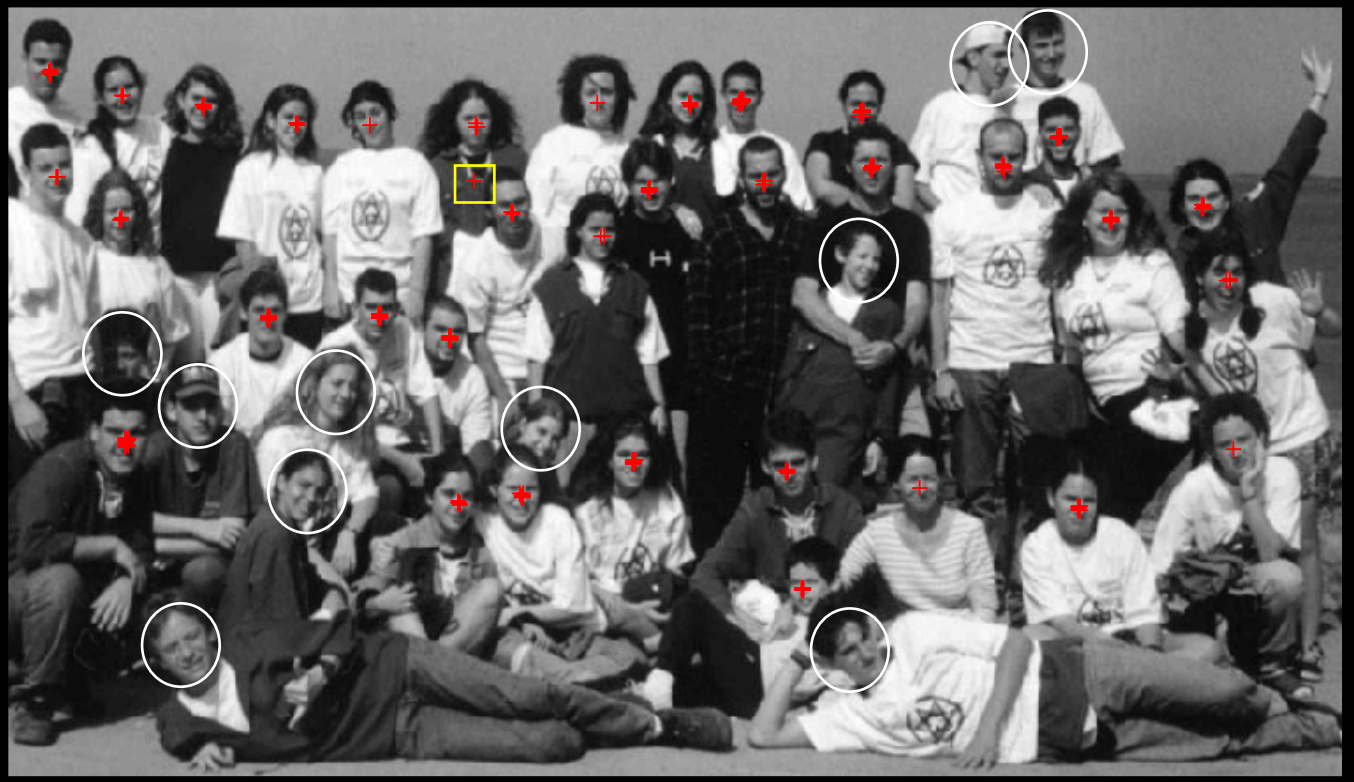
Chapter 7

Results & Conclusions

7.1 Details

- Kernels for finding *faces* (15·15) and *eyes* (7·15).
- Searching for eyes and faces sequentially - very efficient!
- Face DB: 204 images of 40 people (ORL-DB after some screening). Each image is also rotated $\pm 5°$ and vertically flipped - to produce 1224 Face images.
- Non-Face DB: 54 images - All the possible positions in all resolution layers and vertically flipped - about $40 \cdot 10^6$ non-face images.
- Core MRC applied (no fancy improvements).

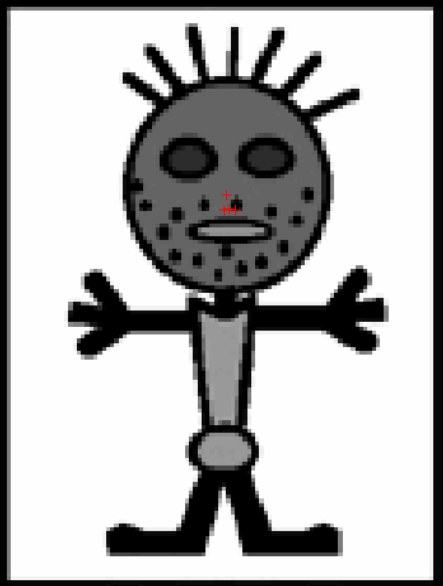
7.2 Results - 1



Out of 44 faces, 10 faces are undetected, and 1 false alarm

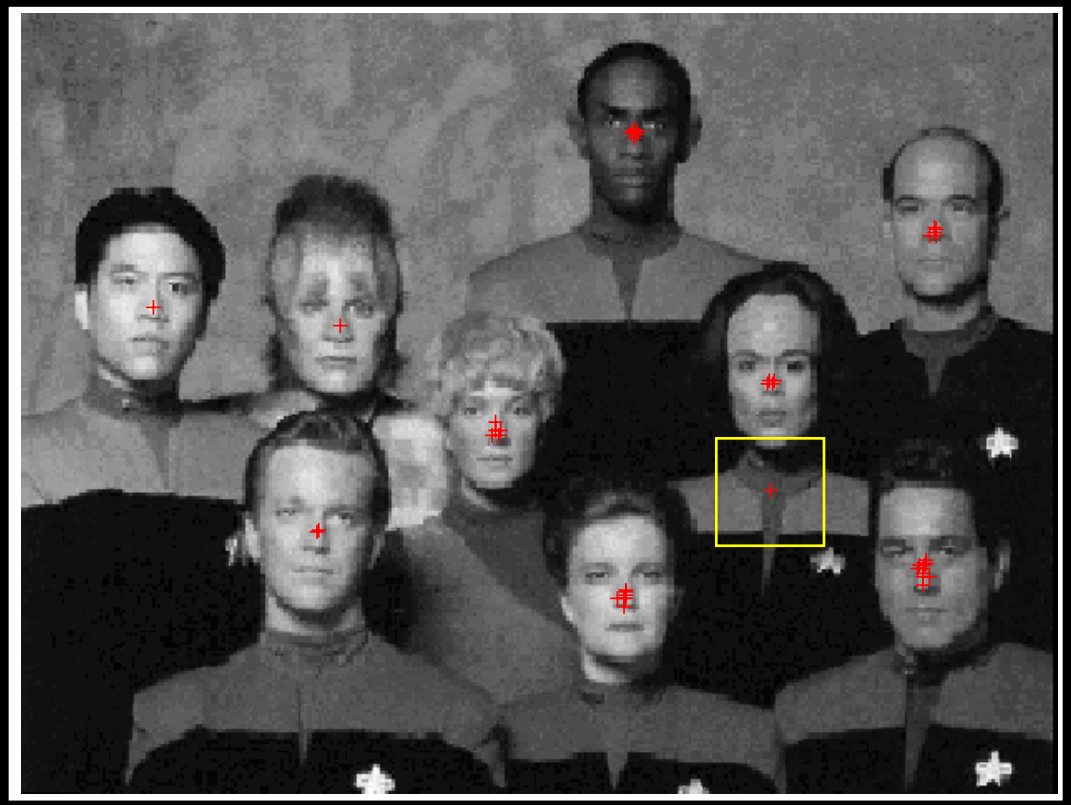
(the undetected faces are circled - they are either rotated or shadowed)

7.3 Results - 2



All faces detected with no false alarms

7.4 Results - 3



All faces detected with 1 false alarm

(looking closer, this false alarm can be considered as face)

7.5 Complexity

- A set of 15 kernels - the first typically removes about 90% of the pixels from further consideration. Other kernels give a rejection of 50%.
- The algorithm requires slightly more that one **convolution** of the image (per each resolution layer).
- Compared to state-of-the-art results:
  - Accuracy – Similar to (slightly inferior in FA) to Rowley and Viola.
  - Speed – Similar to Viola – much faster (factor of ~10) compared to Rowley.

7.6 Conclusions

- MRC: projection onto pre-trained kernels, and thresholding. The process is a rejection based discrimination.
- MRC is simple to apply, with promising results for face-detection in images.
- Further work is required to implement this method and tune its performance.
- If the face set is non-convex (e.g. faces in all angles), MRC can serve as a pre-processing for more complex algorithms.
- More details – http://www-sccm.stanford.edu/~elad

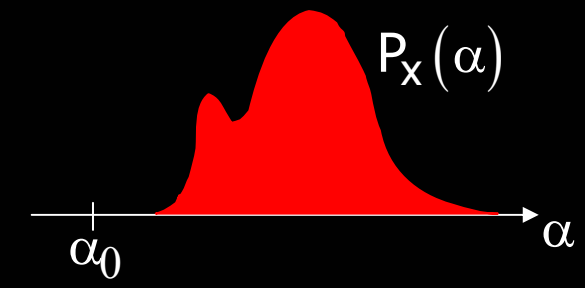
Chapter 8

Appendices

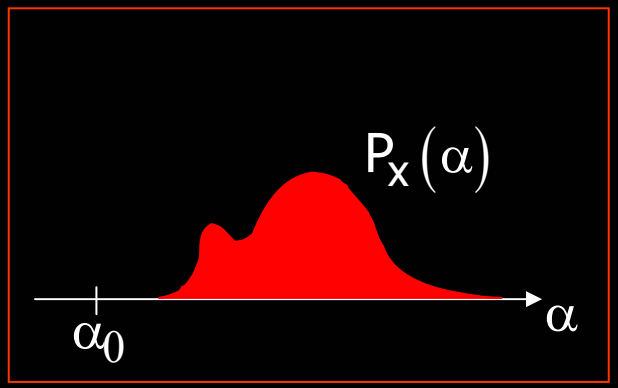
8.1 Scale-Invariant

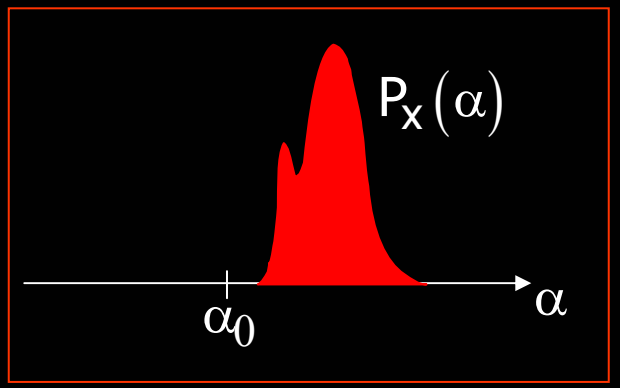
$$D_1\left\{\alpha_0, P_x(\alpha)\right\} = \int_{\alpha} \frac{(\alpha_0 - \alpha)^2}{r_x^2} P_x(\alpha)\, d\alpha$$

$$= \frac{(\alpha_0 - m_x)^2 + r_x^2}{r_x^2}$$



Same distance for





$$f\{\underline{\theta}\} = \frac{\underline{\theta}^T \left\{ [\underline{M}_X - \underline{M}_Y][\underline{M}_X - \underline{M}_Y]^T + R_X + R_Y \right\} \underline{\theta}}{\underline{\theta}^T R_X \underline{\theta}}$$

In this expression:

1. The two classes means are encouraged to get far from each other
2. The Y-class is encouraged to spread as much as possible, and
3. The X-class is encouraged to condense to a near-constant value

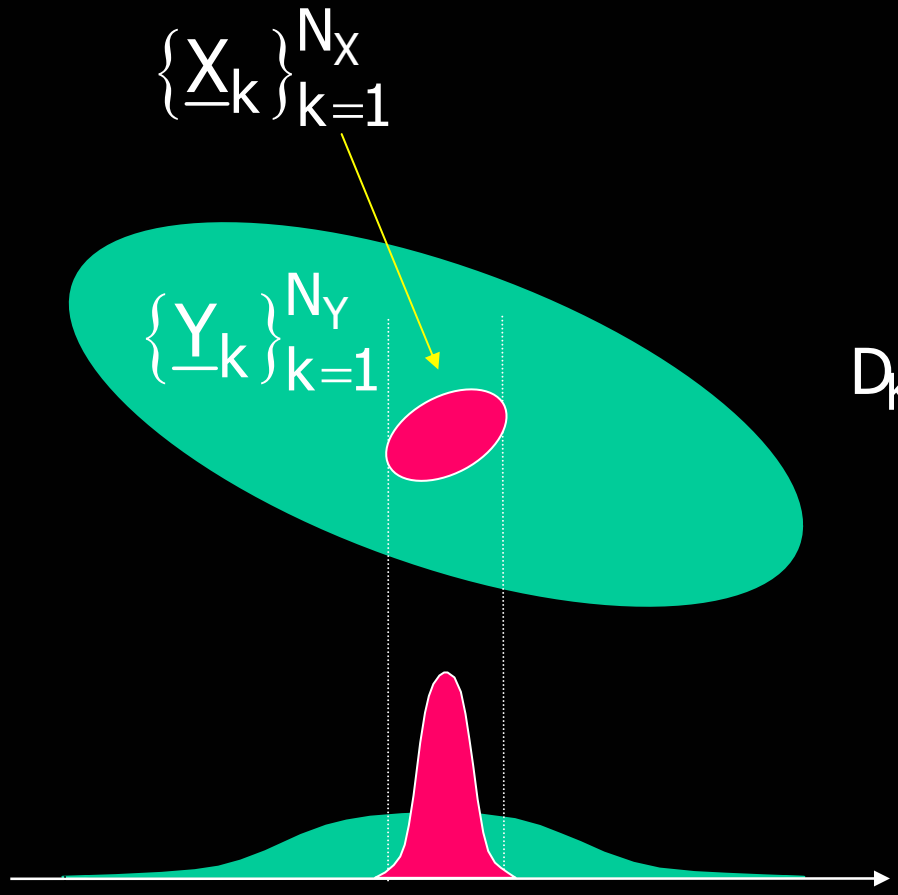
Thus, getting good rejection performance.

8.2 Counting Convolutions

- Assume that the first kernel rejection is $0<\alpha<1$ (I.e. α of the incoming blocks are rejected).
- Assume also that the other stages rejection rate is 0.5.
- Then, the number of overall convolutions per pixel is given by

$$\alpha\cdot 1+(1-\alpha)\sum_{k=2}^{\infty} k\cdot 0.5^{k-1}=3-2\alpha=\begin{cases}\sim 1 & \alpha=0.99\\ 1.2 & \alpha=0.9\\ 1.8 & \alpha=0.6\end{cases}$$

8.3 Different Analysis 1



If the two PDF's are assumed Gaussians, their KL distance is given by

$$D_{KL}\{P_x, P_y\} = \frac{(m_x - m_y)^2 + r_x^2 + r_y^2}{2r_x^2} + \ln\left\{\frac{r_x}{r_y}\right\} - 1$$

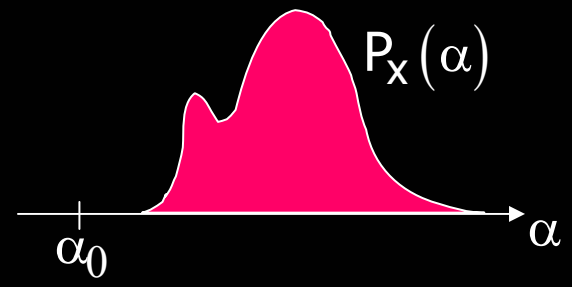
And we get a similar expression

8.4 Different Analysis 2

Define a distance between a point and a PDF by

$$D_1\{\alpha_0, P_x(\alpha)\} = \int_{\alpha} \frac{(\alpha_0 - \alpha)^2}{r_x^2} P_x(\alpha)\, d\alpha$$

$$= \frac{(\alpha_0 - m_x)^2 + r_x^2}{r_x^2}$$



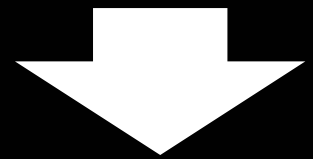
$$D_2\{P_x(\alpha), P_y(\alpha)\} = \int_{\alpha} D_1\{\alpha, Px(\alpha)\} P_y(\alpha) d\alpha = \frac{(m_x - m_y)^2 + r_x^2 + r_y^2}{r_x^2}$$

This distance is asymmetric !! It describes the average distance between points of Y to the X-PDF, $P_X(\alpha)$.

$$D_3\left\{P_x(\alpha), P_y(\alpha)\right\} = P(Y)\frac{(m_x - m_y)^2 + r_x^2 + r_y^2}{r_x^2} + P(X)\frac{(m_x - m_y)^2 + r_x^2 + r_y^2}{r_y^2}$$

In the case of face detection in images we have

P(X)<<P(Y)



We should Maximize $\dfrac{(m_x - m_y)^2 + r_x^2 + r_y^2}{r_x^2}$

8.5 Relation to Fisher

In the general case we maximize the distance

$$P(Y)\frac{(m_x - m_y)^2 + r_x^2 + r_y^2}{r_x^2} + P(X)\frac{(m_x - m_y)^2 + r_x^2 + r_y^2}{r_y^2}$$

If P(X)=P(Y)=0.5 we maximize

$$\frac{(m_x - m_y)^2 + r_x^2 + r_y^2}{r_x^2} + \frac{(m_x - m_y)^2 + r_x^2 + r_y^2}{r_y^2}$$

The distance of the Y points to the X-distribution

The distance of the X points to the Y-distribution

Instead of maximizing the sum

$$\frac{(m_x - m_y)^2 + r_x^2 + r_y^2}{r_x^2} + \frac{(m_x - m_y)^2 + r_x^2 + r_y^2}{r_y^2}$$

Minimize the inverse of the two expressions
(the inverse represent the proximity)

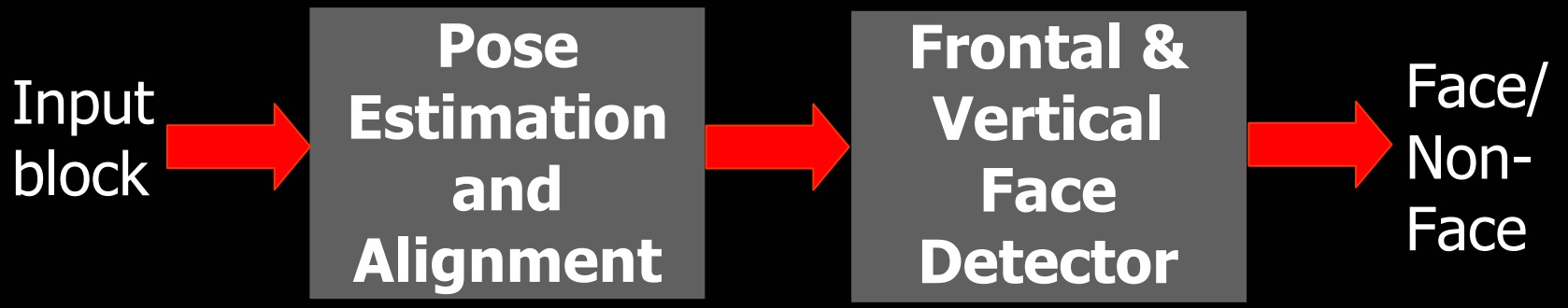
$$\text{Min} \ \frac{r_x^2}{(m_x - m_y)^2 + r_x^2 + r_y^2} + \frac{r_y^2}{(m_x - m_y)^2 + r_x^2 + r_y^2} = \text{Min} \ \frac{r_x^2 + r_y^2}{(m_x - m_y)^2}$$

8.6 Using Color

Several options:

- Trivial approach – use the same algorithm with blocks of L-by-L by 3.
- Exploit color redundancy – work in HSV space with decimated versions of the Hue and the Saturation layers.
- Rejection approach – Design a (possibly non-spatial) color-based simple classifier and use it as the first stage rejection.

8.7 2D-Rotated Faces



Remarks:

1. A set of rotated kernels can be used instead of actually rotating the input block
2. Estimating the pose can be done with a relatively simple system (few convolutions).

8.8 3D-Rotated Faces

A possible solution:

1. Cluster the face-set to same-view angle faces and design a Final classifier for each group using the rejection approach
2. Apply a pre-classifier for fast rejection at the beginning of the process.
3. Apply a mid-classifier to map to the appropriate cluster with the suitable angle

8.9 Faces vs. Targets

- Treating other targets can be done using the same concepts of
  - Treatment of scale and location
  - Building and training sets
  - Designing a rejection based approach (e.g. MRC)
  - Boosting the resulting classifier
- The specific characteristics of the target in mind could be exploited to fine-tune and improve the above general tools.

8.10 Further Improvements

- Pre-processing – linear kind does not cost
- Regularization – compensating for shortage in examples
- Boosted training – enrich the non-face group by finding false-alarms and train on those again
- Boosted classifier – Use the sequence of weak-classifier outputs and apply yet another classifier on them –use ada-boosting or simple additional linear classifier
- Constrained linear classifiers for simpler classifier
- Can apply kernel methods to extend to non-linear version

8.11 Possible Extensions

- Relation to Boosting and Ada-Boosting algorithms,
- Bounding the rejection rate,
- Treating non-convex inner classes:
  - Non-dyadic decision tree,
  - Sub-clustering of the inner-class,
  - Kernel functions,
  - Combine boosting and MRC.
- Replacing the target-functional for better rejection performance – SVM-like approach.