Prof. Marios Savvides

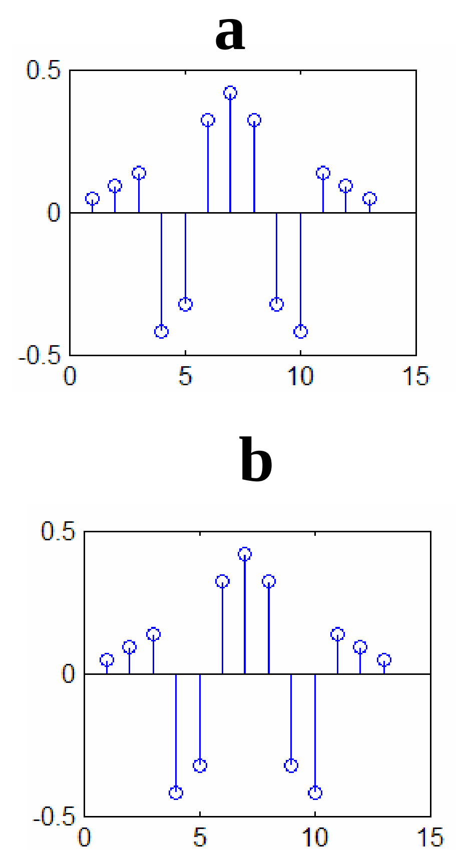
# Pattern Recognition Theory

**Lecture 12 : Correlation Filters**

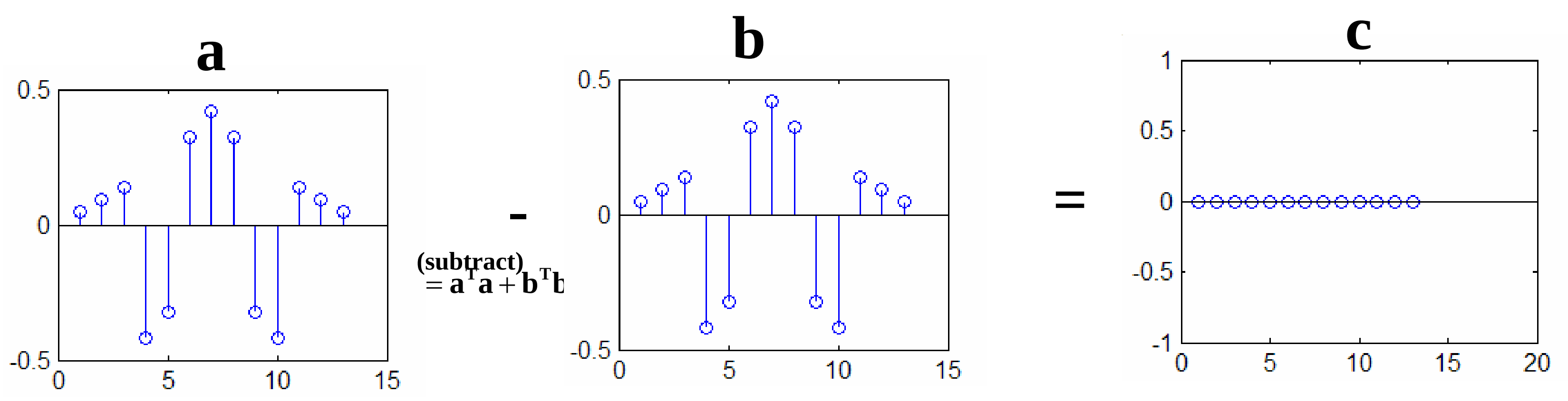
Electrical & Computer ENGINEERING

# Pattern Matching

- How to match two patterns?
- How do you locate where the pattern is in a long sequence of patterns?
- Are these two patterns the same?
- How to compute a match metric?



# Pattern Matching




**Lets define mean squared error, i.e.**

$$e = \|\mathbf{a} - \mathbf{b}\|^2 = (\mathbf{a} - \mathbf{b})^{\mathrm{T}}(\mathbf{a} - \mathbf{b}) = \mathbf{a}^{\mathrm{T}}\mathbf{a} + \mathbf{b}^{\mathrm{T}}\mathbf{b} - 2\mathbf{a}^{\mathrm{T}}\mathbf{b}$$

$$\mathbf{a}^{\mathrm{T}}\mathbf{a} = \sum_{i=1}^{N} a(i)^2 = energy\_of\_a$$

$$\mathbf{b}^{\mathrm{T}}\mathbf{b} = \sum_{i=1}^{N} \mathbf{b}(i)^2 = energy\_of\_\mathbf{b}$$

$$\mathbf{a}^{\mathrm{T}}\mathbf{b} = \sum_{i=1}^{N} \mathbf{a}(i)\mathbf{b}(i) = correlation\_term$$

# Pattern Matching

$$e = \|\mathbf{a} - \mathbf{b}\|^2 = (\mathbf{a} - \mathbf{b})^{\mathrm{T}}(\mathbf{a} - \mathbf{b}) = \mathbf{a}^{\mathrm{T}}\mathbf{a} + \mathbf{b}^{\mathrm{T}}\mathbf{b} - 2\mathbf{a}^{\mathrm{T}}\mathbf{b}$$

$$\mathbf{a}^{\mathrm{T}}\mathbf{a} = \sum_{i=1}^{N} a(i)^2 = energy\_of\_a$$

$$\mathbf{a}^{\mathrm{T}}\mathbf{b} = \sum_{i=1}^{N} \mathbf{a}(i)\mathbf{b}(i) = correlation\_term$$

$$\mathbf{b}^{\mathrm{T}}\mathbf{b} = \sum_{i=1}^{N} \mathbf{b}(i)^2 = energy\_of\_\mathbf{b}$$

- Assume we normalize energy of a and b to 1 i.e. $a^{T}a=b^{T}b=1$
- Then to minimize error, we seek to maximize the correlation term.
- So performing correlation, the maximum correlation point is the location where the two pattern match best.

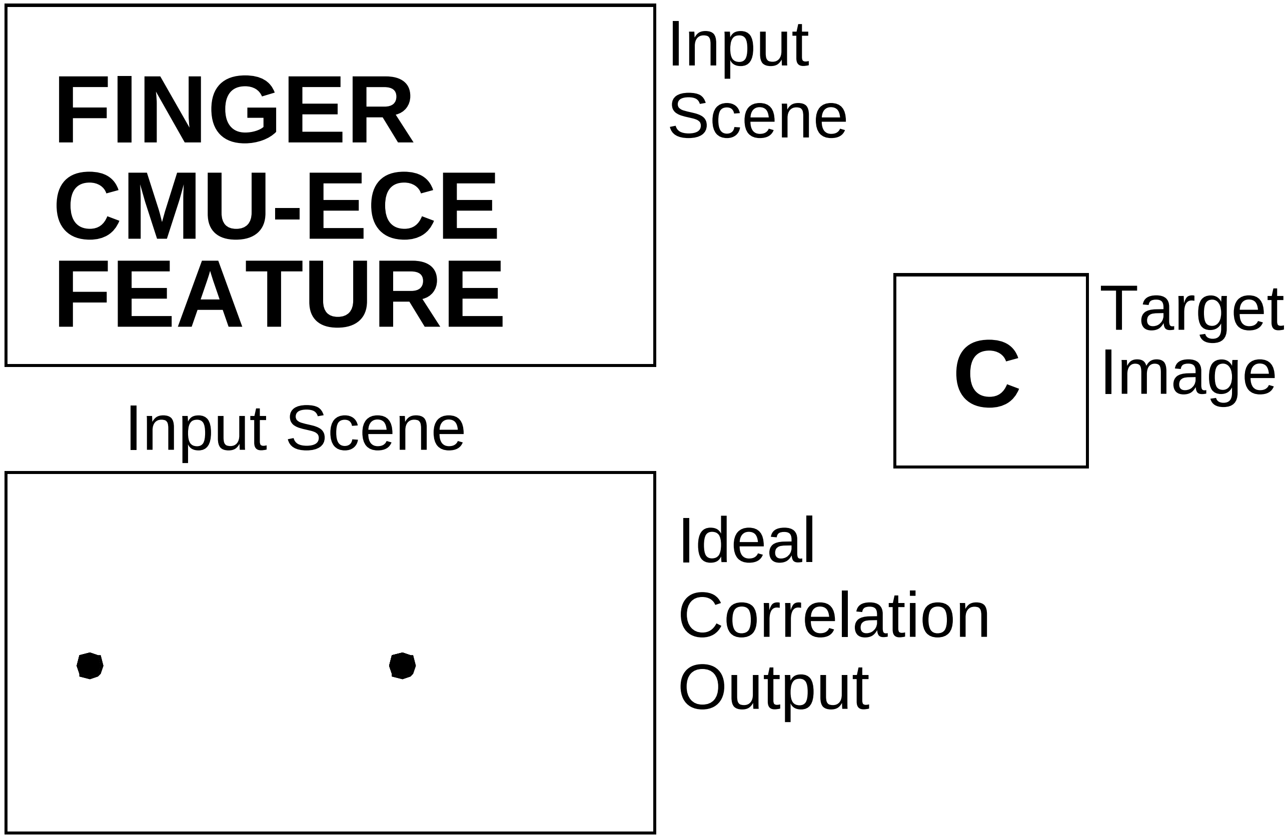
# Correlation Pattern Recognition

- r(x) test pattern
- s(x) reference pattern

$$-1 \leq \frac{\mathbf{a}^{T}\mathbf{b}}{\sqrt{(\mathbf{a}^{T}\mathbf{a})(\mathbf{b}^{T}\mathbf{b})}} \leq 1$$

- Normalized correlation between a(x) and b(x) gives 1 if they match perfectly (i.e. only if a(x) = b(x) ) and close to 0 if they don't match.
- Problem: Reference patterns rarely have same appearance
- Solution: Find the pattern that is consistent (i.e., yields large correlation) among the observed variations.

# Object Recognition using correlation




Goal: Locate all occurrences of a target in the input scene

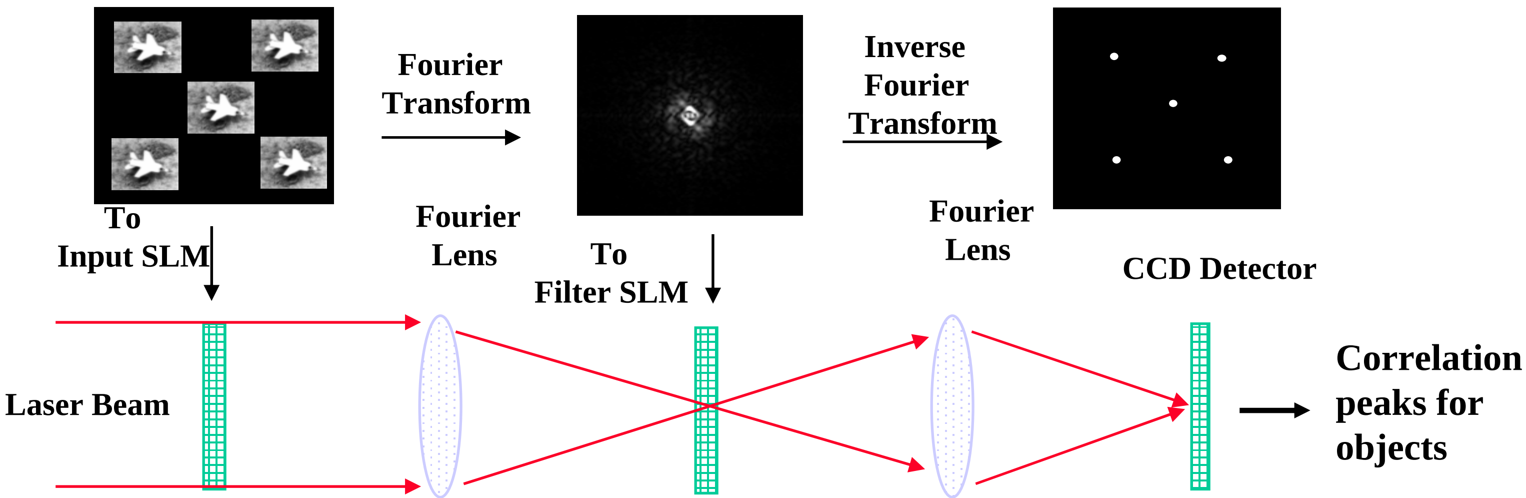


Marios Savvides

# Why correlation filters?

- Built-in Shift-Invariance: shift in the input image leads to a corresponding shift in the output peak. Classification result remains unchanged.
- Matched filters are just replicas of the object that we are trying to find. Problem is we need as many matched filters as the different appearances that object can look under different conditions. (i.e. a matched filter for every pose, illumination and expression variation).
- Using Matched filters is computationally and memory very expensive.
- We can synthesize distortion tolerant filters that can recognize more than one view of an object.
- We can build different types of distortion-tolerance in each filter (e.g. scale, rotation, illumination etc).
- We will show advanced correlation filters exhibit graceful degradation with input image distortions.

# Optical Correlation @ light speed




SLM: Spatial Light Modulator
CCD: Charge-Coupled Detector

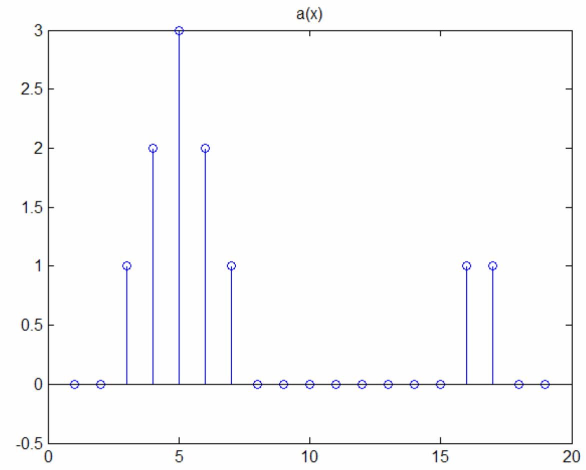
# How to do Correlations Digitally and Efficiently?

- Use Fast Fourier Transforms…
- How? Fourier Transform property tells us:
  - Convolution Theorem:
    - Convolution of two functions a(x) and b(x) in the spatial domain is equivalently computed in the Fourier domain by multiplying the FT{a(x)} with the FT{b(x)}.
    - i.e. in matlab this would be
      - Convolution=IFFT ( FFT(a) .* FFT(b) ) (assuming 1 D)
  - Correlation Theorem:
    - Similar to convolution except the correlation of functions a(x) and b(x) in the spatial domain is equivalently computed in the Fourier domain by multiplying FT{a(x)} with conjugate of FT{b(x)}.
      - Correlation=IFFT ( FFT(a) .* conj(FFT(b)) )

# Some Digital Signal Processing basics

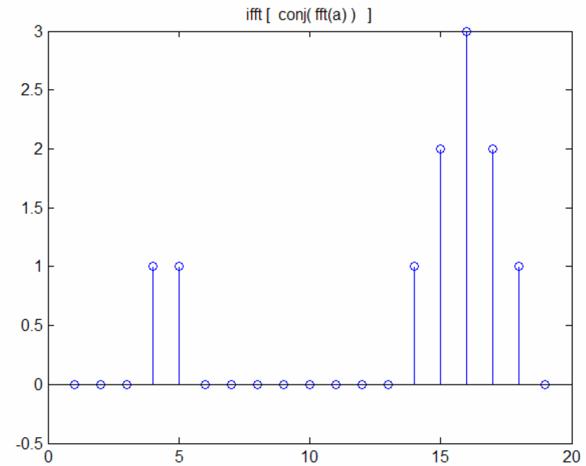
- Take a signal a(x)
- a=[ 0 0 1 2 3 2 1 0 0 0 0 0 0 0 0 1 1 0 0];

Looks like this ->




[1]Compute its Discrete Fourier Transform or FFT, a(x)->A(u).

- [2] Take the conjugate: conj(A(u))
- [3] Take inverse Fourier Transform of conj( A(u) )



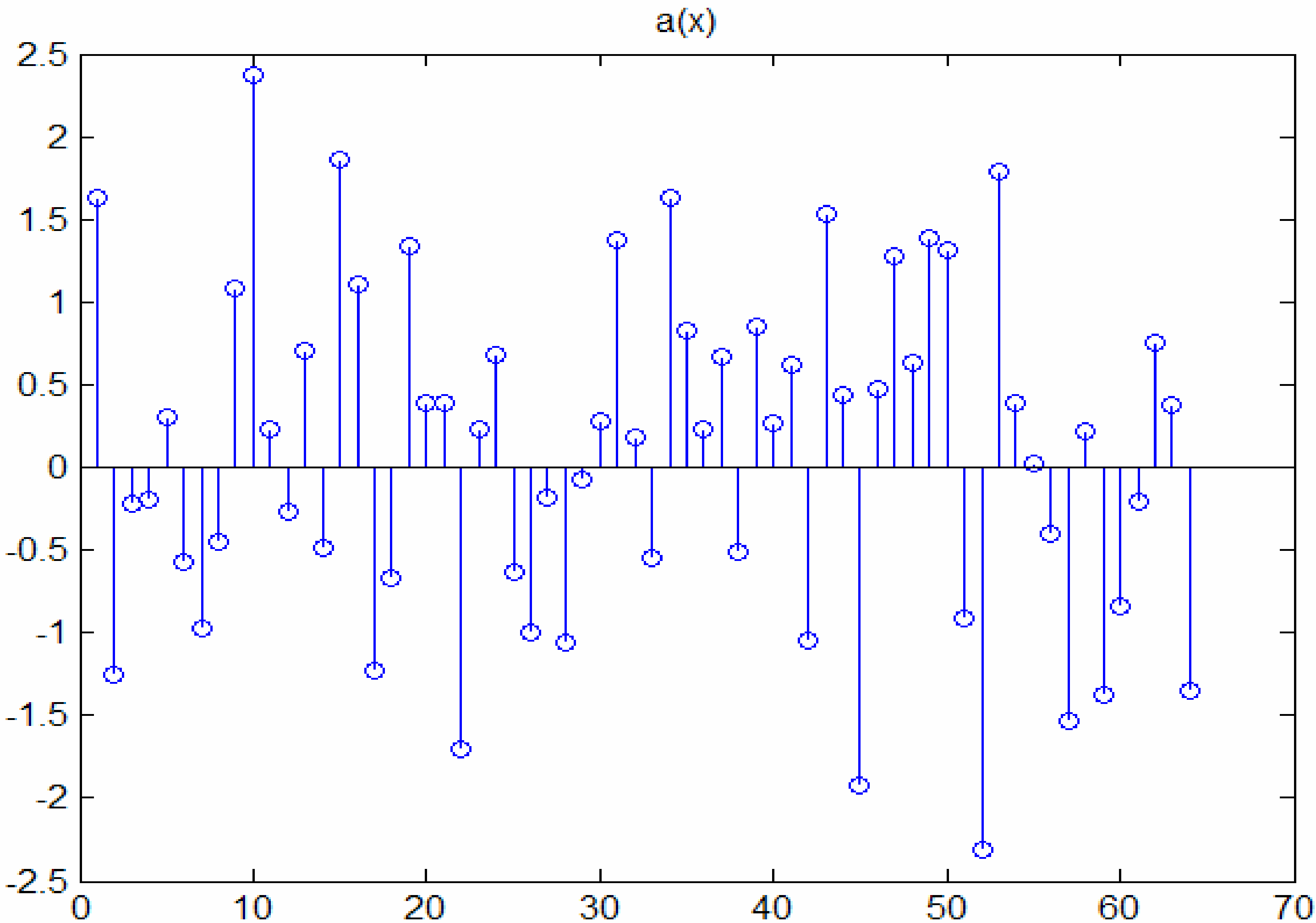

**Conjugation in Frequency domain leads to time-reversal in the time domain**

# So Correlation in Fourier domain is…..

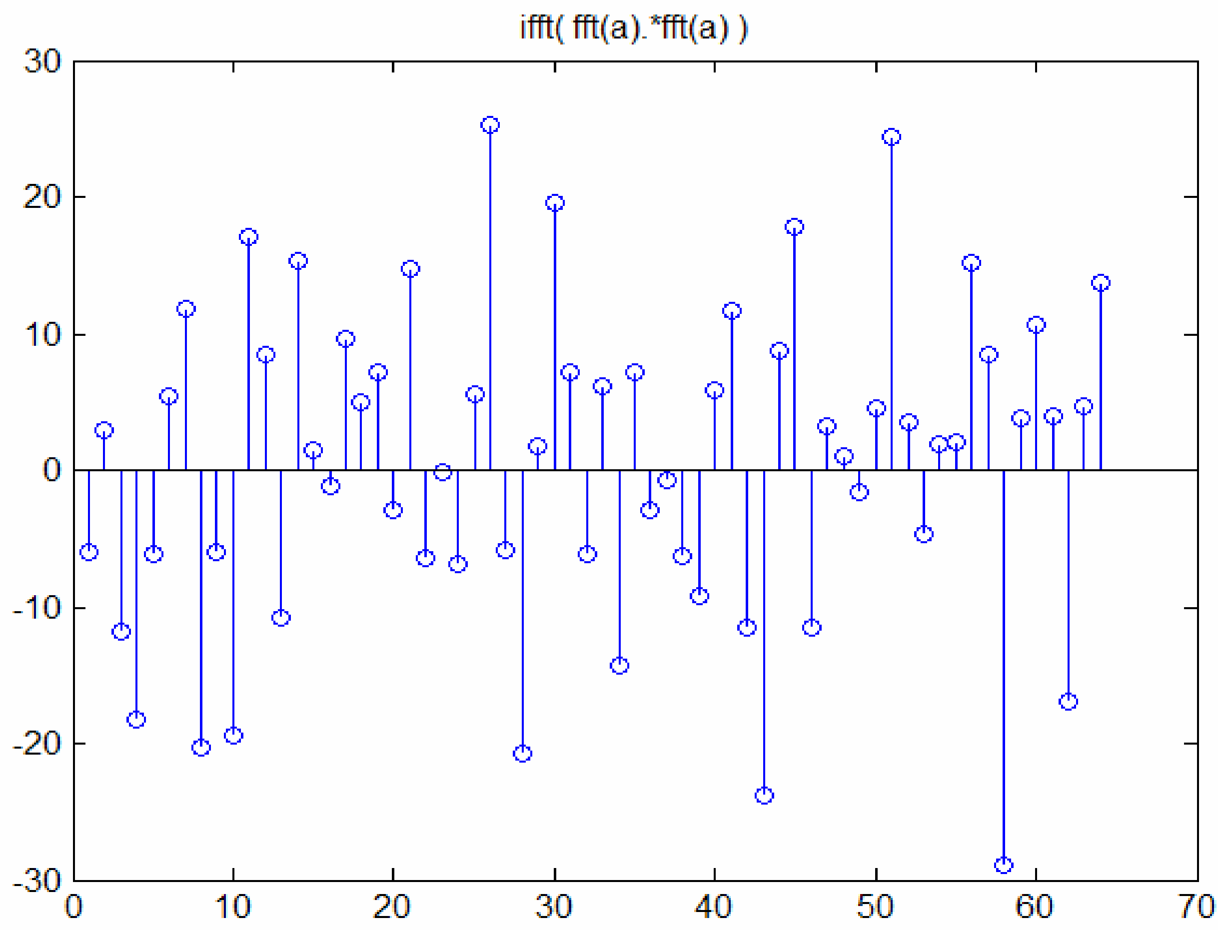
- Is just like convolution except we convolve the test signal t(x) with a time-reversed signal h(x).
- Taking the conjugate in the Fourier domain time-reverses the signal in the time domain.

- Since convolution automatically time-reverses h(x) also..(there is a double time reversal, which cancels out, meaning that you end up computing inner-products with the reference signal and the test signal in the Fourier domain.

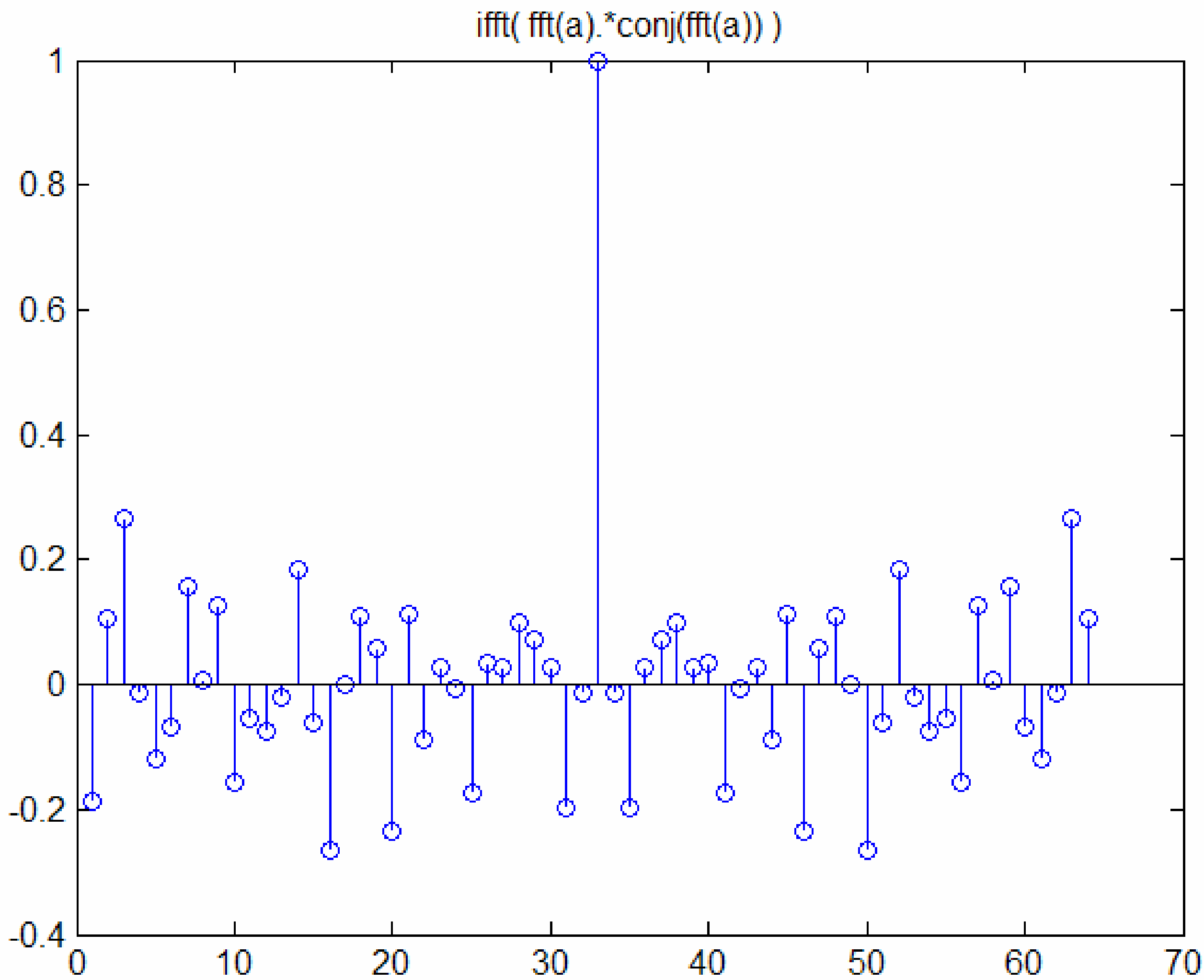
# Lets start with a random sample signal a(x)



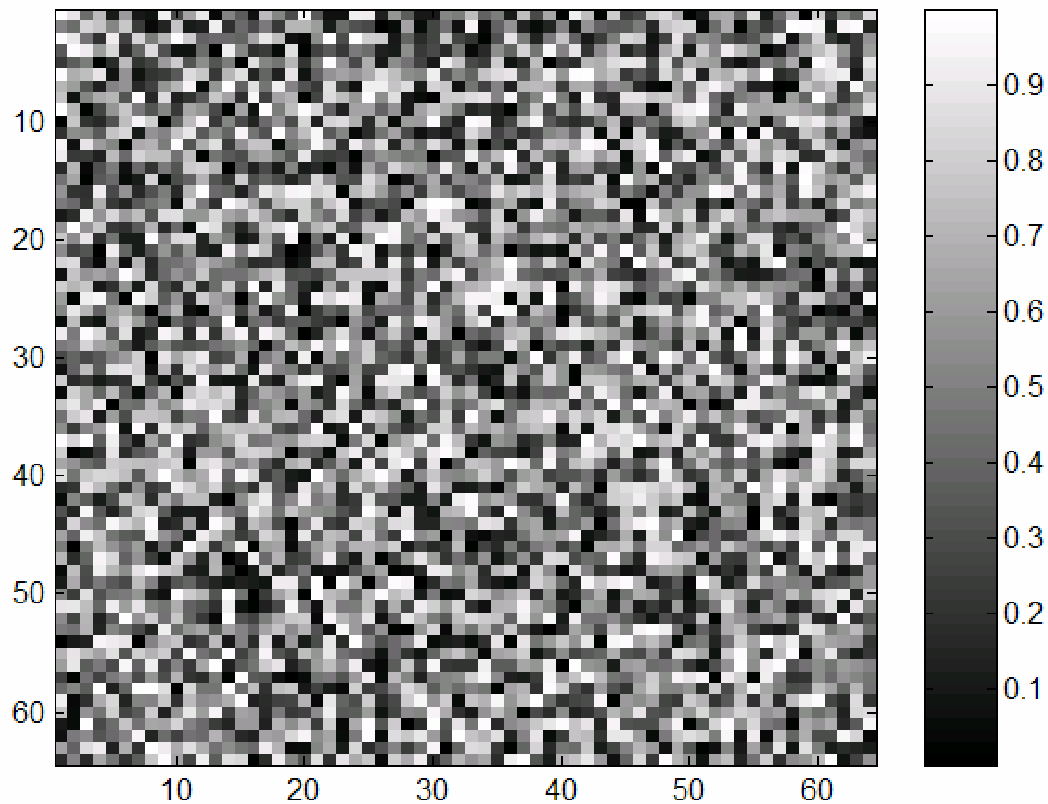
# Here is the result of convolving a(x) with a(x)



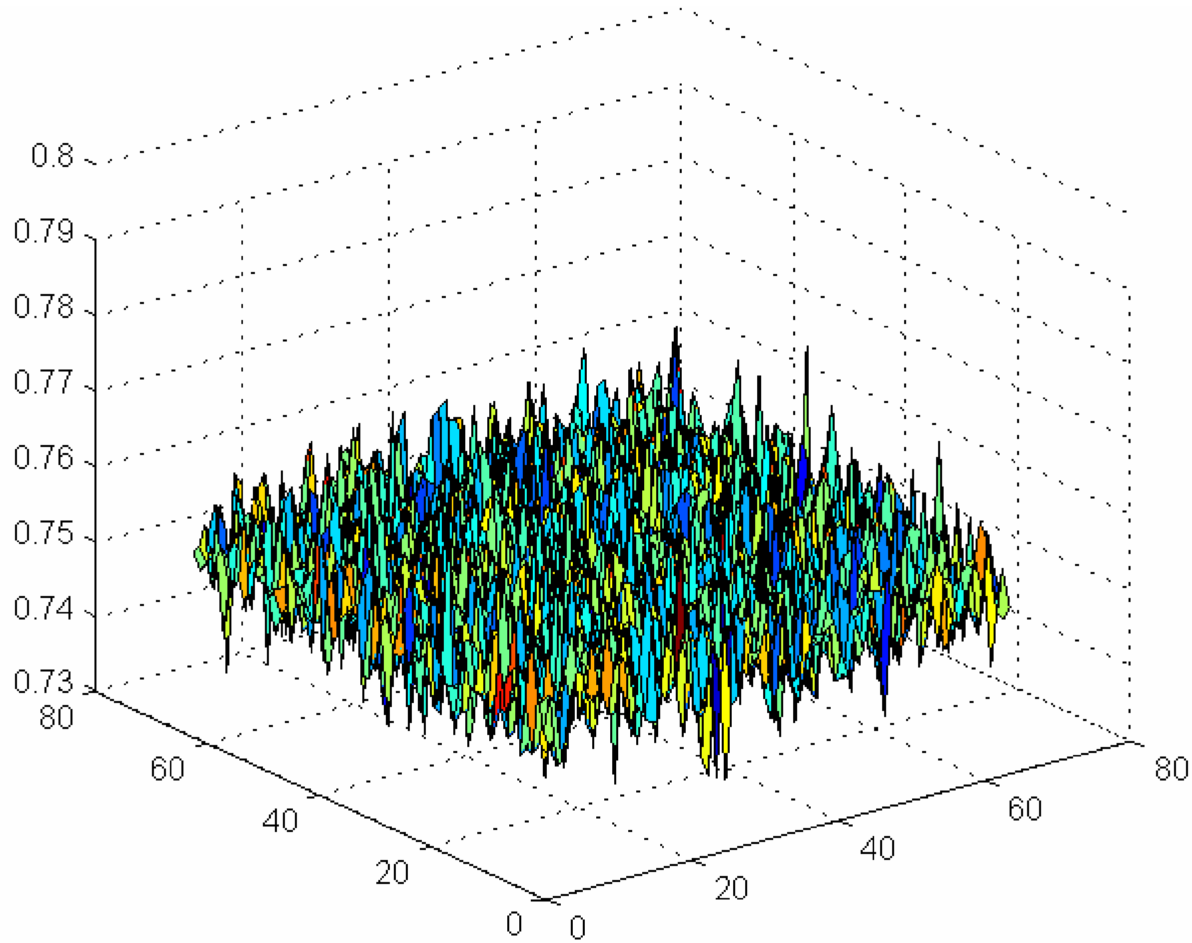
# Here is the result of correlating a(x) with a(x)



# 2D random image s(x,y)

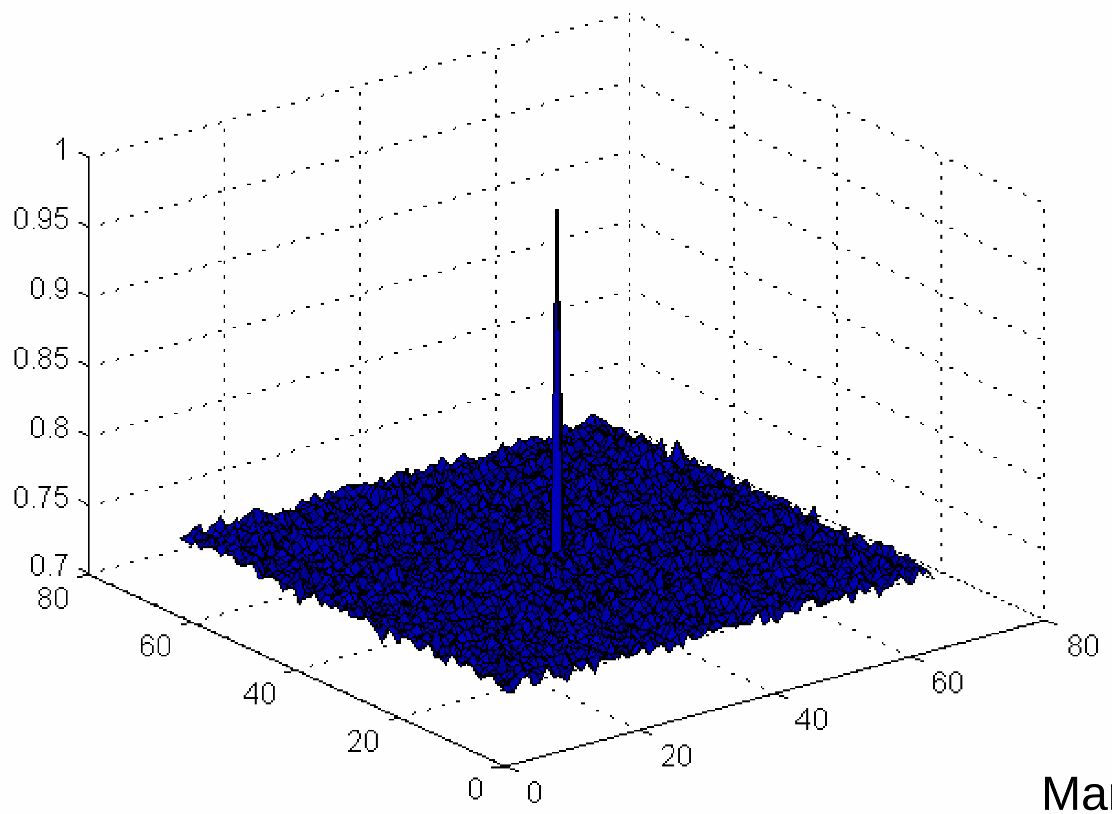


# Convolution of s(x,y) with s(x,y)



# Correlation of s(x,y) with s(x,y) (auto-correlation)

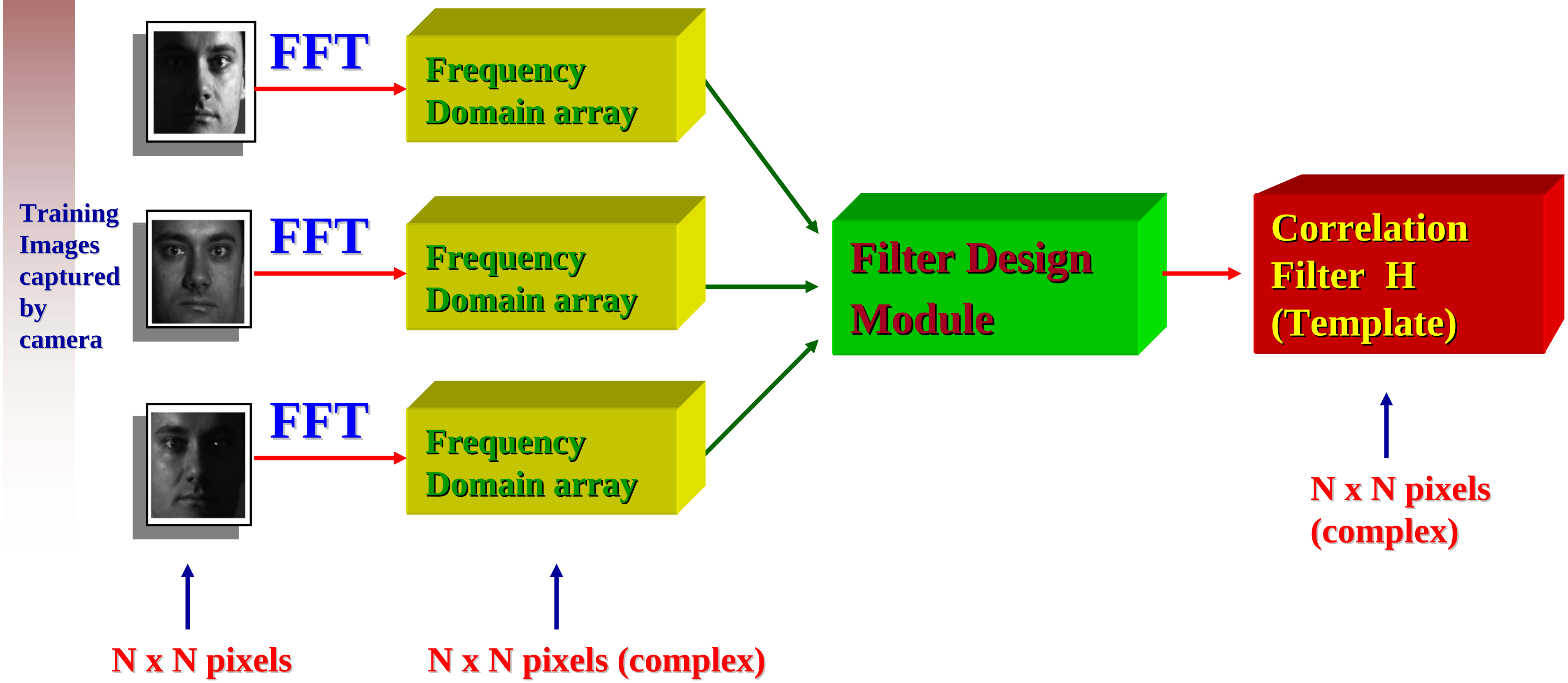
- Notice nice peak..with height of 1 at the location where the images perfectly align. The peak height indicates confidence of match. Because s(x,y) is random signal, no other shifts of the signal match and there is only a single peak appearing exactly where the two signals are aligned.



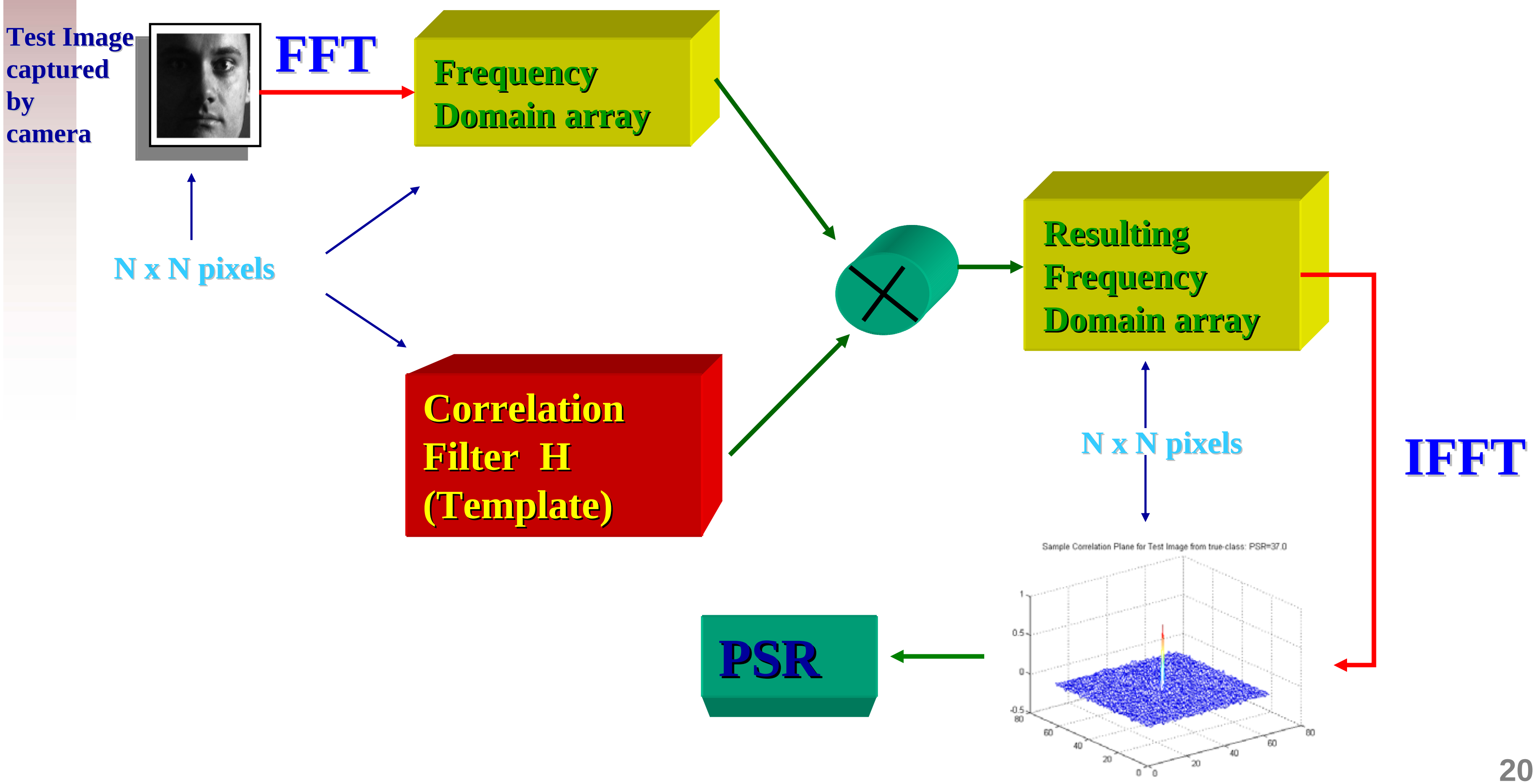
# Matched Filter

- Matched Filter is simply the reference image that you want to match in the scene.
- Matched Filter is optimal for detecting the known reference signal in additive white Gaussian noise (AWGN) – it has maximum Signal-to-Noise Ratio.
- OK...but what are the short-comings of Matched Filters?
  - Matched Filters only match with the reference signal, and even slight distortions will not produce a match.
  - Thus you need as many matched filters as number of training images (N).
  - Not feasible from a computational viewpoint as you will need to perform N correlations and store all N filters.

# Typical Enrollment for Correlation Filters



# Recognition stage



# Equal Correlation Peak Synthetic Discriminant Function (ECP-SDF) Filter

- Idea is to overcome the limitations of MFs, by building a single correlation filter from multiple training images that will be able to recognize all of them.
- How?
- So we want to keep the ideas from MF, in that we want the peak at the origin (i.e. when the two signals align) to be 1 for all the training reference images.
- And we want the filter to be in the convex hull of the training images, i.e. it is made up of a linear combinations of MF (we need to determine the weights).

# ECP-SDF Filter - details

- Assume h is a vector containing the impulse response of the filter (vectorized from 2D).
- Let $x_i$ be vectorized ith training image into a column vector.
- We want the peak at the origin to be 1, i.e. we want the inner-product (since correlations are just inner-products of the filter h and signal x at different spatial shifts).

$$\mathbf{h}^T\mathbf{x}_i = 1$$

So for all i=1..N images we write this as:

$$\mathbf{X}^T\mathbf{h}=\mathbf{c}$$

- Where matrix $X=[\,x_1\, x_2\, x_3\, x_4\,]$ contains the images $x_i$ in vectorized format along its columns.
- c is a column vector containing the peak contraints- in this case a vector of all 1's

# ECP-SDF Filter continued….

- We also said that the ECP-SDF filter is a linear combination of Matched Filters (or equivalently) a linear weighted combination of the training images.
- i.e. $\mathrm{h} = \mathrm{X\,a}$

Where

$$\begin{bmatrix} | \\ \mathbf{h} \\ | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \mathbf{x}_1 & \mathbf{x}_2 & & \mathbf{x}_N \\ | & | & & | \end{bmatrix} \begin{bmatrix} | \\ \mathbf{a} \\ | \end{bmatrix} = a_1\mathbf{x}_1 + a_2\mathbf{x}_2 + a_3\mathbf{x}_3 + \ldots + a_N\mathbf{x}_N$$

Subst $\mathrm{h} = \mathrm{X\,a}$ in $\mathrm{X}^T\mathrm{h} = \mathrm{c}$ to get

$$\mathrm{X}^T\mathrm{X}\mathrm{a} = \mathrm{c}$$

# ECP-SDF Filter continued….

- Use $X^TXa=c$ to solve for the linear combination weights a

$$a = ( X^TX )^{-1} c$$

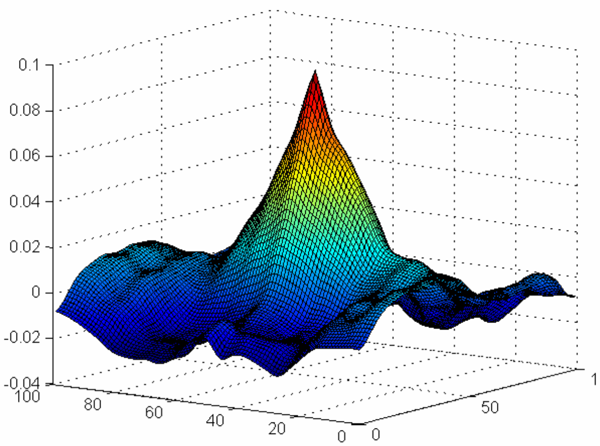
- $X^TX$ is an inner-product matrix or Gram matrix of size NxN where N is the number of training images, thus computationally efficient.
- OK.. Now we know the linear combination weights so lets plug $a$ back in our filter equation $h= X a$ to get

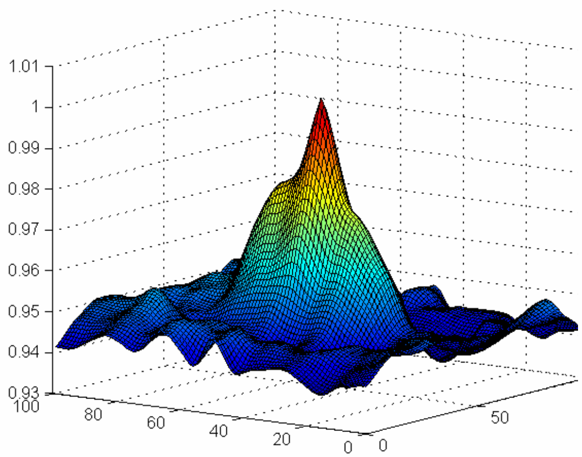
$$h= X ( X^TX )^{-1} c$$

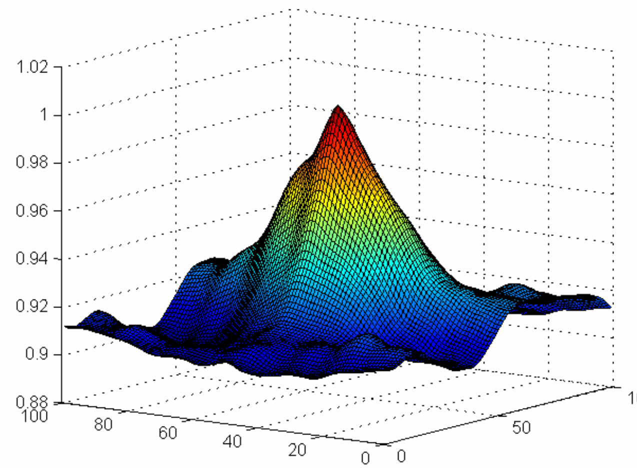
# Example Correlation Outputs

- Use these images for training (peak=1 for all correlation outputs)

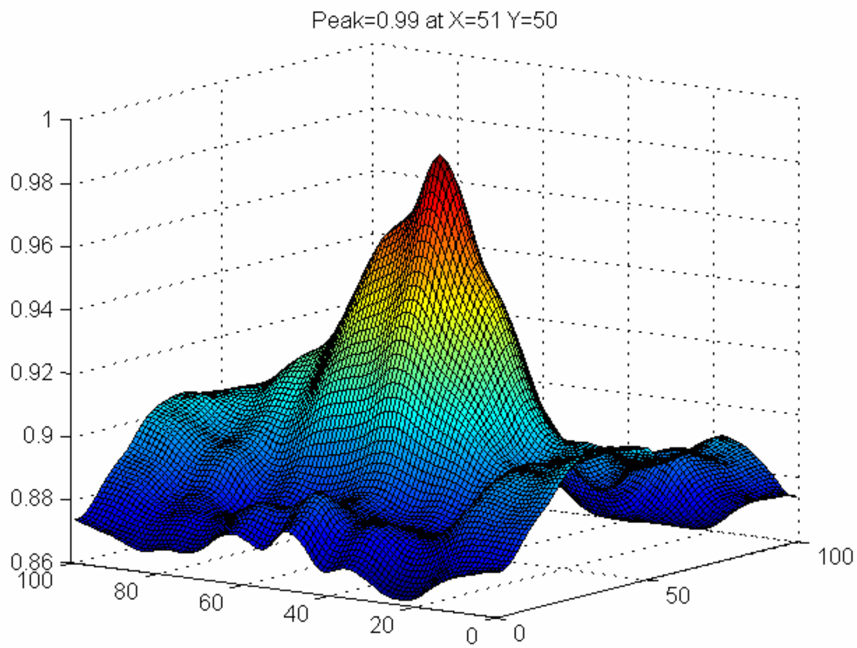


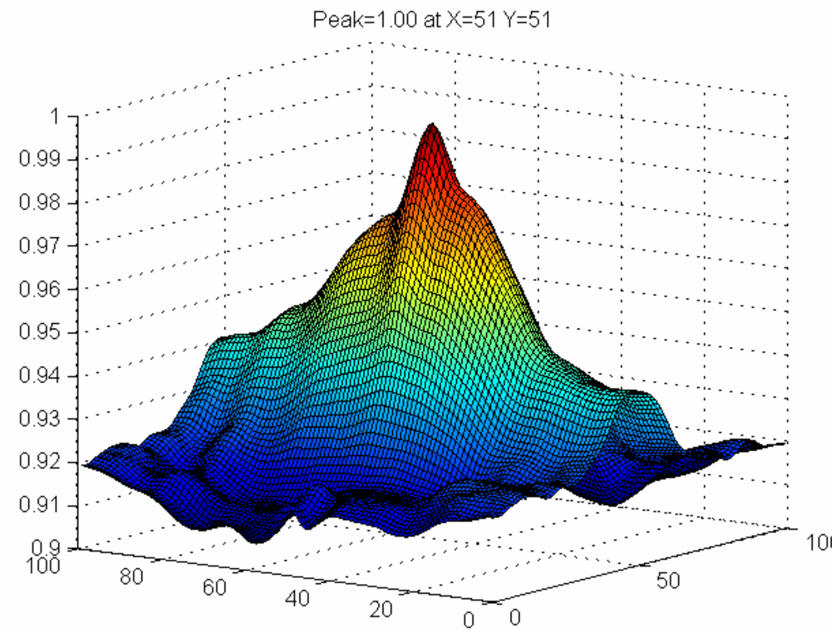


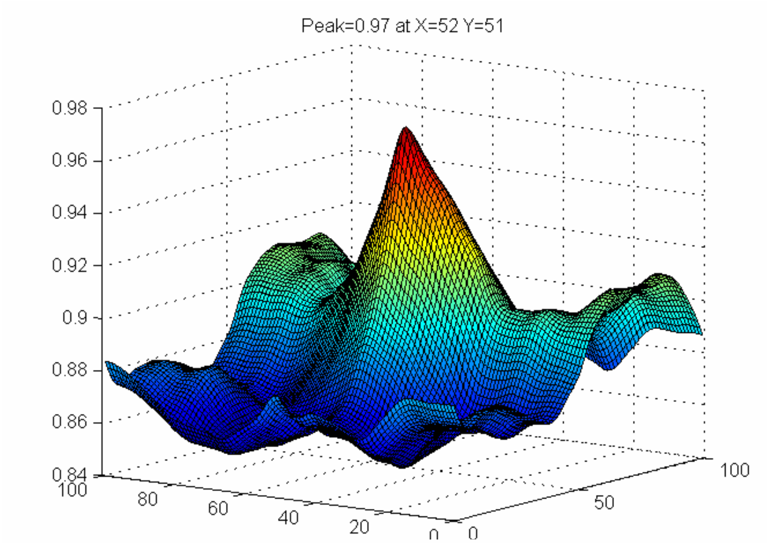
# Example Correlation Output Impostors using ECP-SDF

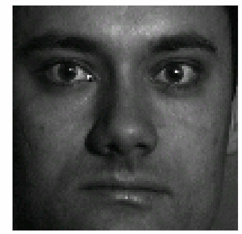


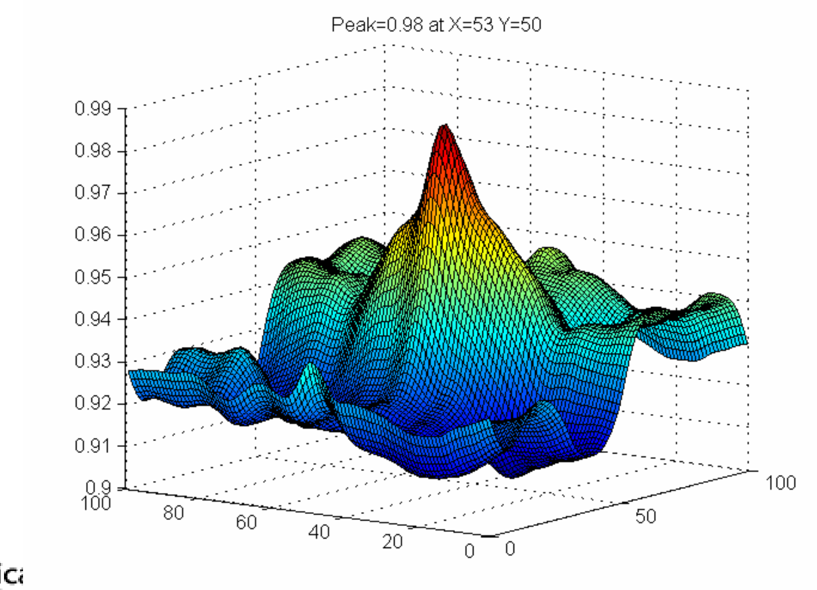




**Very high peaks (0.99 and 1.0 for impostor class! Not good!**

# Example correlation output for authentic people (but slightly different illumination)






# Conclusion – ECP-SDF

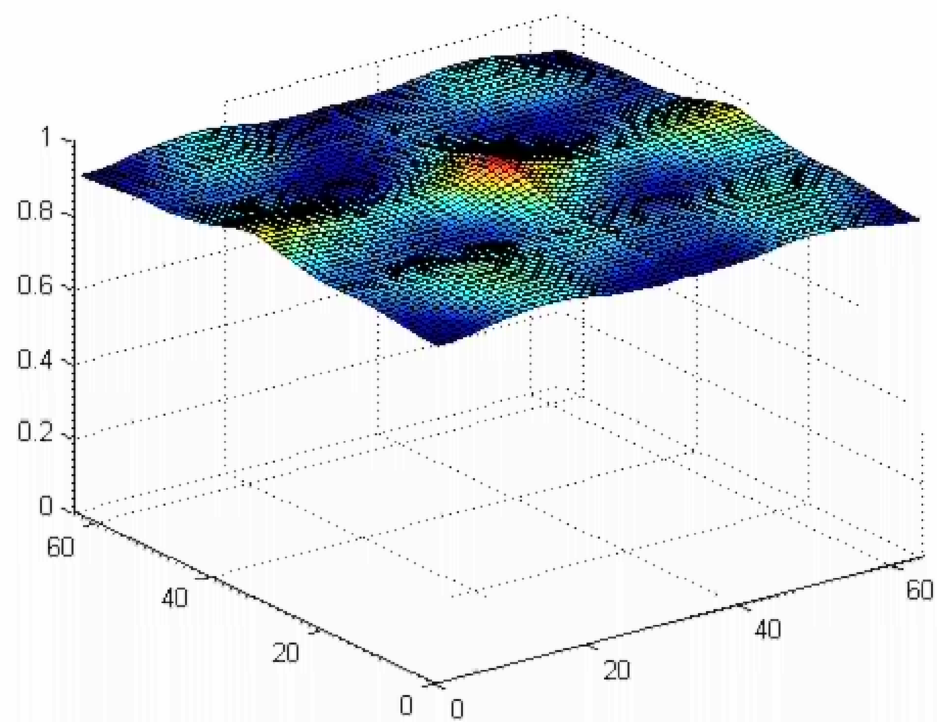
- ECP-SDF does not generalize well…atleast not for illumination variations.
- Discrimination is poor, as we saw we got high peaks for impostor classes.
- Does not guarantee peak is at origin, we may get a sharp peak at some other location.
- Solution:- Minimum Average Correlation Filters (MACE) filters, to produce sharp peaks.

# Minimum Average Correlation Energy (MACE) Filters

- MACE filter minimizes the average energy of the correlation outputs while maintaining the correlation value at the origin at a pre-specified level.
- Sidelobes are reduced greatly and discrimination performance is improved

**MACE Filter produces a sharp peak at the origin**

**Correlation Plane**



# MACE Filter Formulation

- Minimizing spatial correlation energy can be done directly in the frequency domain by expressing the *i*th correlation plane ($c_i$) energy $E_i$ as follows:

$$E_i = \tfrac{1}{d}\sum_{p=1}^{d} |c_i(p)|^2 = \tfrac{1}{d}\sum_{k=1}^{d} |H(k)|^2 |X_i(k)|^2 = \mathbf{h}^+ \mathbf{X}_i \mathbf{X}_i^* \mathbf{h} = \mathbf{h}^+ \mathbf{D}_i \mathbf{h}$$

**Parseval's Theorem!**

- The Average correlation plane energy for the $N$ training images is given by $E_{ave}$

$$E_{ave} = \tfrac{1}{N}\sum_{i=1}^{N} E_i = \mathbf{h}^+ \left[ \tfrac{1}{N}\sum_{i=1}^{N} \mathbf{X}_i \mathbf{X}_i^* \right] \mathbf{h} = \mathbf{h}^+ \mathbf{D} \mathbf{h}$$

# MACE Filter Formulation (Cont'd.)

- The value at the origin of the correlation of the *i-th* training image is:

$$c_i(0) = \sum_{p=1}^{d} H(p)^* X_i(p) e^{j2\pi 0 p} = \sum_{p=1}^{d} H(p)^* X_i(p) = \mathbf{h}^+ \mathbf{x}_i$$

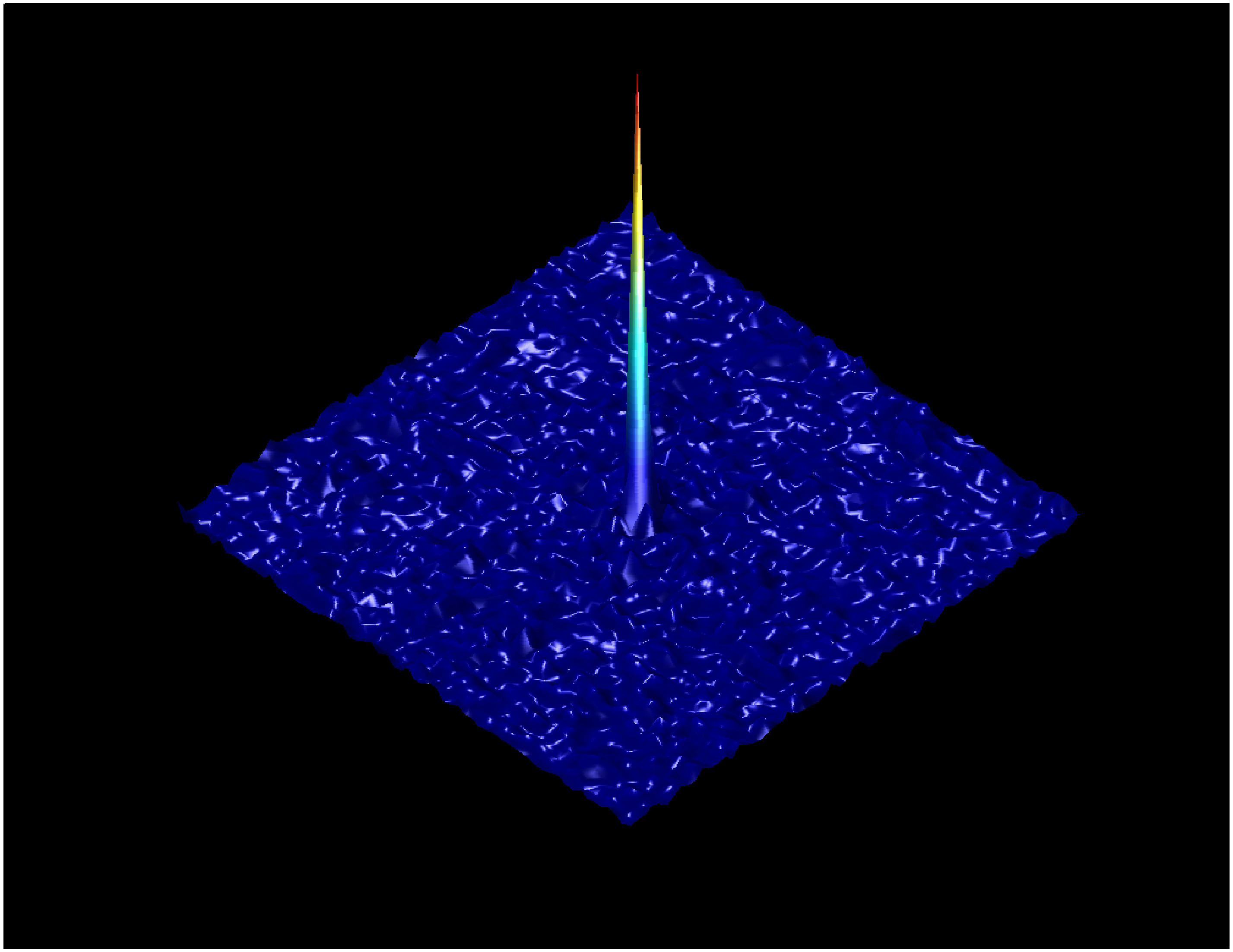
- Specify the correlation peak values for all the training image using column vector u

$$\mathbf{X}^+ \mathbf{h} = \mathbf{u}$$

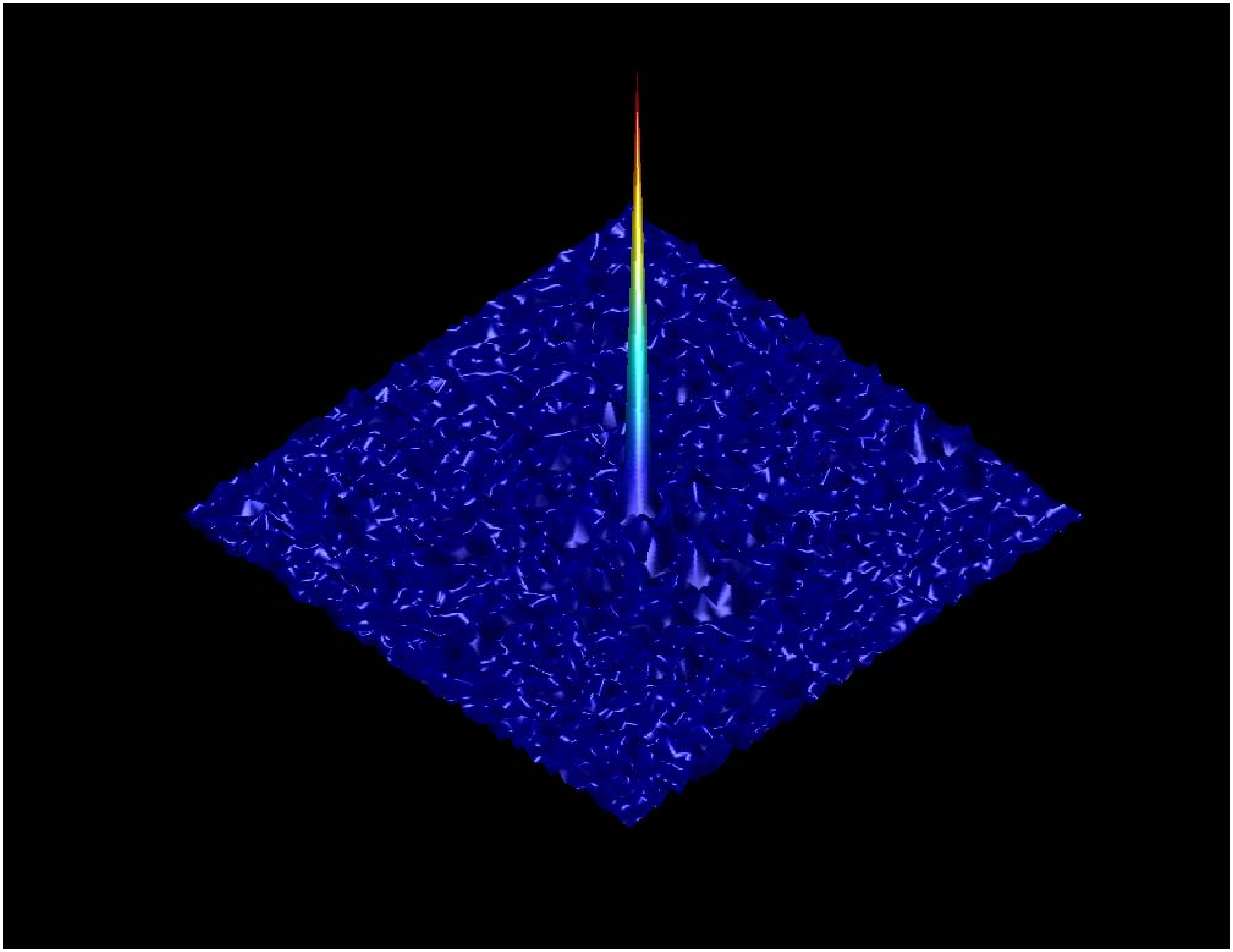
- Minimizing the average correlation energies $h^+Dh$ subject to the constraints $X^+h=u$ leads to the MACE filter solution.

$$h_{MACE} = D^{-1} X (X^+ D^{-1} X)^{-1} u$$

# Example Correlation Outputs from an Authentic



# Example Correlation Outputs from an Impostor

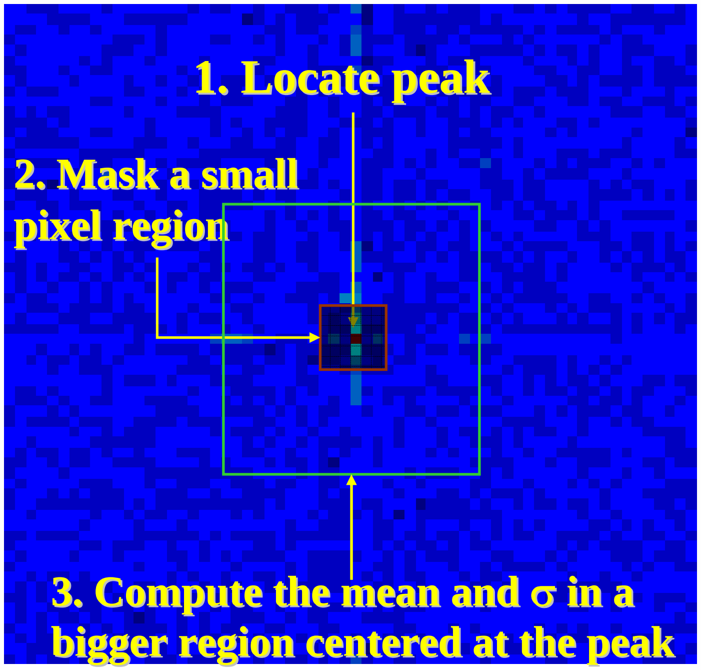


# Different Figure of Merit (FOM)

- For Matched Filter we showed that the peak height is what was used for recognition.
- For MACE filters, the optimization is not only to keep the peak height at 1 but also to create sharp peaks.
- Thus it makes sense to use a metric that can measure whether we have achieved this optimization
- We need a metric that can measure the Peak sharpness! As images that resemble the training classes will produce a sharp peak whereas impostor classes will not produce sharp peaks as they were not optimized to do so!

# Peak to Sidelobe Ratio (PSR)

- PSR invariant to constant illumination changes



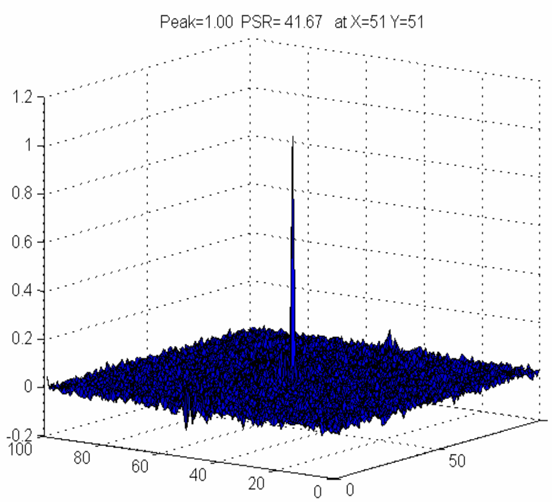

$$PSR = \frac{Peak - mean}{\sigma}$$

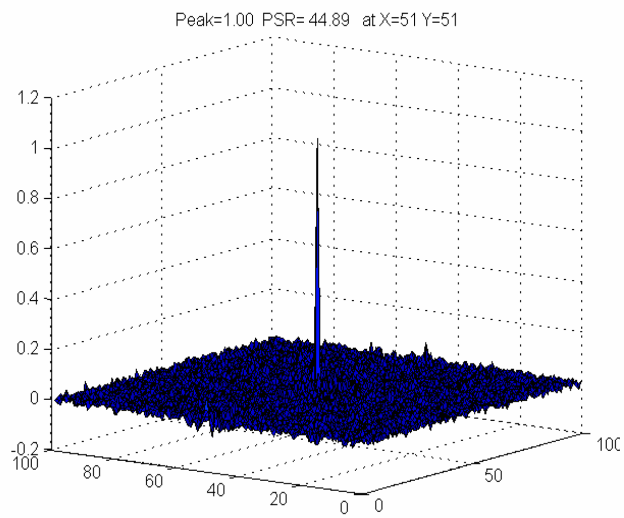
- Match declared when PSR is large, i.e., peak must not only be large, but sidelobes must be small.

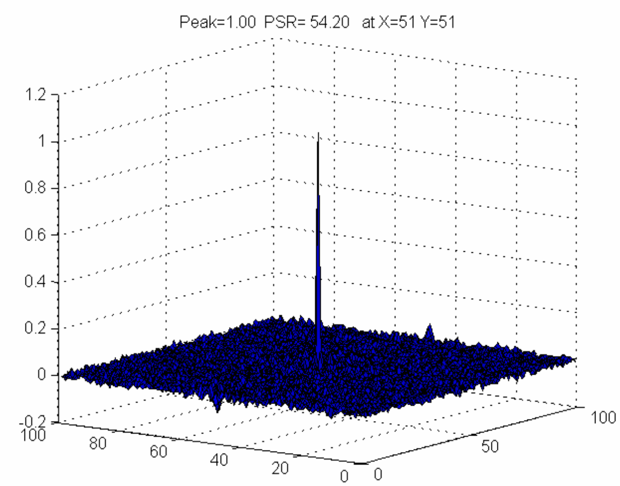
# Example Correlation Outputs

- Use these images for training (peak=1 for all correlation outputs)



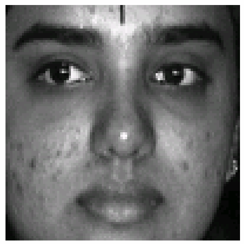


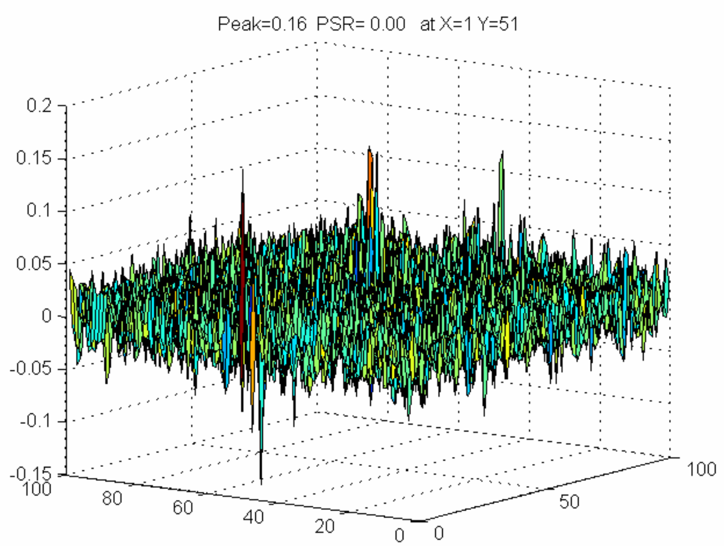



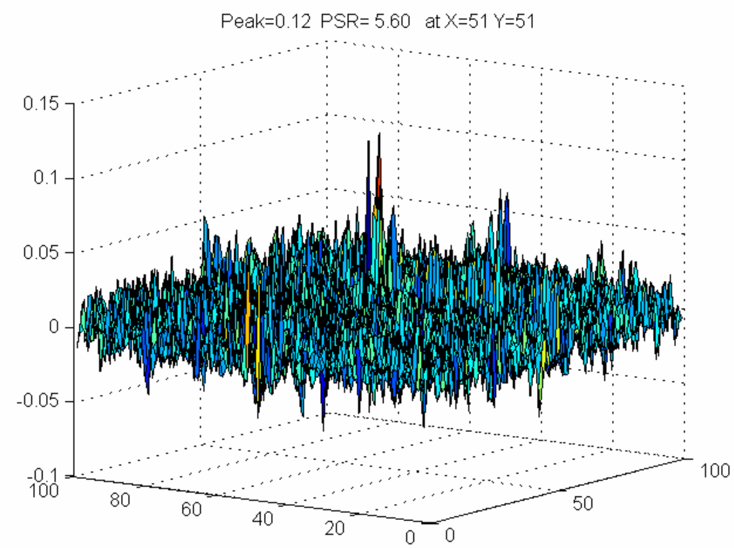




**PSR: 41.6 44.8 54.2**

# Example Correlation Output Impostors using MACE

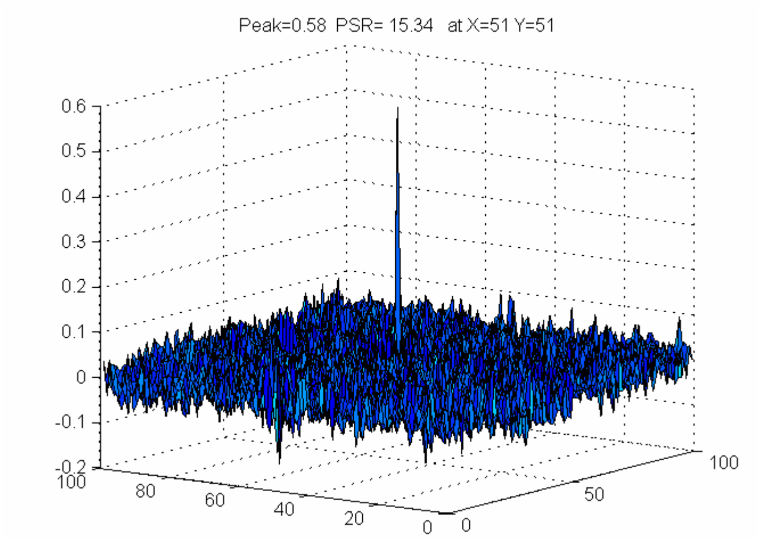


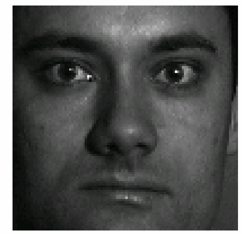


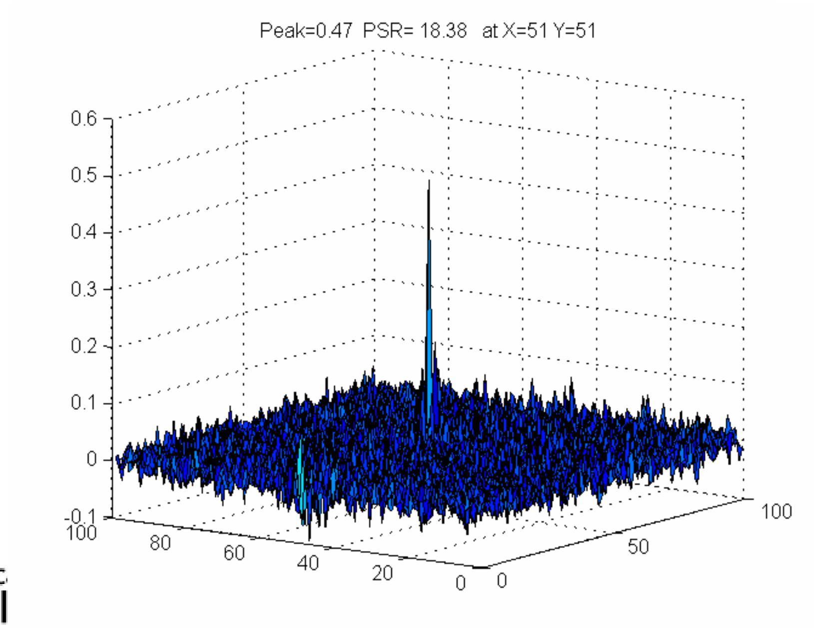




**No discernible peaks (0.16 and 0.12 and PSR 5.6 for impostor class! Very good!**

# Example correlation output for authentic people (but slightly different illumination)



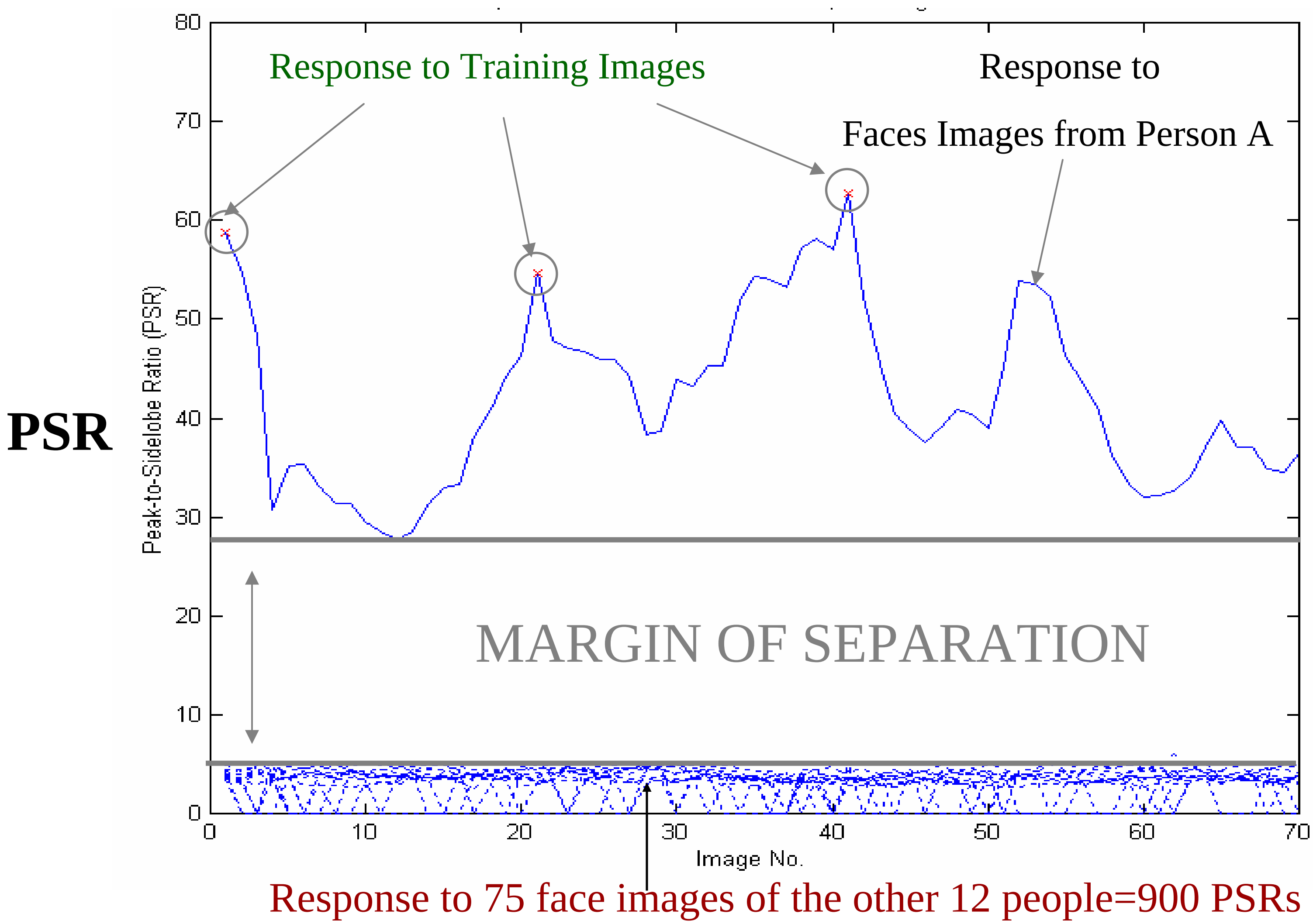



# Facial Expression Database

- Facial Expression Database (AMP Lab, CMU)
  - 13 People
  - 75 images per person
  - Varying Expressions
  - 64x64 pixels
  - Constant illumination

- 1 filter per person made from 3 training images

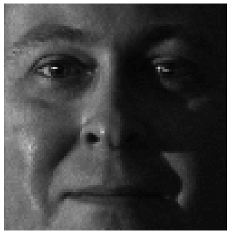
# PSRs for the Filter Trained on 3 Images



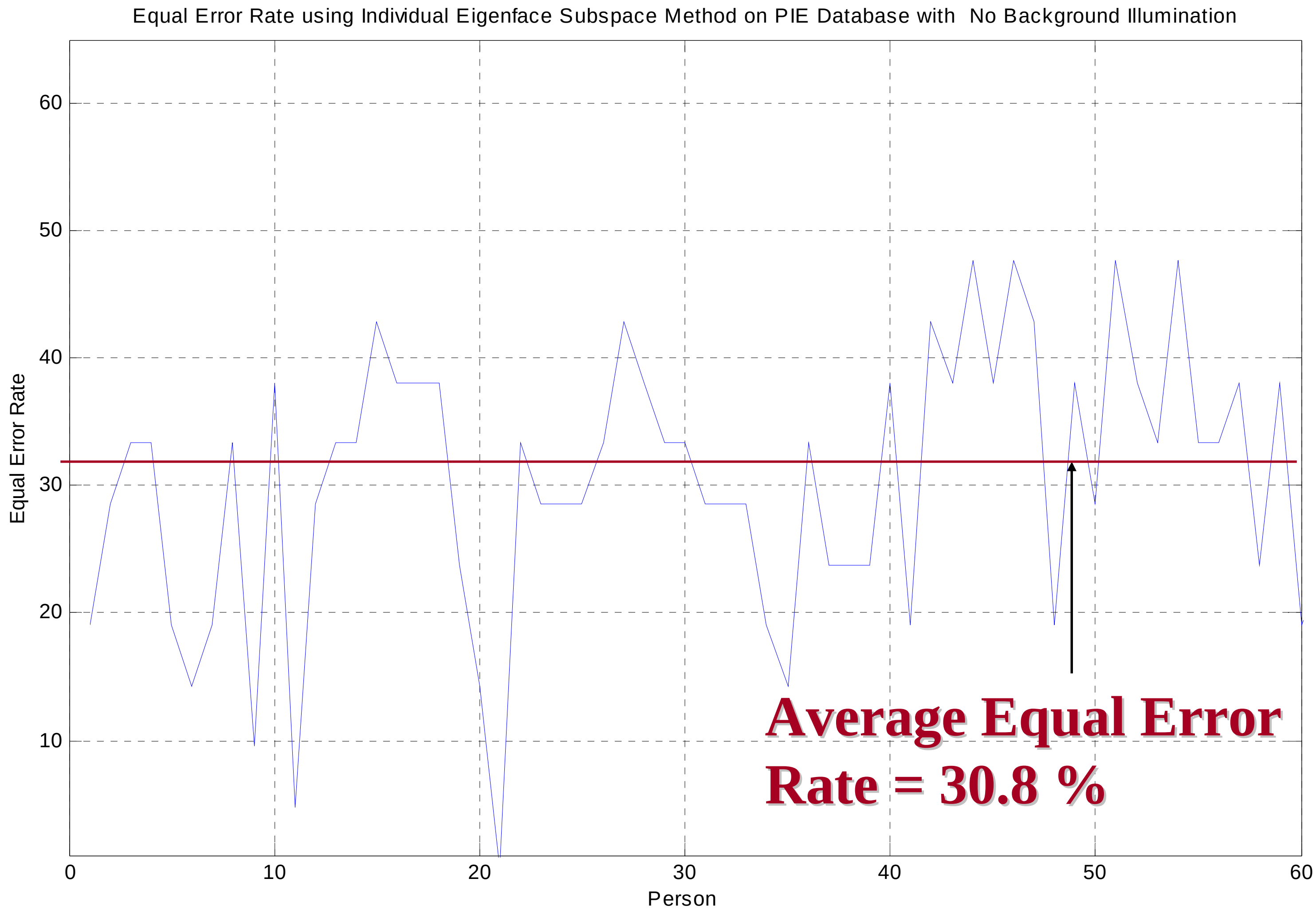
# 49 Faces from PIE Database illustrating the variations in illumination

# Training Image selection

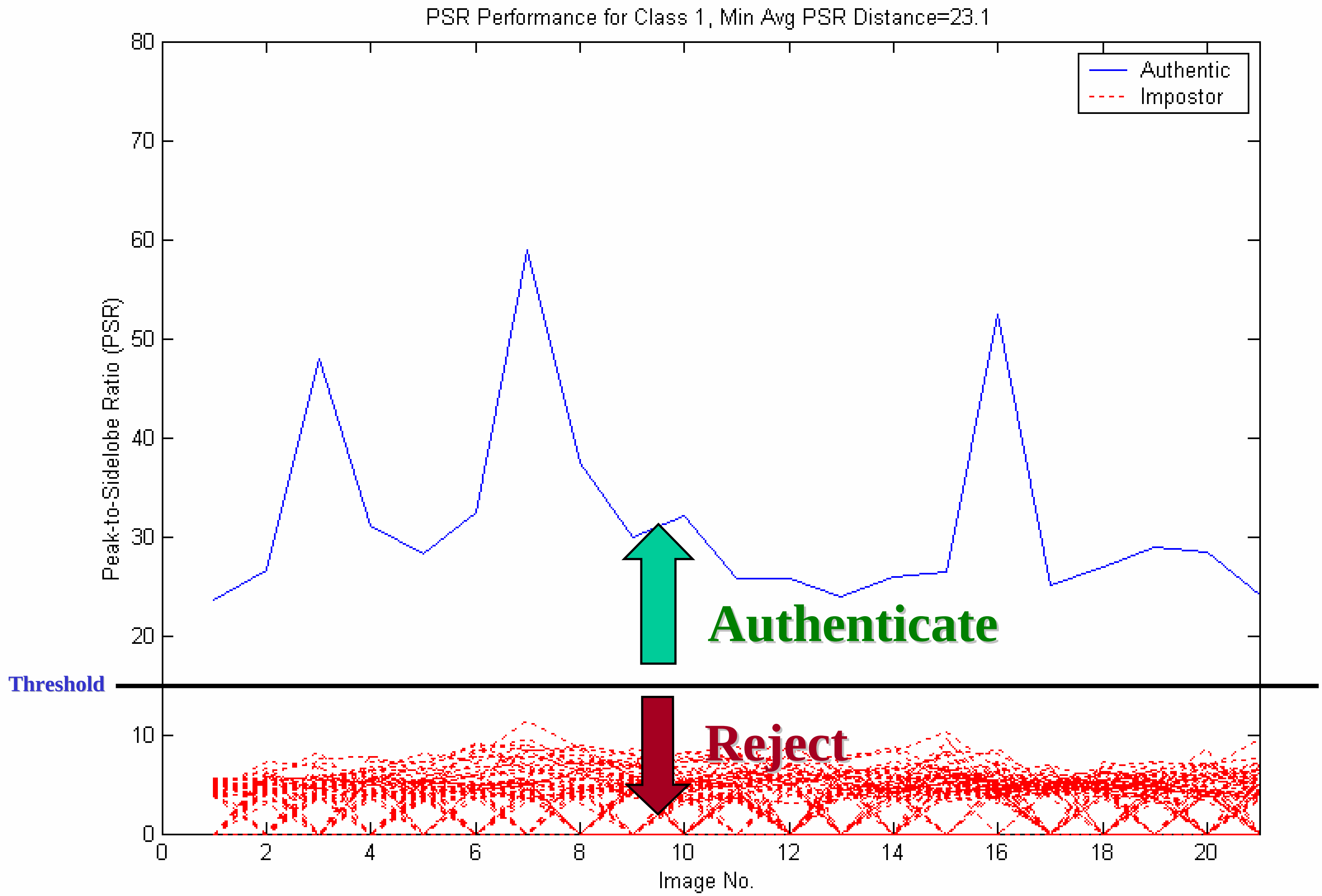
- We used *three* face images to synthesize a correlation filter
- The three selected training images consisted of 3 extreme cases (dark left half face, normal face illumination, dark right half face).

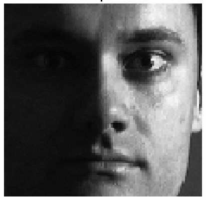


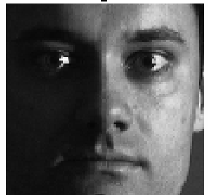
# EER using IPCA with no Background illumination



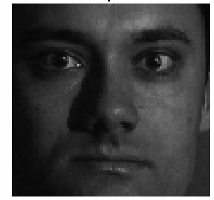
# EER using Filter with no Background illumination

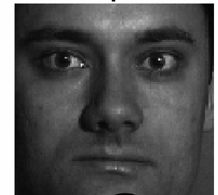


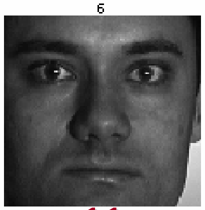
1


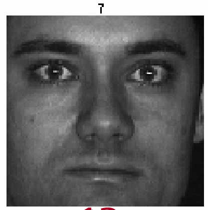
2


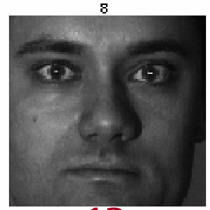
3

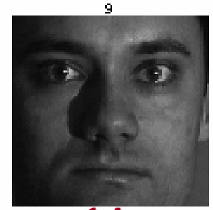

4


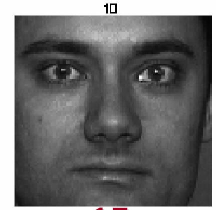
5


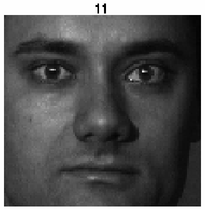
6


7


8


9


10


11


12

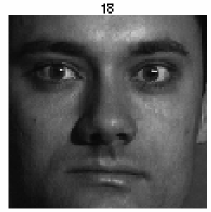

13

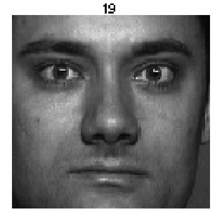

14


15


16


17


18


19


20


21

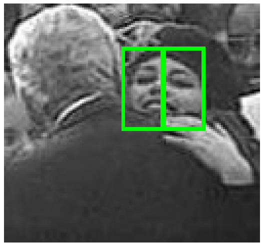

**21 different illuminations of a person's face in the database**

# Recognition Accuracy using Frontal Lighting Training Images

| PIE dataset (face images captured with room lights off) | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Frontal Lighting | IPCA | | 3D Linear Subspace | | Fisherfaces | | MACE Filters | | UMACE Filters | |
| Training Images | # Errors | %Rec Rate | # Errors | %Rec Rate | # Errors | %Rec Rate | # Errors | % Rec Rate | # Errors | % Rec Rate |
| 5,6,7,8,9,10, 11,18,19,20 | 33 | 97.6% | 31 | 97.3% | 36 | 97.3% | 0 | 100% | 0 | 100% |
| 5,6,7,8,9, 10 | 110 | 91.4% | 40 | 97.1% | 145 | 89.3% | 1 | 99.9% | 0 | 100% |
| 5,7,9,10 | 337 | 72.4% | 93 | 93.2% | 390 | 71.4% | 1 | 99.9% | 3 | 99.7% |
| **7,10,19** | **872** | **36.1%** | **670** | **50.9%** | **365** | **73.3%** | **10** | **99.1%** | **10** | **99.1%** |
| 8,9,10 | 300 | 78.0% | 30 | 97.8% | 244 | 82.1% | 1 | 99.9% | 1 | 99.9% |
| 18,19,20 | 122 | 91.0% | 22 | 98.4% | 79 | 94.2% | 2 | 99.9% | 1 | 99.9% |

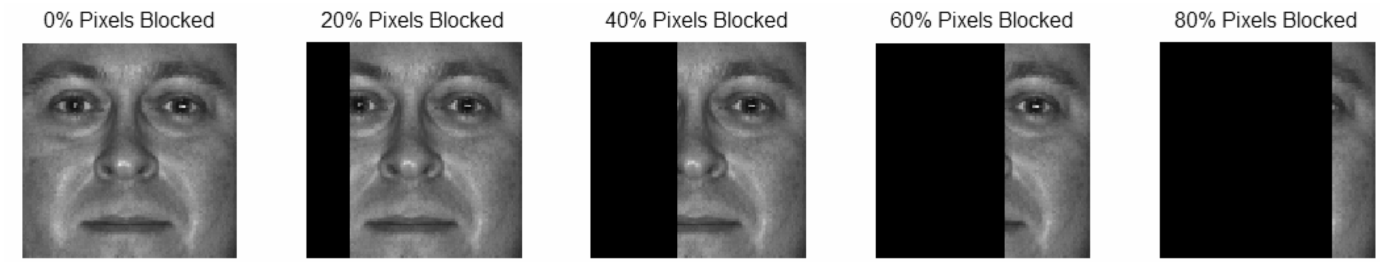
# Face Identification from Partial Faces

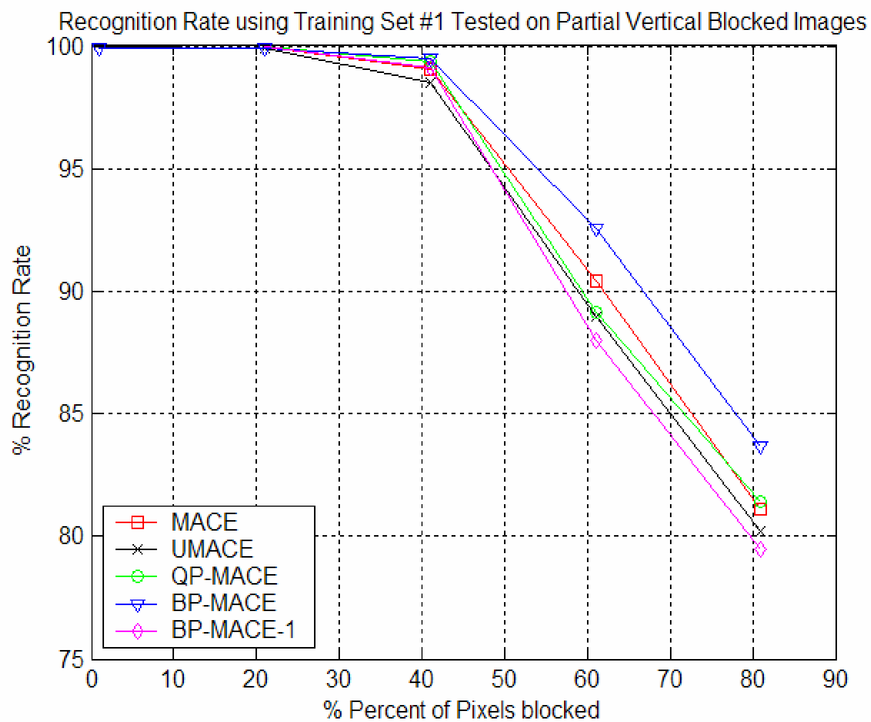
- We have shown that these correlation filters seem to be tolerant to illumination variations, even when half the face is completely black and still achieve excellent recognition rates.
- What about partial face recognition?
- In practice a face detector will detect and retrieve part of the face (another type of registration error). In many cases, occluded by another face or object. Other face recognition methods fail in this circumstance.

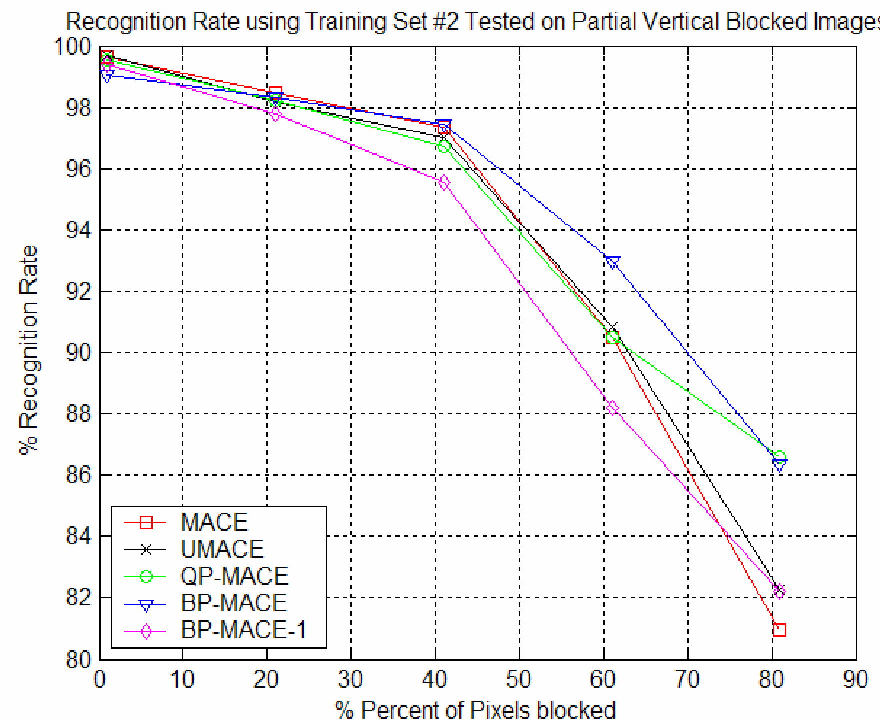


*M. Savvides, B.V.K. Vijaya Kumar and P.K. Khosla, "Robust, Shift-Invariant Biometric Identification from Partial Face Images", Defense & Security Symposium, special session on Biometric Technologies for Human Identification (OR51) 2004.

# Vertical cropping of test face image pixels (correlation filters are trained on FULL size images)



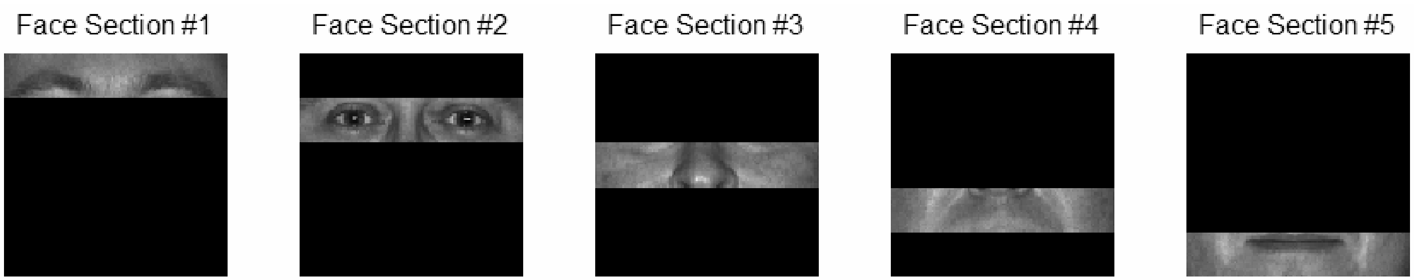



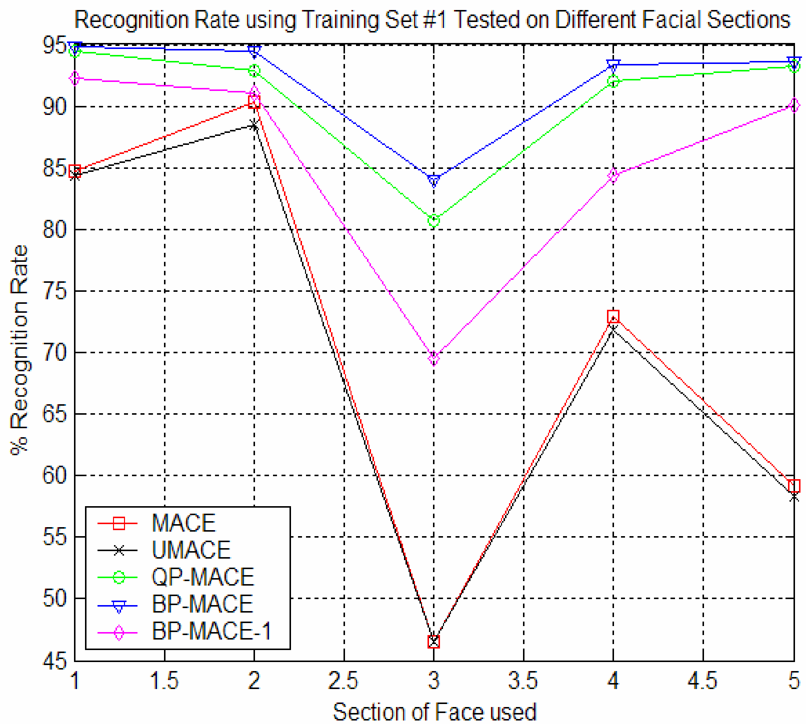




**Using Training set #1 (3 extreme lighting images)**

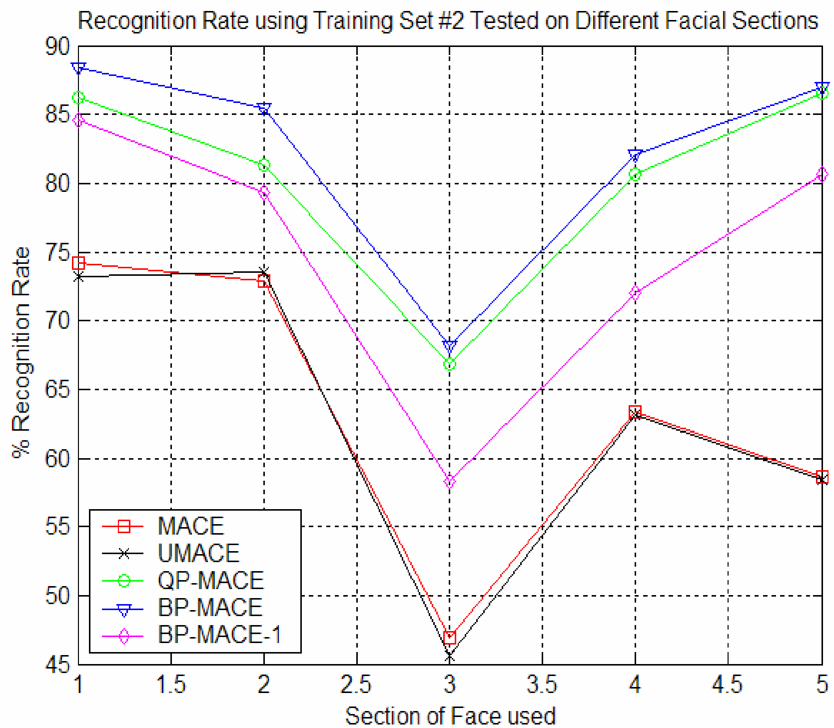
**Using Training set #2 (3 frontal lighting images)**

# Recognition using selected face regions



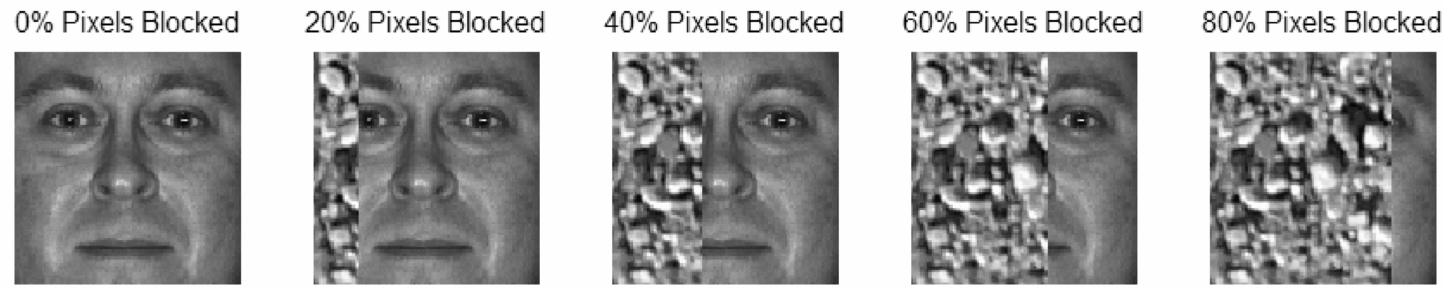



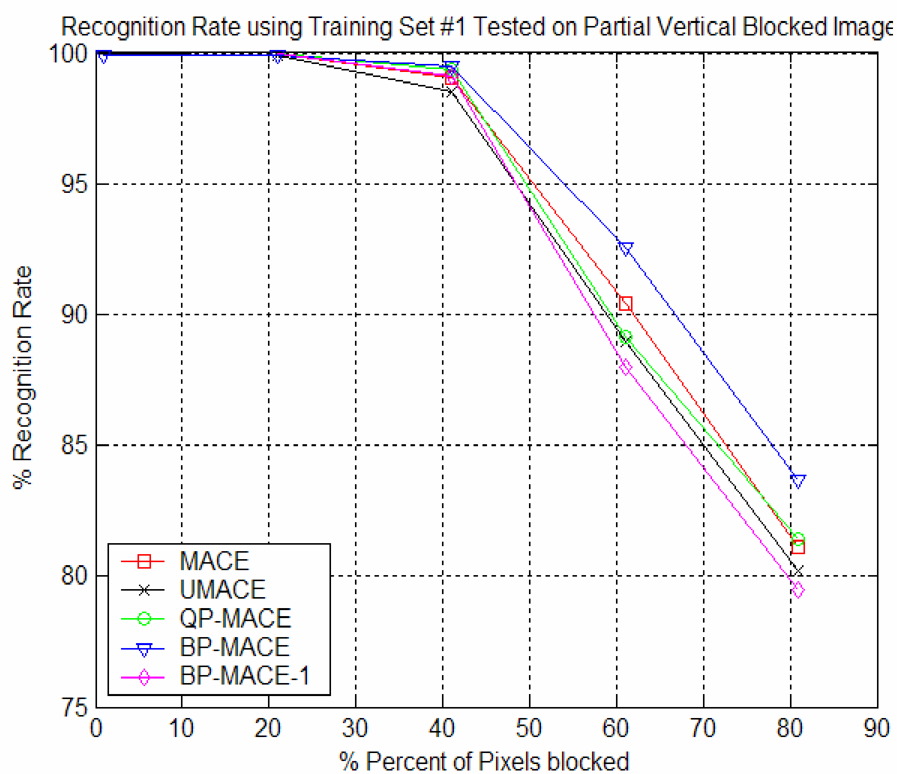

**Using Training set #1 (3 extreme lighting images)**



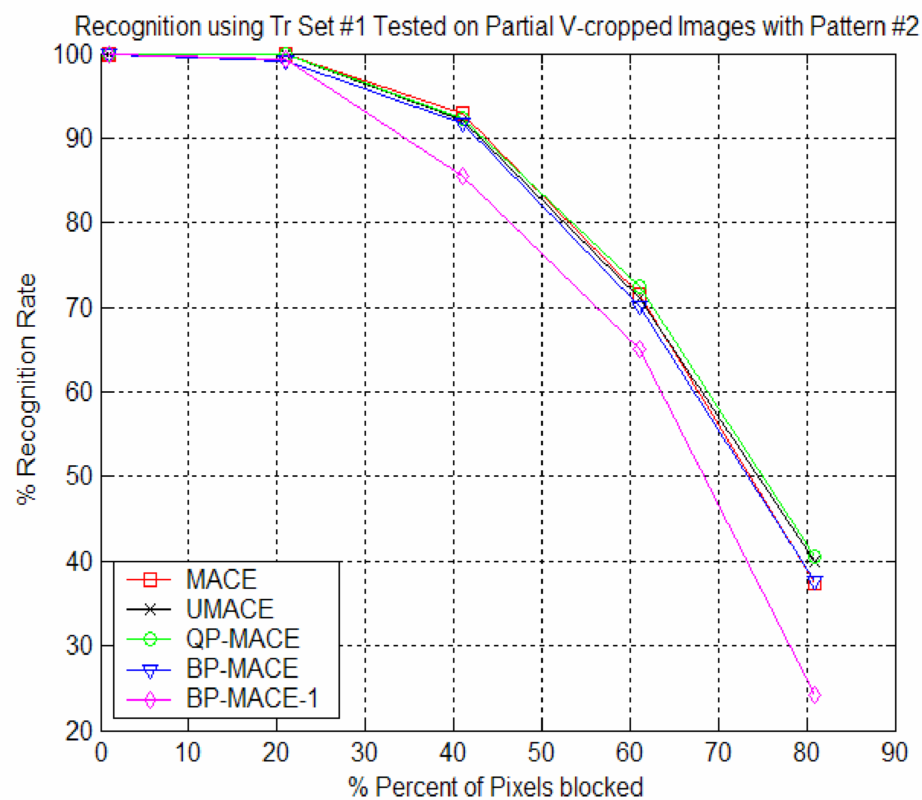

**Using Training set #2 (3 frontal lighting images)**

# Vertical crop + texture #2




**Zero intensity background**

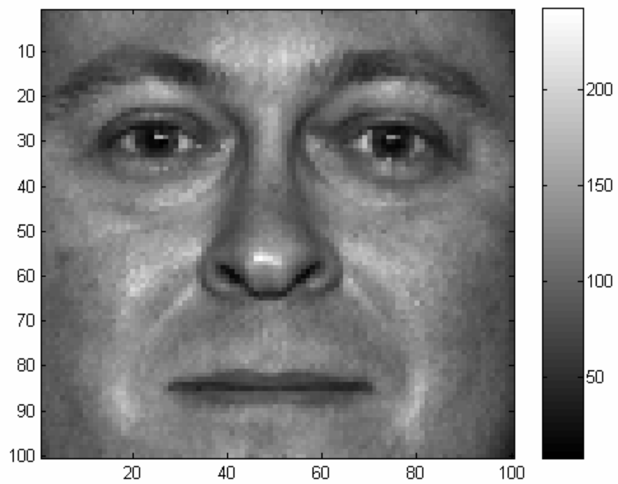



**Textured background**



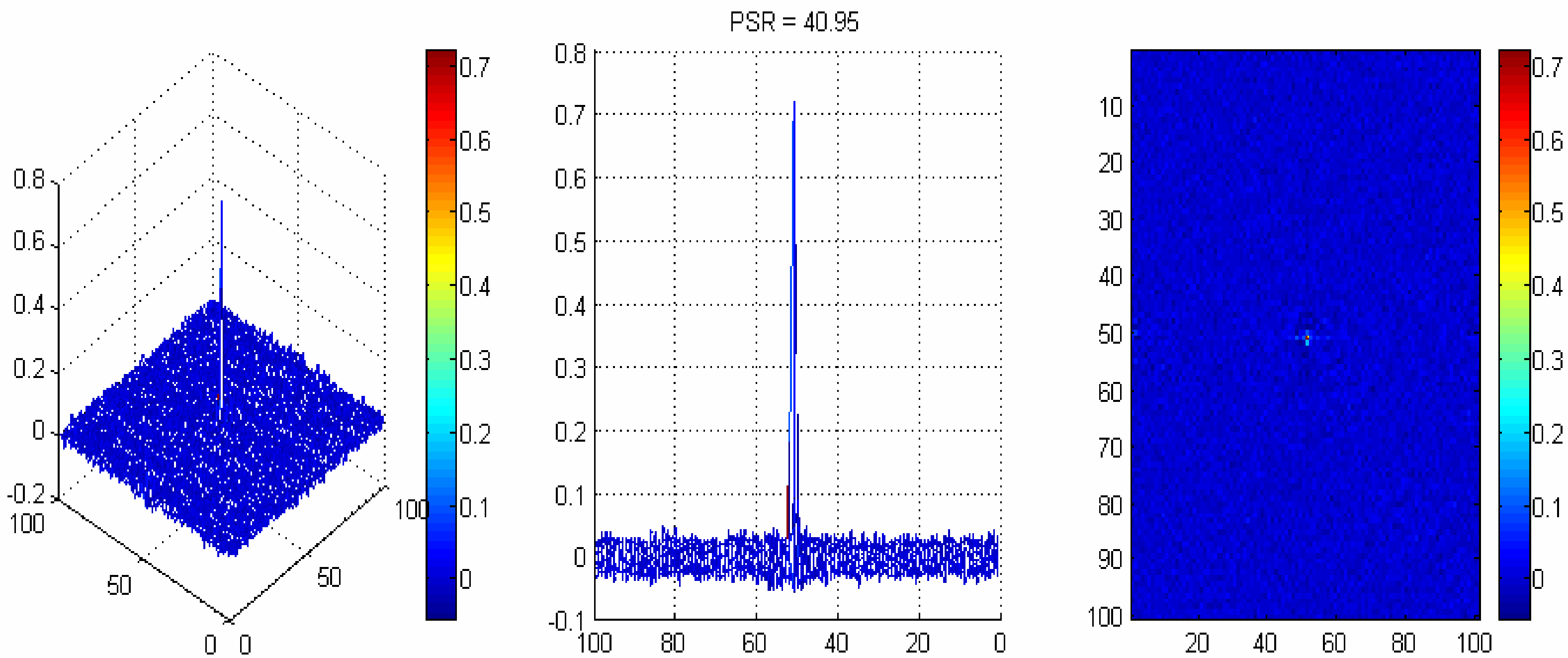
Electrical & Computer ENGINEERING

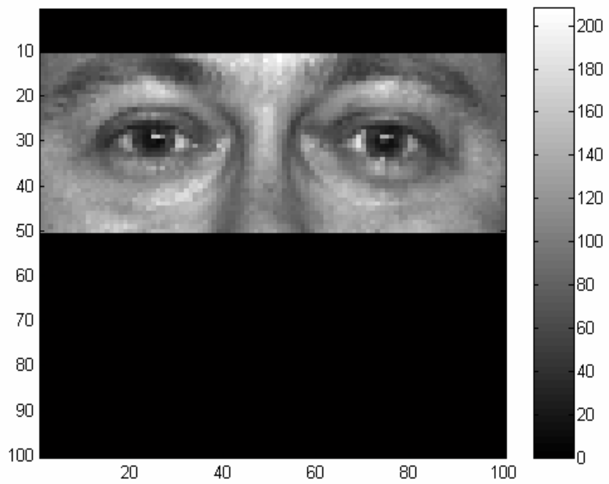
*M. Savvides, B.V.K. Vijaya Kumar and P.K. Khosla, "Robust, Shift-Invariant Biometric Identification from Partial Face Images", Defense & Security Symposium, special session on Biometric Technologies for Human Identification (OR51) 2004.



**Train filter on illuminations 3,7,16.**

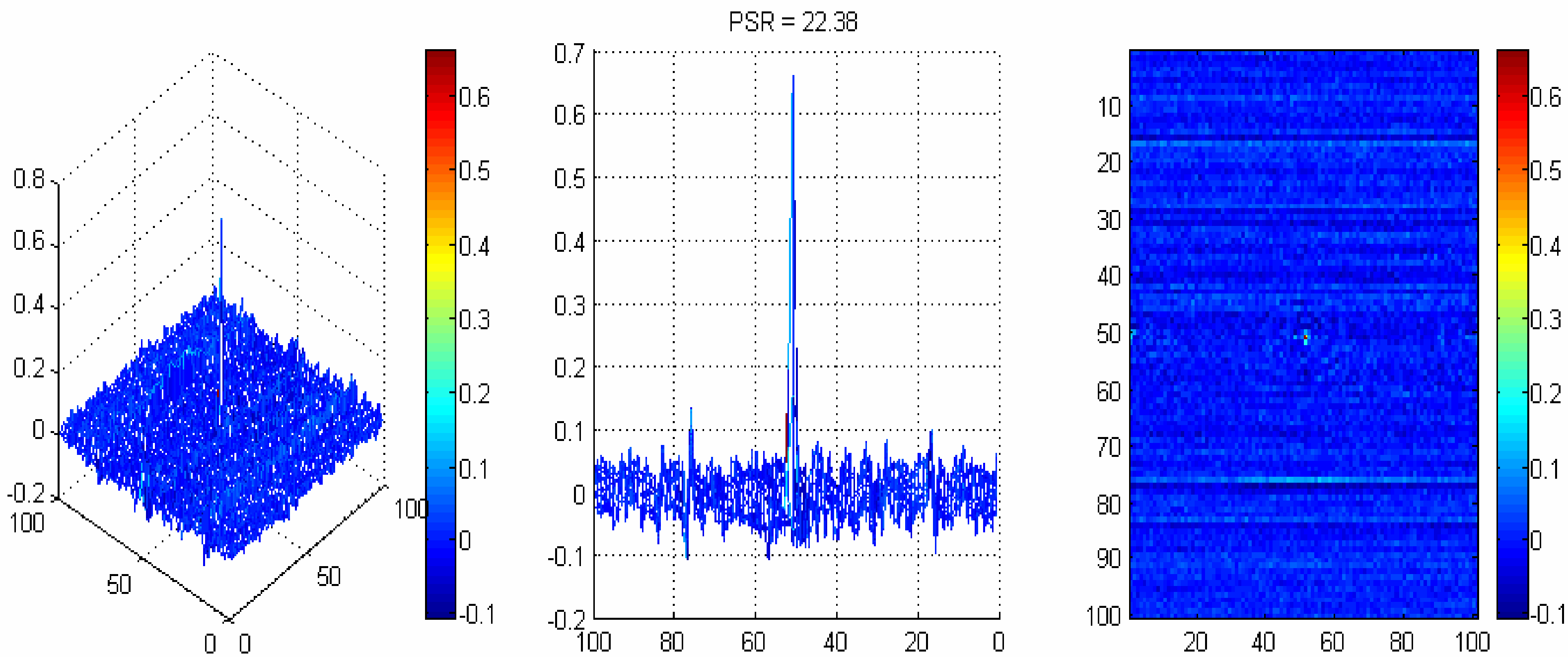
**Test on 10.**

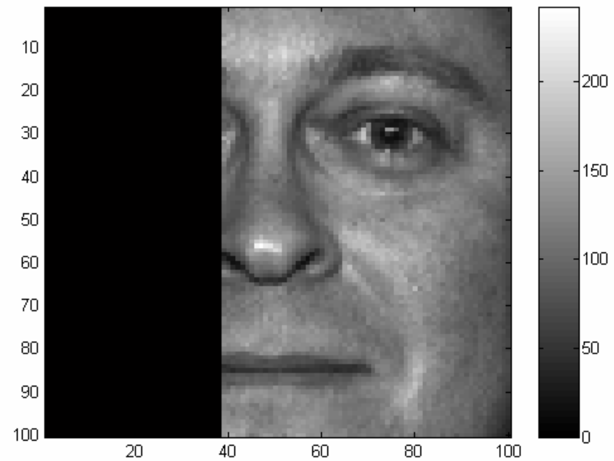




**Using same Filter trained before,**

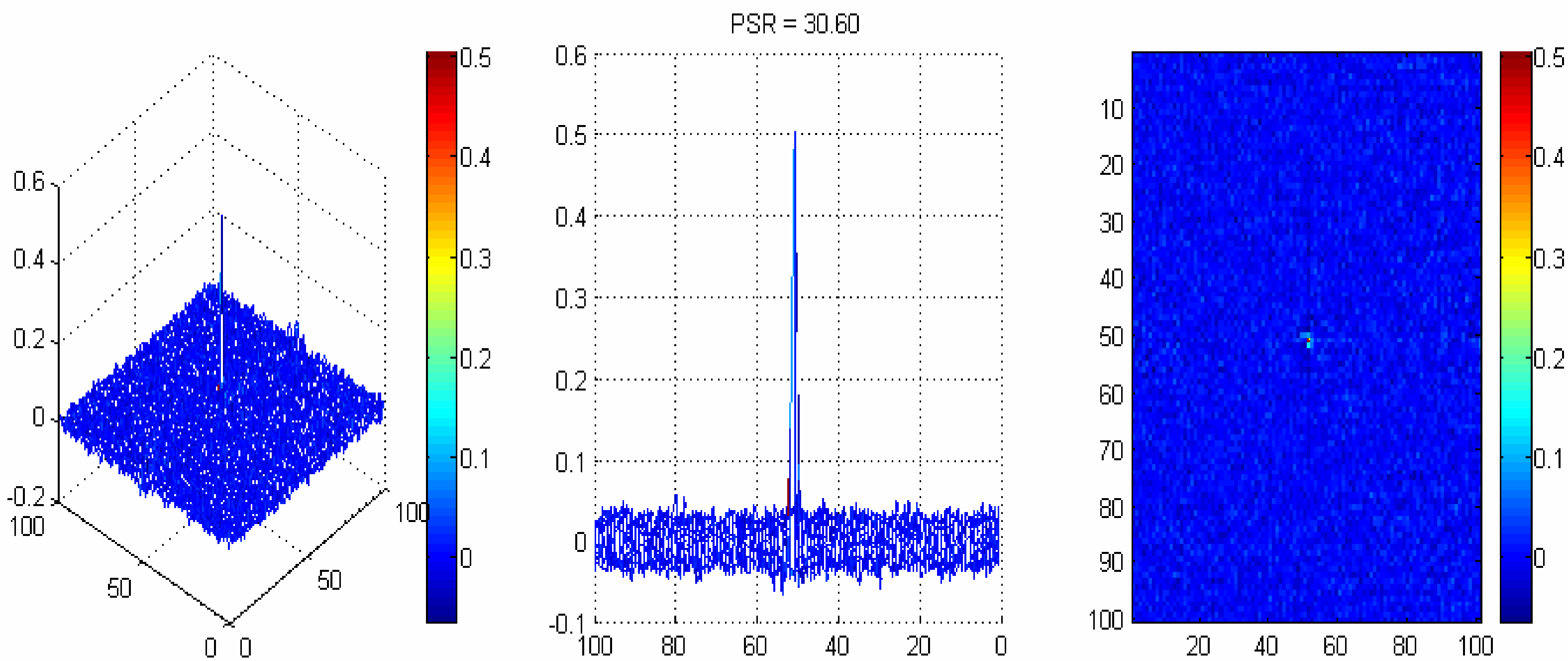
**Perform cross-correlation on cropped-face shown on left**

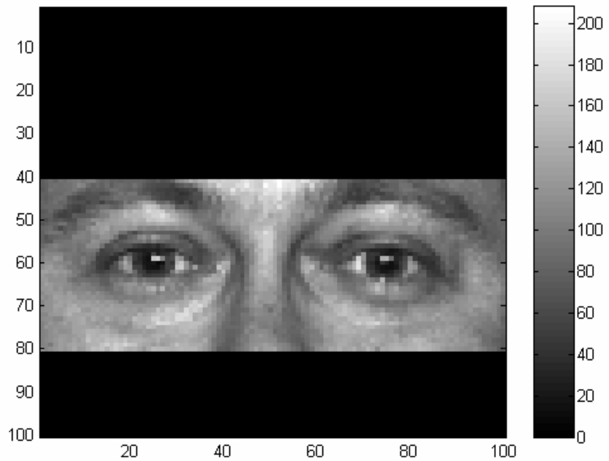




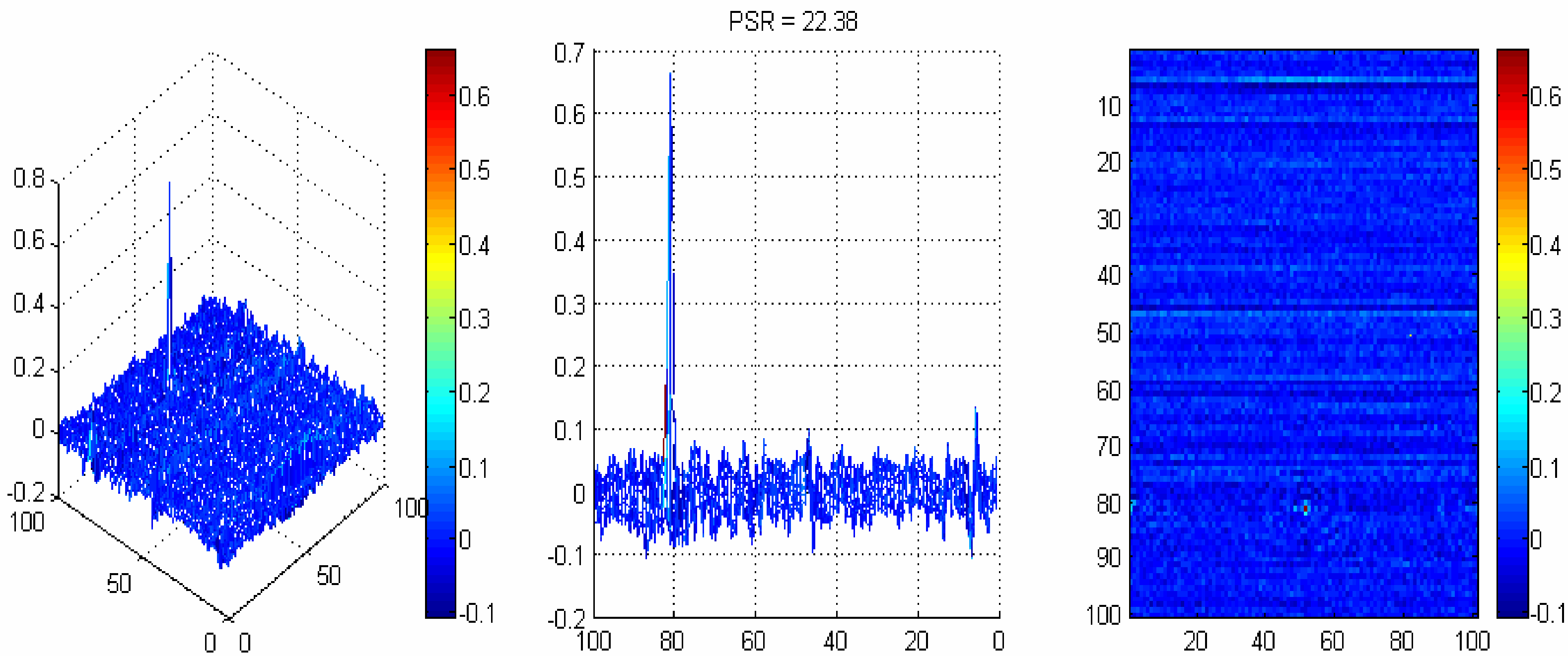
**Using same Filter trained before,**

**Perform cross-correlation on cropped-face shown on left.**

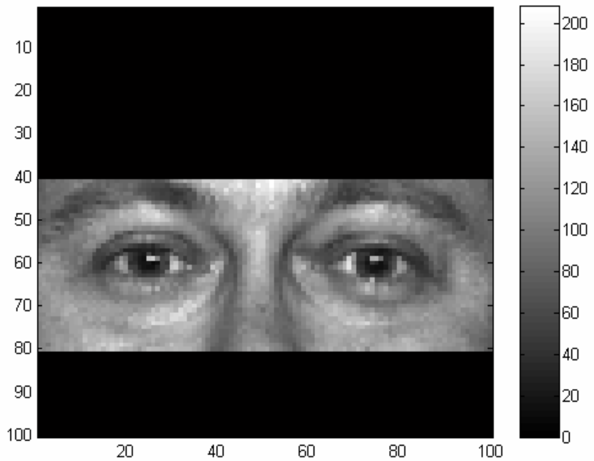




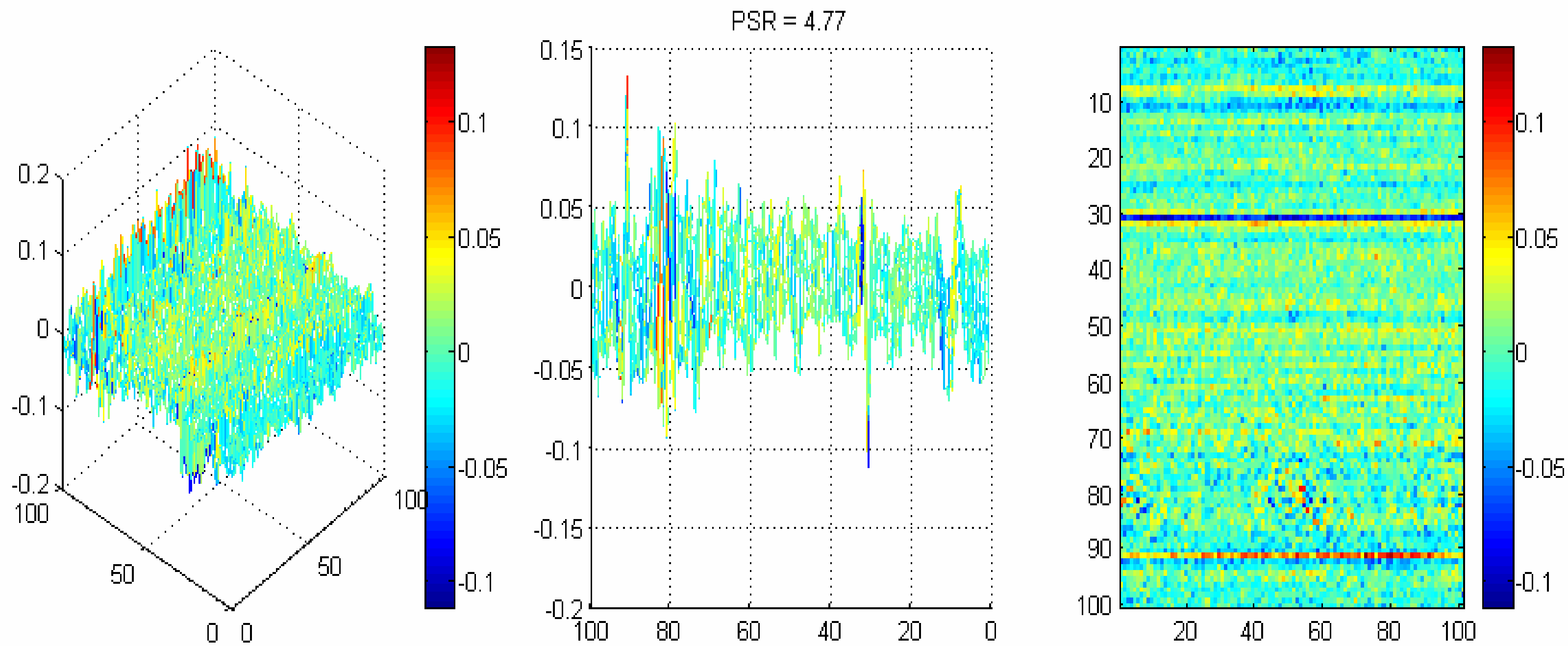
- **CORRELATION FILTERS ARE SHIFT-INVARIANT**
- **Correlation output is shifted down by the same amount of the shifted face image, PSR remains SAME!**




*M.Savvides and B.V.K. Vijaya Kumar, "Efficient Design of Advanced Correlation Filters for Robust Distortion-Tolerant Face Identification", IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) 2003.



- Using SOMEONE ELSE'S Filter,…. Perform cross-correlation on cropped-face shown on left.
- As expected very low PSR.

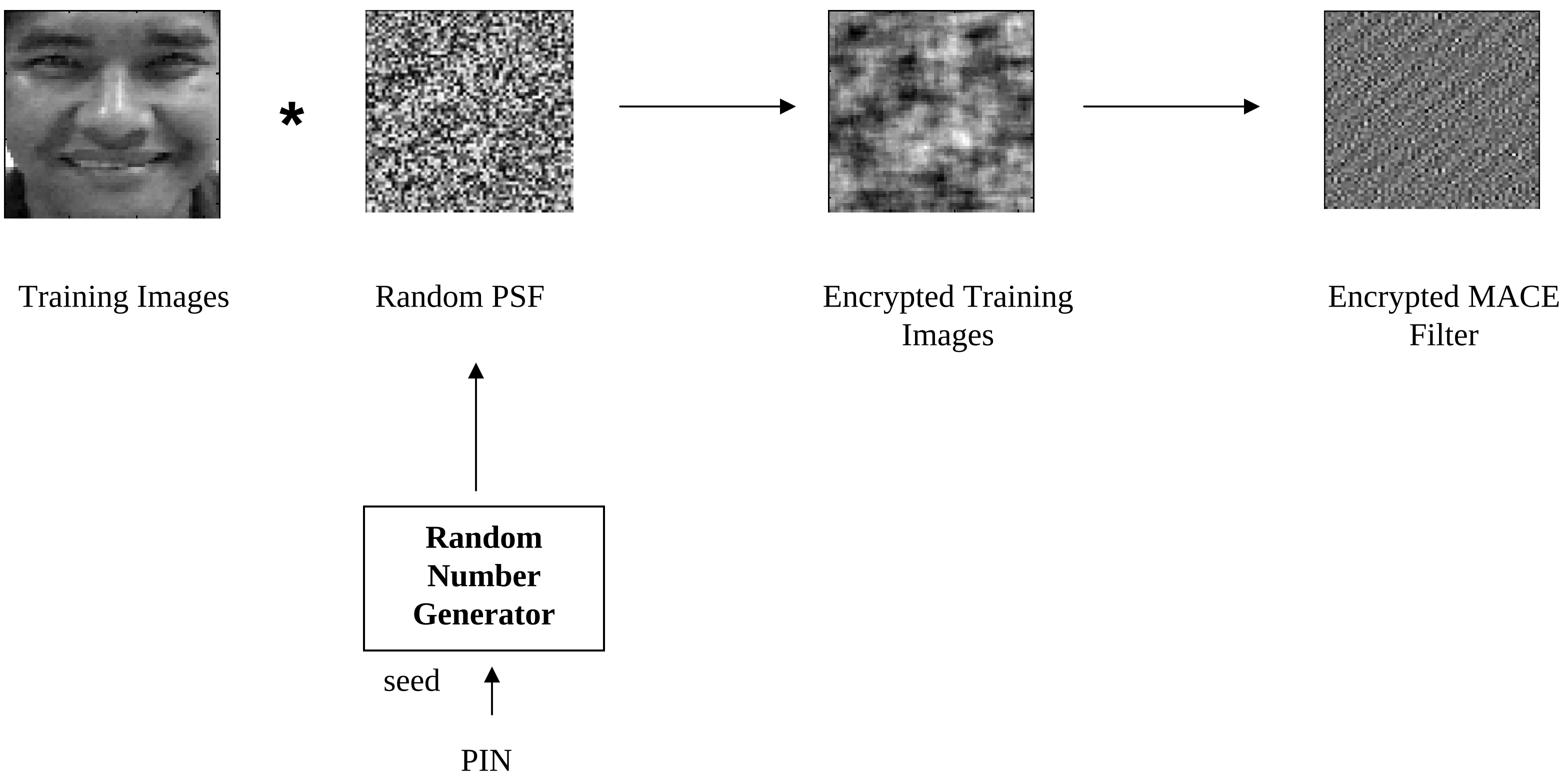


# Cancellable Biometric Filters:-practical ways of deploying correlation filter biometrics

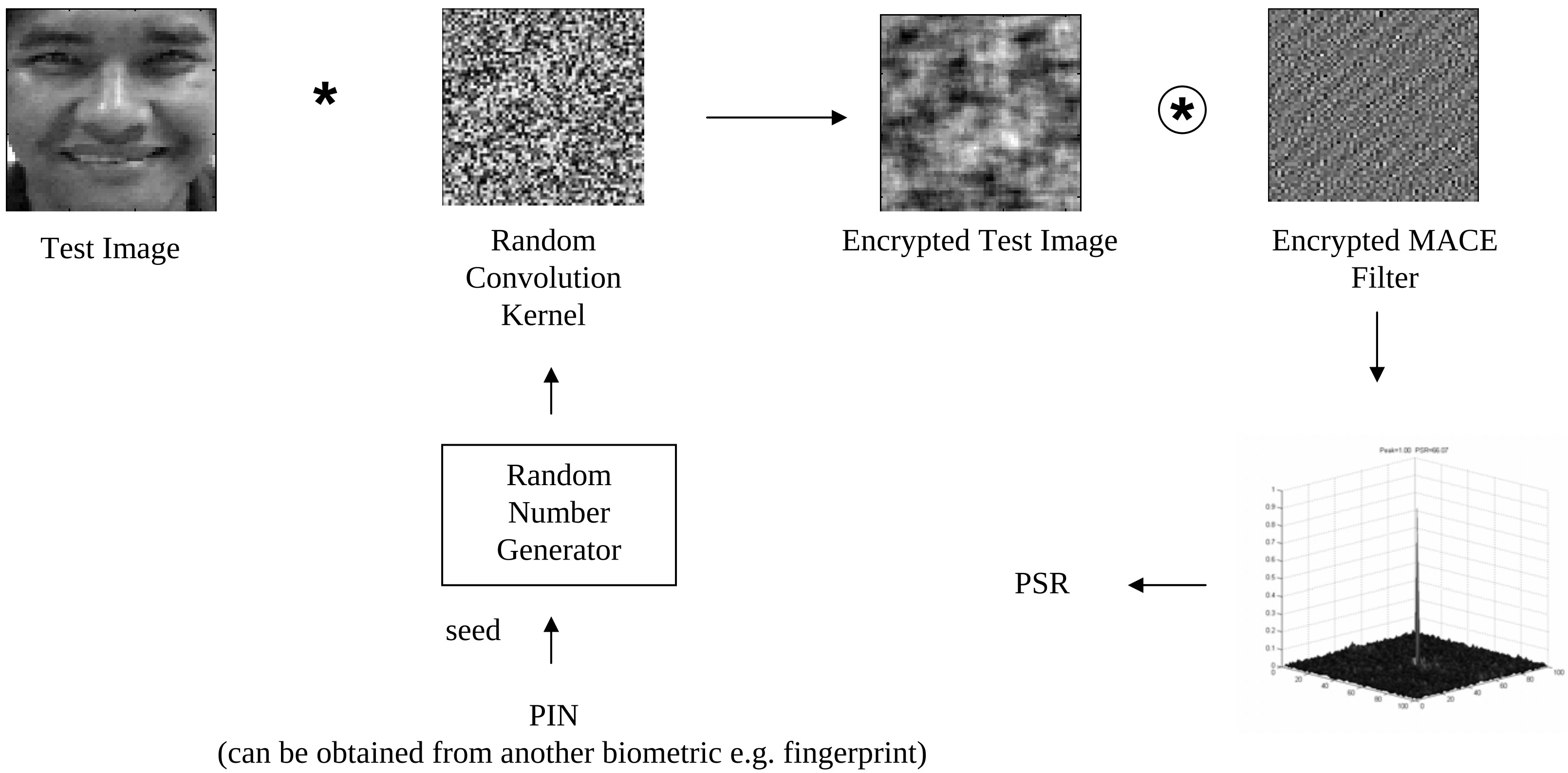
- A biometric filter (stored on a card) can be lost or stolen
  - Can we re-issue a different one (just as we re-issue a different credit card)?
  - There are only a limited set of biometric images per person (e.g., only one face)
  - We can use standard encryption methods to encrypt the biometrics and then decrypt them for use during the recognition stage, however there is a 'window' of opportunity where a hacker can obtain the decrypted biometric during the recognition stage.
  - We have figure out a way to encrypt them and 'work' or authenticate in the encrypted domain and NOT directly in the original biometric domain.

*M. Savvides, B.V.K. Vijaya Kumar and P.K. Khosla, "Authentication-Invariant Cancelable Biometric Filters for Illumination-Tolerant Face Verification", Defense & Security Symposium, special session on Biometric Technologies for Human Identification, 2004.

# Enrollment Stage



# Authentication Stage

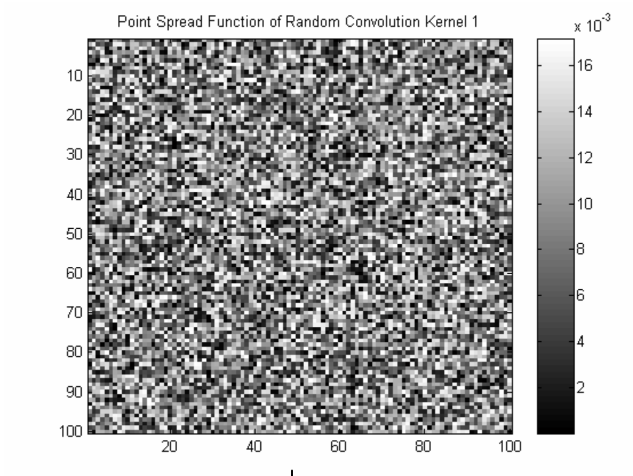


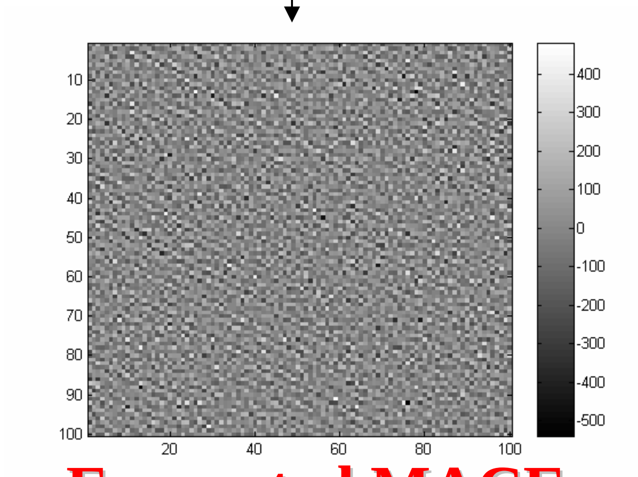
# What about performance?

- We can show theoretically that performing this convolution pre-processing step does not affect resulting Peak-to-Sidelobe ratios.

- Thus, working in this encrypted domain does not change the verification performance

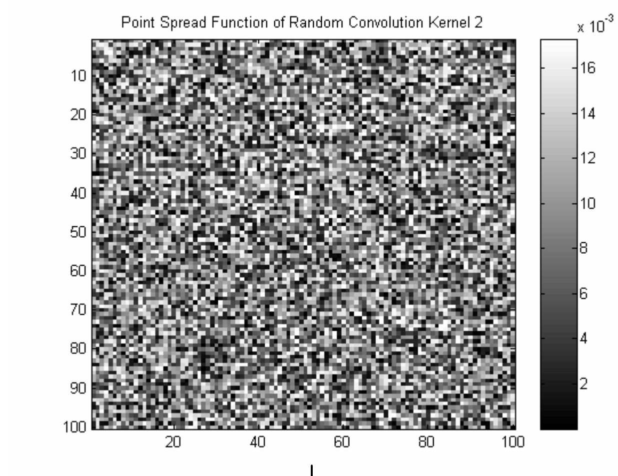
## Random Convolution Kernel 1

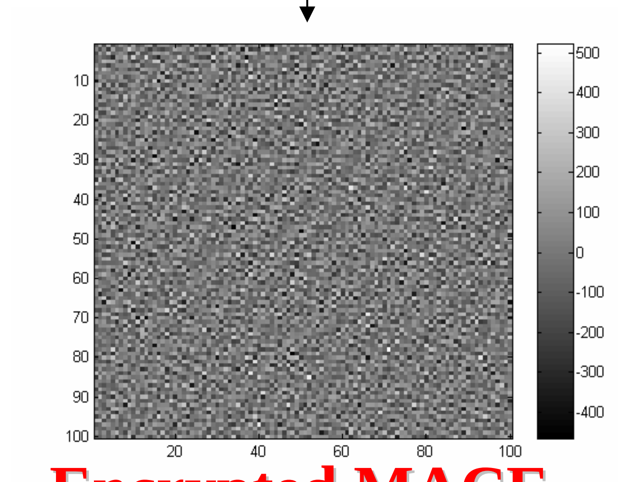


## Encrypted MACE Filter 1

## Random Convolution Kernel 2



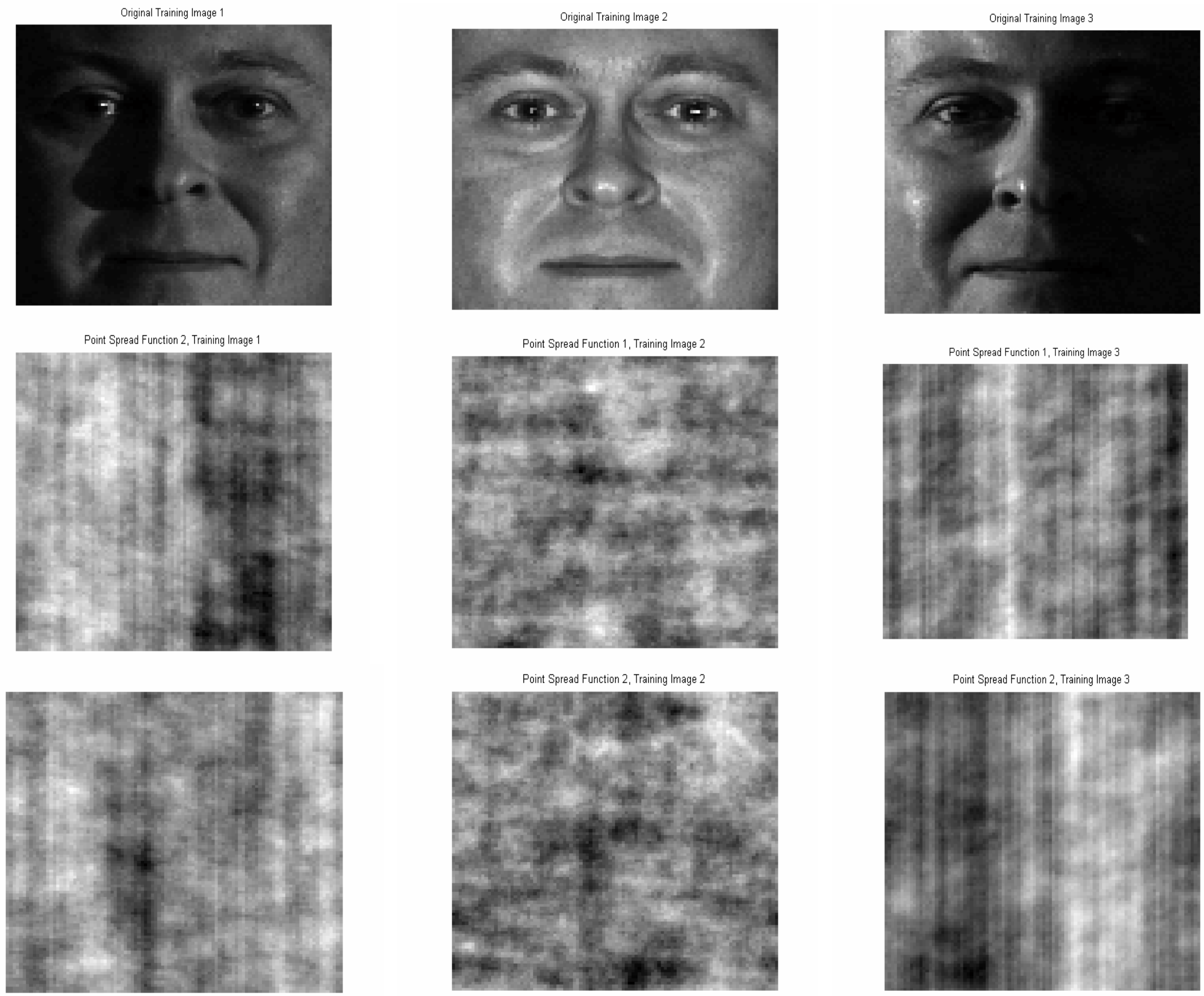
## Encrypted MACE Filter 2

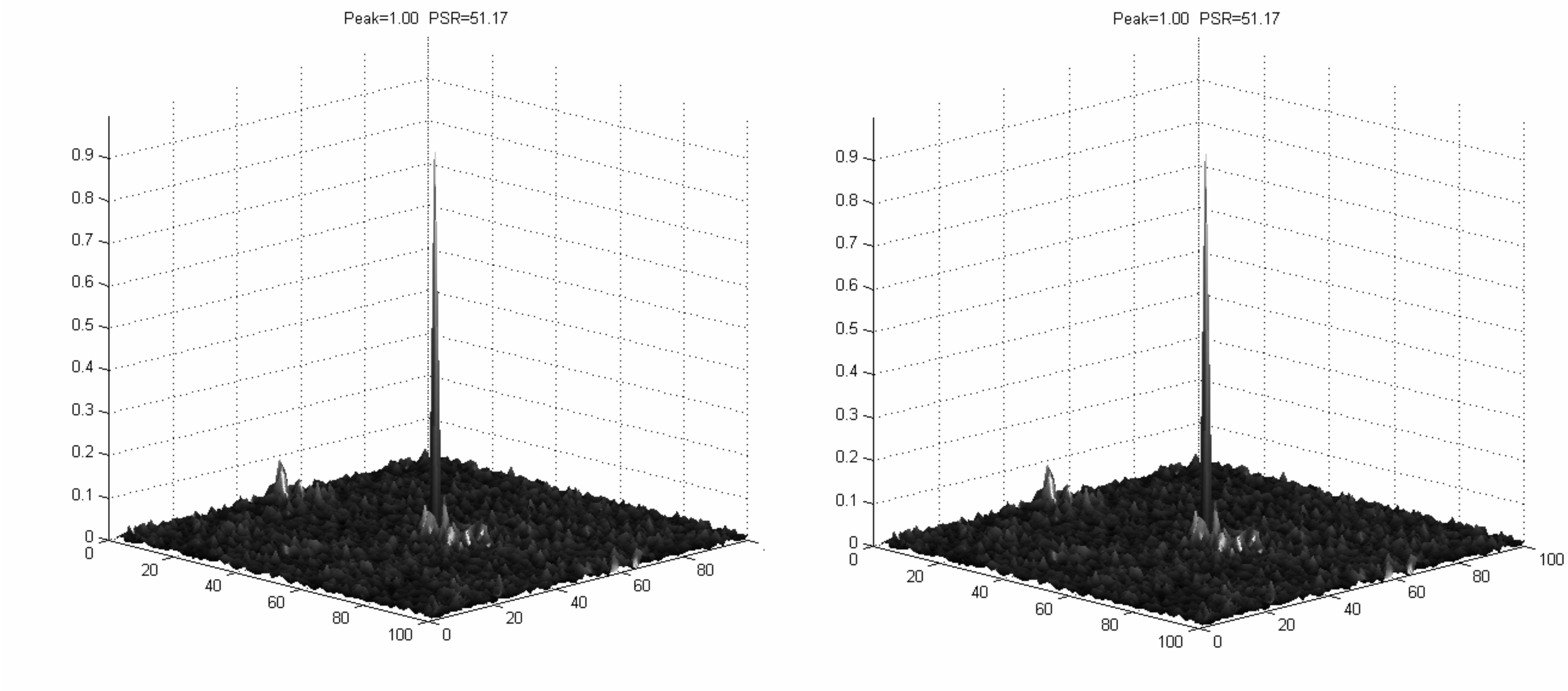
Electrical & Computer ENGINEERING

**Original Training Images**

**Convolved with Random Convolution Kernel 1**

**Convolved with Random Convolution Kernel 2**






**Correlation Output from Encrypted MACE Filter 1**

**Correlation Output from Encrypted MACE Filter 2**