EFFICIENT DESIGN OF ADVANCED CORRELATION FILTERS FOR ROBUST DISTORTION-TOLERANT FACE RECOGNITION

Marios Savvides and B.V.K. Vijaya Kumar

Department of Electrical and Computer Engineering
Carnegie Mellon University, Pittsburgh PA 15213

ABSTRACT

This paper summarizes new research in performing face recognition using advanced correlation filters. We examine the performance of such filters in area of Biometrics for face authentication. We also compare results when applied to perform face identification. Our results are based on the illumination subsets of the CMU PIE database. We also present methods that reduce the memory requirements of these filters to run on limited computational resources including computationally efficient methods of synthesizing these filters. Finally we describe an online training algorithm implemented on a face verification system for synthesizing correlation filters from a video stream to handle pose/scale variations. This system also uses an efficient scheme to perform face localization within the current framework during the authentication stage.

1. INTRODUCTION

Authenticating a user based on their Biometrics is currently an active research field. This paper summarizes new research on using advanced correlation filters for robust face authentication in presence of distortions such as illumination changes and minor pose variations that may be exhibited during the authentication stage (note that throughout this paper we use the terms authentication and verification interchangeably). Related work examining distortion tolerance to facial expressions is presented elsewhere [1][2]. We will also show results demonstrating that these correlation filters can perform illumination tolerant face identification. Similarly face recognition in this context is combining face verification and identification together.

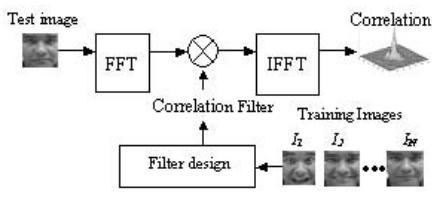

Figure1. Block diagram showing the correlation process.

Advanced correlation filters [3] are of interest as they can model the distortions exhibited in a given set of training images. These advanced filters are very different from the Matched Filter [4], as only a single filter template is synthesized from multiple training images, and more importantly the filter design optimizes a criterion to produce a desired correlation output plane. For example, we typically seek to obtain correlation outputs with sharp correlation peaks in order to simplify detection. The minimum average correlation energy filter [5] is one filter design that synthesizes filters that aim to produce such correlation filter outputs.

In the case of face verification we want only the face images belonging to the authentic class to produce such sharp peaks. We will show in fact that this is the case in our experiments with MACE type filter. Figure 2 shows example correlation outputs resulting from using MACE filter. The left correlation output shows a sharp correlation output resulting from cross-correlating the filter with a test image from the authentic class, and the right image shows a correlation output resulting from a impostor face image. Notice that there is no discernible peak visible in the impostor output. In the next section, we introduce MACE type filter designs in closed form formulations, and we discuss the peak-to-sidelobe metric that we use to quantify the peak sharpness.

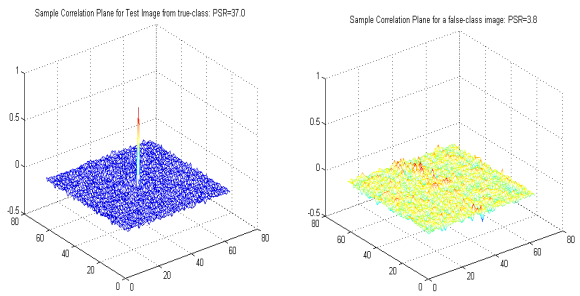

Figure 2. Correlation outputs when using a MACE filter designed for authentic person. *(left)* Correlation output due to face image from the authentic person. *(right)* correlation output when the input is a face image belonging to impostor.

Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'03)
0-7695-1971 3 $17.00 © 2003 **IEEE**

1.2 Minimum Average Correlation Energy Filters

Minimum Average Correlation Energy (MACE) [5] filters are synthesized efficiently in the Fourier domain using closed form equations. This is equation is obtained by optimizing a criterion that seeks to minimize the average correlation energy resulting from cross-correlations with the given training images while satisfying linear constraints to provide a specific value at the origin of the correlation plane for each training image. In achieveing these goals, the resulting correlation outputs from the training images resemble 2D-delta function type outputs, i.e. sharp peaks at the origin with values close to zero elsewhere and the co-ordinates of detected peak indicates the location of the object.

The MACE filter is given in the following closed form equation:

$$\mathbf{h} = \mathbf{D}^{-1}\mathbf{X}(\mathbf{X}^{+}\mathbf{D}^{-1}\mathbf{X})^{-1}\mathbf{u} \quad (1)$$

If we have N training images each of size d1xd2 pixels, then **X** in Eq. (1) is an LxN matrix, where L is the total number of pixels in a single training image (L=d1xd2). **X** is a matrix that contains along its columns lexicographically re-ordered versions of the 2-D Fourier transforms of the N training images. **D** is a diagonal matrix of dimension LxL containing the average power spectrum of the training images along its diagonal. The column vector **u** contains N entries, corresponding to desired values at the origin of the correlation plane of the training images. These constraint values are typically set to 1 for all training images from the authentic class. Another variant of the MACE filter called the unconstrained minimum average correlation energy (UMACE) filter [6], also tries to minimize the average correlation energy but instead of constraining the correlation outputs at the origin to a specific value, it only tries to maximize the peak height at the origin. This optimization leads to the following filter equation:

$$\mathbf{h} = \mathbf{D}^{-1}\mathbf{m} \quad (2)$$

Where **m** is a column vector containing the average of the 2D Fourier transforms of the training images. UMACE filters are computationally more attractive as they require the inversion of only a diagonal matrix.

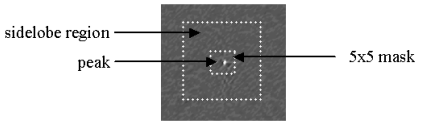

Fig. 3. Peak to Sidelobe Ratio (PSR) computation. A 20x20 region of the correlation output centered at the peak is used to perform the computation

Noise tolerance can be built in to the filters as described in [7]. This is done by substituting **D** with **D'** as follows: **D'**=α**D** + sqrt(1-α^2)**C**, where **C** is a diagonal matrix containing the noise power spectral density. For white noise, **C** is the identity matrix. α ranges from 0 to 1 and is chosen to trade-off noise tolerance for discrimination.

1.2.1 Peak-to-Sidelobe Ratio (PSR)

The Peak-to-Sidelobe Ratio (PSR) is a metric used to measure peak sharpness of the resulting correlation outputs. An authentic test image should yield a large PSR, and impostors very low PSRs. The PSR is computed as follows: first, the test image is cross-correlated with the synthesized MACE filter and the resulting correlation output is searched for the largest value (simply called the peak value). A rectangular region (we use 20x20 pixels) centered at the peak is extracted and used to compute the PSR in the following manner. A 5x5 rectangular region centered at the peak is masked out and the remaining annular region defined as the sidelobe region is used to compute the mean and standard deviation of the sidelobes. The peak-to-sidelobe ratio is then given as follows:

$$PSR = \frac{peak - mean}{\sigma} \quad (3)$$

MACE filters try to maximize peak sharpness, and that is exactly what is captured by the PSR metric. Therefore the larger the PSR the more likely the test image belongs to the authentic class. It is also notable to point out that in computing the authentication decision we use many projections (not just single projection or inner product) which should produce specific response in order for the input to be labeled as belonging to the authentic class, i.e., the peak value should be large, and the neighboring correlation values (which correspond to inner products of the MACE point spread function with shifted versions of the test image) should yield values close to zero. This holds for both types of MACE filters. Another important attribute of the PSR metric is that it is invariant to any uniform scale changes in illumination of the test images.

2 EXPERIMENTS

In our experiments we used 65 people from the two illumination subsets of the PIE database [8] containing 21 images per person. One dataset was captured with the room lights on (which we refer to as PIE-L) and the other database was captured with no ambient background lighting (we refer to this as the PIE-NL). Figure 4. shows 21 images of Person 2 from the PIE-NL dataset (the harder dataset).

In one of the experiments, we synthesized a single filter for each person using the three extreme lighting variations Image 3 (left shadow), Image 7 (frontal lighting), Image 16 (right shadow). We then cross-correlated the whole database with each person's filter and recorded the

computed PSRs. In this experiment we observed that the authentic PSRs were always greater than all impostor PSRs as shown in an example PSR plot for person 2 in Figure 5. In fact there was a clear margin of separation yielding a single threshold that completely discriminates the authentics from the impostors for all 65 people (i.e., 100% verification accuracy was achieved on both datasets).

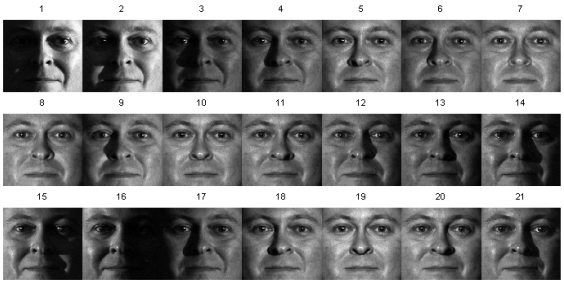

Fig. 4. Sample images of Person 2 from the Illumination subset of PIE database captured with no background lighting.

This can be partly explained by the findings that all possible illuminations lie in a 3-D linear subspace for a lambertian surface model [9]. In this case we are using 3 linearly independent training images that capture the extreme lighting variations, hence clearly any face image that lies in the convex hull of these training images will be perfectly recognized yielding a peak value of 1 and high PSRs near to the training PSRs. In practice, images that fall near this space will also yield high PSR values, and thus be correctly verified.

To examine what happens when we train on near frontal lighting, we repeated the experiment, but used images 7,10,19 (near frontal lighting), and tested on the whole database. We observed a verification accuracy of 93.5 % at False Acceptance Rate (FAR)=0 %. This shows that correlation filters exhibit built-in tolerance to illumination variations, and this can be explained as follows: in order for these filters to produce sharp peaks that resemble 2D delta outputs, they emphasize higher spatial frequency components of the image (which can constitute the relative geometrical structure of the facial features such as eyes, nose and mouth). At the same time, they attenuate lower spatial frequencies, which are mostly affected by lighting changes, therefore achieving tolerance to illumination variations to some degree (as shadows will affect some of the higher spatial frequencies).

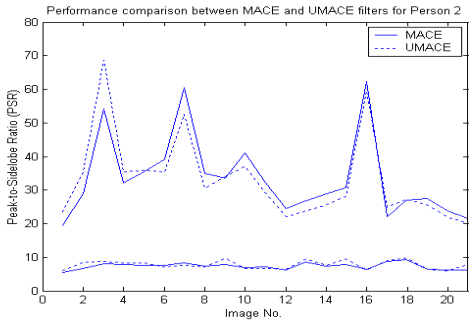

Figure 5. PSR comparison between MACE and UMACE filters using the PIE-NL dataset (with *no*-background lighting) for Person 2. The other people in the database exhibit similar behavior.

We examined the identification performance of MACE type filters. The difference between identification and verification is that in identification we label the class based on the filter that scores a relative maximum PSR, however for verification an authentic test image must achieve a score above a pre-set threshold. Face recognition can be performed by combining the two methods together. Table 1 summarizes some of the results based on using various (near frontal lighting) images and identification performed on unknown illuminations. We observe that the identification accuracy is very high (99.1%-100%). Thus we can conclude that in this case that identification is an easier task than verification, however we still observe that MACE type filters are tolerant to unknown illumination conditions. We compared this to using Individual PCA[1], clearly advanced correlation filters performed much better.

PIE L (FACES IMAGES CAPTURED WITH ROOM LIGHTS ON)							PIE NL (FACES IMAGES CAPTURES WITH ROOM LIGHTS OFF)					
	IPCA		MACE FILTERS		UMACE FILTERS		IPCA		MACE FILTERS		UMACE FILTERS	
TRAINING IMAGES	NO. ERRORS	%ID. RATE	NO. ERRORS	%ID. RATE	NO. ERRORS	%ID RATE	NO. ERROR	%ID. RATE	NO. ERRORS	% ID RATE	NO. ERRORS	% ID. RATE
5,6,7,8,9,10,11,18,19,20	301	78.9%	0	100%	0	100%	526	61.4%	0	100%	0	100%
5,7,9,10	507	64.5%	0	100%	0	100%	832	39.0%	1	99.9%	3	99.7
7,10,19	573	59.9%	0	100%	0	100%	941	31.0%	10	99.1%	10	99.1%
6,7,8	703	50.8%	0	100%	0	100%	832	39.0%	2	99.8%	4	99.7%
18,19,20	471	67.0%	0	100%	0	100%	789	42.2%	2	99.9%	1	99.9%

Table 1. Face identification performance comparison of IPCA, MACE, and UMACE filters using training images captured from near frontal lighting (testing is performed on the whole database) for both PIE illumination datasets

In the verification experiment, we also took all the impostor PSRs resulting from all 65 people (65*64*21=87,360 PSRs), and fit a Gaussian model to model the impostor PSRs distribution. We also repeated this for the authentics (65*21=1365 PSRs). We then examined the probability of error resulting for fixing the PSR verification threshold at various levels. Table 2, summarizes the probability of false acceptance (i.e. probability that an impostor will achieve a PSR greater than the specified threshold) as well as the probability of false rejection (i.e. the probability that an authentic will achieve a PSR lower than the specified threshold).

Using as low threshold as 14, we can achieve a low probability of false acceptance of 7.33×10^{-7} with a probability of false rejection of 2.2×10^{-3}.

Table 2. Probability of False Acceptance and False Rejection at different PSR thresholds

PSR Verification Threshold	Probability of False Acceptance	Probability of False Rejection
7	0.056	4.5×10^{-4}
8	0.002	5.7×10^{-4}
9	0.006	7.2×10^{-4}
10	1.48×10^{-3}	9.1×10^{-4}
11	2.99×10^{-4}	1.1×10^{-3}
12	4.93×10^{-5}	1.4×10^{-3}
13	6.67×10^{-6}	1.7×10^{-3}
14	7.33×10^{-7}	2.2×10^{-3}

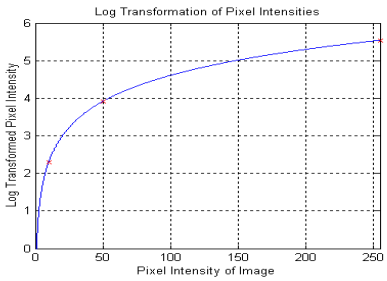

Fig.6 Log transformation of pixel intensities

2.1 Logarithm Transformation for face image enhancement

Lighting variations resulting from the changing position of various light sources cause portions of the face to illuminate, while other portions may be in complete darkness (commonly observed when the face images are captured during night). Figure 6 shows the mapping of pixel intensities ranging from 0 to 255 (assuming 8-bit grayscale images) shown on the horizontal axis to the logarithmic transformed values on the vertical axis. A face image that is partially shadowed will contain pixel intensities from the 10-75 range in intensity belonging to the non-illuminated image region. Taking the log transform will non-linearly map the pixel intensities and as a result the shadowed regions will be intensity enhanced producing a final image with more intelligibility. To evaluate the effects of such transformations, we performed some experiments on the illumination subset of the CMU PIE database [6] that was captured under no background lighting conditions. This dataset exhibits the larger illumination variation, where many images contain substantial shadows. Fig. 7 shows an example image (image 1) selected from Person 2 in the database. The left image shows the original image and the right image is the resulting enhanced log transformed image. We can see that the transformation has increased the intelligibility of the darkened regions, especially in image 1, the left portion of person 2 is originally completely dark due to self shadowing; however the post-processed image shows substantial detail around the eye, mouth that was not previously visible. Table 3 shows a 3.5% improvement in verification performance for one particular set of training images (image no. 7,10,19).

Fig. 7. Person 2 from the illumination subset of CMU PIE database captured with no background lighting. Image 1 on right is the corresponding Log transformed image.

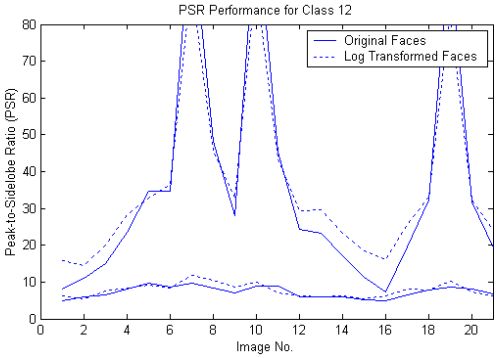

Fig. 8. PSRs comparing the performance of using Log transformed face images with MACE filters. The solid line represents the PSRs using the original database images and the dashed lines refer to using log transformed face images. The top plots belong to the authentic person and the bottom plots are the maximum impostor PSR. For images yielding low PSRs for the authentic person (<30), the Log transformed face images provide an improvement in the margin of separation between impostor and authentic PSRs.

Table 3. Verification accuracy at zero False Acceptance Rate (FAR=0).

Images used	MACE	UMACE
Original	93.50 %	93.59 %
Log Transformed	97.18 %	97.09 %

3 QUAD PHASE MACE FILTERS

In 2D images, phase information in the Fourier domain is more important than the magnitude information for performing image reconstruction as described in [10] Since phase contains most of the intelligibility of an image, and can be used to retrieve the magnitude information, we can reduce the memory storage requirement of MACE filters by retaining only the phase of the filter and quantizing it 4 levels, hence the name Quad-Phase MACE filter. Each element in the filter array will take on ±1 for the real component and ±*j* for the imaginary component in the following manner.

$$H_{QP-MACE}(u,v) = \begin{cases} +1 & \Re\{H_{MACE}(u,v)\} \geq 0 \\ -1 & \Re\{H_{MACE}(u,v)\} < 0 \\ +j & \Im\{H_{MACE}(u,v)\} \geq 0 \\ -j & \Im\{H_{MACE}(u,v)\} < 0 \end{cases} \quad (4)$$

Essentially 2 bits/frequency are needed to encode the 4 phase levels, namely π/4, 3π/4, 5π/4, 7π/4. Tutorial on partial information filters can be found here [11].

3.1 Four-level correlator

The QP-MACE filter described in Eq. (4) has unit magnitude at all frequencies, encoding 4 phase levels. When multiplying the QP-MACE with the conjugate of the Fourier transform of the test image the phases should cancel out in order to provide a large peak. The only way the phases will cancel out is if the Fourier transform of the test image is also phase quantized in the same way such that phases can cancel out to produce a large peak at the origin. Therefore in this architecture, we also quantize the Fourier transform of the test images as in Eq. (4). This yields correlation outputs that are similar to the full complexity MACE filters as shown in Figs. 9 and 10. In fact we are still able to achieve 100% verification performance when using the 3,7,16 index training images to synthesize QP-MACE filters.

The described Quad Phase MACE filters only need 2 bits/pixel in the frequency domain achieving a compression ratio up to 32:1 making them very attractive for implementation on platforms with limited computational and memory resources.

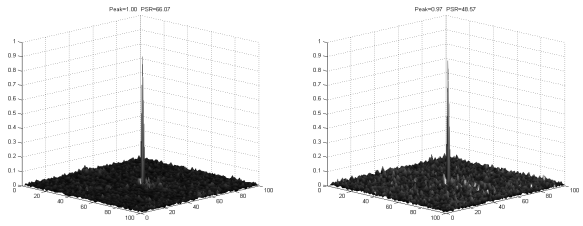

Fig. 9. Correlation Outputs: (Left) full phase MACE filter (Peak =1.00, PSR=66) (right) Quad Phase MACE Filter using Four-level correlator (Peak=0.97, PSR=48)

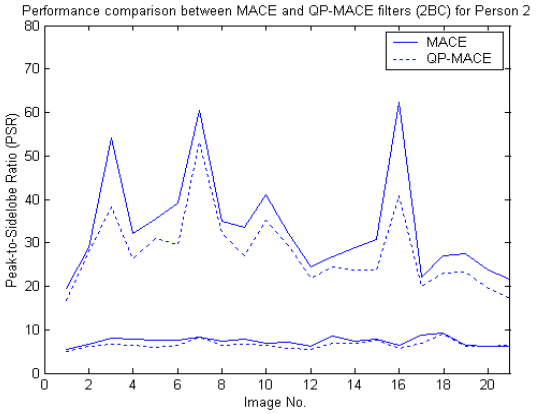

Fig. 10. PSR plot for Person 2 comparing the performance of full-complexity MACE filters and the reduced-complexity Quad Phase MACE filter using the Four-level correlator on the *harder* illumination dataset (PIE-NL) that was captured with background lights off. The top plots represent the PSRs for authentics and the bottom plots are PSRs from the impostors.

4. EFFICIENT METHODS TO SYNTHESIZE MACE FILTERS

There are many application scenarios where training and recognition is to be performed with limited computational resources. This section concentrates of synthesizing constrained MACE filters efficiently by showing computationally efficient methods of computing the inverse of $(\mathbf{X}^{+}\mathbf{D}^{-1}\mathbf{X})^{-1}$ needed in the MACE formulation. Examining the MACE filter in Eq. (1). we can re-write it in the following form.

$$\begin{aligned} \mathbf{h} &= \mathbf{D}^{-0.5}\mathbf{D}^{-0.5}\mathbf{X}(\mathbf{X}^{+}\mathbf{D}^{-0.5}\mathbf{D}^{-0.5}\mathbf{X})^{-1}\mathbf{u} \\ &= \mathbf{D}^{-0.5}(\mathbf{D}^{-0.5}\mathbf{X})((\mathbf{D}^{-0.5}\mathbf{X})^{+}(\mathbf{D}^{-0.5}\mathbf{X}))^{-1}\mathbf{u} \\ &= \mathbf{D}^{-0.5}\mathbf{X}'(\mathbf{X}'^{+}\mathbf{X}')^{-1}\mathbf{u} \end{aligned} \quad (5)$$

where

$$\mathbf{X}' = \mathbf{D}^{-0.5}\mathbf{X} \quad (6)$$

Thus we can see that **X'** is nothing but the original Fourier transformed training images X pre-whitened by the average power spectrum D.

Writing in this format allows us to form an alternative way of incrementally computing the inverse of the inner-product matrix (also commonly referred to known as the Gram matrix) $(\mathbf{X}'^{+}\mathbf{X}')$ [12]

$$(\mathbf{X}'_t{}^{\mathbf{T}}\mathbf{X}'_t)^{-1}=\begin{bmatrix}(\mathbf{X}'_{t-1}{}^{\mathbf{T}}\mathbf{X}'_{t-1})^{-1} & \mathbf{0}^{\mathbf{T}}\\ \mathbf{0}^{\mathbf{T}} & 0\end{bmatrix}+\ldots \quad (7)$$

$$\ldots+k_t^{-1}\begin{bmatrix}(\mathbf{X}'_{t-1}{}^{\mathbf{T}}\mathbf{X}'_{t-1})^{-1}\mathbf{X}'_{t-1}{}^{\mathbf{T}}\mathbf{x}_t\\ 1\end{bmatrix}\begin{bmatrix}-(\mathbf{X}'_{t-1}{}^{\mathbf{T}}\mathbf{x}_t)^{\mathbf{T}}(\mathbf{X}'_{t-1}{}^{\mathbf{T}}\mathbf{X}'_{t-1})^{-1} & 1\end{bmatrix}$$

where the scalar constant k_t is defined as

$$k_t=\mathbf{x}_t{}^{\mathbf{T}}\mathbf{x}_t-(\mathbf{X}'_{t-1}{}^{\mathbf{T}}\mathbf{x}_t)^{\mathbf{T}}(\mathbf{X}'_{t-1}{}^{\mathbf{T}}\mathbf{X}'_{t-1})^{-1}(\mathbf{X}'_{t-1}{}^{\mathbf{T}}\mathbf{x}_t) \quad (8)$$

This is can be computationally simpler to compute than the direct inverse for large number of training images. The only other constraint that must be satisfied in order to compute the matrix inverse, is that the training image at time instant t must not be a linear combination of any previous training images as this would result in $\mathbf{X}'_t{}^{\mathbf{T}}\mathbf{X}'_t$ being singular. Therefore we have to test whether the determinant of $\mathbf{X}'_t{}^{\mathbf{T}}\mathbf{X}'_t$ is non-zero with the addition of each new training image. Note that the pre-whitening step does not affect the linear independence of the column space, hence we can equivalently test the determinant of $\mathbf{X}_t{}^{\mathbf{T}}\mathbf{X}_t$ if desired.

Computing the determinant using standard techniques is very expensive, hence we summarize a more efficient way to test for linear independence. The gram matrix $\mathbf{X}'_t{}^{\mathbf{T}}\mathbf{X}'_t$ has a special structure that can be exploited to formulate an efficient iterative method to compute the determinant as new images are collected.

$$\mathbf{X}'_t{}^{\mathbf{T}}\mathbf{X}'_t=\begin{bmatrix}\mathbf{X}'_{t-1}{}^{\mathbf{T}}\\ \mathbf{x}'_t{}^{\mathbf{T}}\end{bmatrix}\begin{bmatrix}\mathbf{X}'_{t-1}{}^{\mathbf{T}} & \mathbf{x}'_t{}^{\mathbf{T}}\end{bmatrix}=\begin{bmatrix}\mathbf{X}'_{t-1}{}^{\mathbf{T}}\mathbf{X}'_{t-1} & \mathbf{X}'_{t-1}{}^{\mathbf{T}}\mathbf{x}'_t\\ (\mathbf{X}'_{t-1}{}^{\mathbf{T}}\mathbf{x}'_t)^{\mathbf{T}} & \mathbf{x}'_t{}^{\mathbf{T}}\mathbf{x}'_t\end{bmatrix}$$

It is important to note that the constant term k_t in equation (8) is known as the Schur complement of the partitioned Gram matrix $\mathbf{X}'_t{}^{\mathbf{T}}\mathbf{X}'_t$ shown above. Assuming that $\mathbf{X}'_{t-1}$ has linearly independent columns then the linear combination vector $\mathbf{e}_t$ will be zero vector only if the image $\mathbf{x}_t$ is a linear combination of the other training images.

$$\mathbf{e}_t=[\mathbf{X}'_{t-1}\,\mathbf{x}_t\;]\mathbf{a}_t \quad (9)$$

The norm squared of the error $\mathbf{e}_t$ is $(\mathbf{e}_t{}^{\mathbf{T}}\mathbf{e}_t)$ which is also known as the Schur complement of $\mathbf{X}'_t{}^{\mathbf{T}}\mathbf{X}'_t$ can be computed as shown as k_t in Eqt (8).

It can be shown in [12] that the determinant of the Gram matrix can be iteratively computed as follows:

$$\det(\mathbf{X}'_t{}^{\mathbf{T}}\mathbf{X}'_t)=k_t\det(\mathbf{X}'_{t-1}{}^{\mathbf{T}}\mathbf{X}'_{t-1})=\prod_{i=1}^{t}k_i \quad (10)$$

If $\det(\mathbf{X}'_{t-1}{}^{\mathbf{T}}\mathbf{X}'_{t-1})$ is non-zero, then $\det(\mathbf{X}'_{t-1}{}^{\mathbf{T}}\mathbf{X}'_{t-1})$ is nonzero only if k_t is non-zero. Therefore we need only compute and test if the Schur complement of the augmented Gram matrix $\mathbf{X}'_t{}^{\mathbf{T}}\mathbf{X}'_t$ is non-zero with the addition of each new training image. This is a computationally more efficient way to test for linear independence given a new training image.

4.1 Incremental Updating of UMACE Filters

UMACE filters have simpler implementation structure that can be exploited to produce a memory efficient incremental synthesis procedure, thus alleviating the need to store all the training images for synthesize.

From Eq. (2) we note that since **D** is a diagonal matrix, the elements of the mean image in the frequency domain are divided by the elements of **D** along its diagonal, therefore we essentially do not need to divide by the number of training images to form the mean image and the average power spectrum as the scalar divisor cancels out as follows:

$$\mathbf{h}_{\text{UMACE}}=\left[\frac{1}{N}\sum_{i=1}^{N}(\mathbf{X}''_i\mathbf{X}''^{*}_i)\right]^{-1}\frac{1}{N}\sum_{i=1}^{N}(\mathbf{x}_i\mathbf{x}_i^{*})=\mathbf{D}'^{-1}\mathbf{m}' \quad (11)$$

thus given a new image x_t, we can incrementally update as follows.

$$\mathbf{D}'_t=\mathbf{D}'_{t-1}+\mathbf{X}''_t\mathbf{X}''^{*}_t \quad (12)$$

$$\mathbf{m}'_t=\mathbf{m}'_{t-1}+\mathbf{x}_t \quad (13)$$

where $\mathbf{X}_t$**"**is a diagonal matrix containing the Fourier transform of the training image at time step t, lexicographically re-ordered and placed along the diagonal. Therefore, for incrementally synthesizing a Quad Phase UMACE filter, we can simplify the update process further showing that we only need to store and update the mean image or simply m_t' since the power spectrum is positive it will not affect the sign of the elements in the UMACE filter in Eq. (11), therefore the same QP-UMACE filters are formed.

5. ONLINE TRAINING SCHEME FOR SYNTHESIZING DISTORTION TOLERANT CORRELATION FILTERS

The performance of any recognition system depends heavily on the choice of training images. This section we propose an online-training algorithm for synthesizing correlation filters. We describe this in the context of enrolling a person in a face authentication system, but this scheme is general for any other applications.

The advanced correlation filters we described are attractive since they allow us to synthesize a single filter template from multiple training images. A question that can arise what is the maximum limit of training images that can be used to synthesize a single filter? Currently there is no clear way to quantify this, as the number of training images depends on the amount of variability present in these training images. Remembering that MACE filters perform a correlation energy minimization, it is logical to expect that training images with a larger range of variability will synthesize filters that are not able minimize the average correlation energy in comparison to training sets with smaller variations. Thus the resulting PSRs are smaller for the filter synthesized from a larger variation training set, this includes the PSRs obtained even from the training images used to synthesize the filter.

Therefore one possible way to quantify the quality of a filter is to measure the PSRs resulting from each of the training images that were used in synthesis. If these PSRs are smaller than some threshold ϑ_1 then this filter is not capable to perform well, as we expect the PSRs resulting due to test images to be less than that of the training images. If the training image PSRs are low to begin with, then clearly the filter will not be able to provide sufficient discrimination.

During the enrollment process, we have collected a video stream of face images, and the assumption we can make is that the difference in images between successive frames (assuming a reasonable capture frame rate) is not great. Thus we can build a filter from a couple of images, then cross-correlate the next frame with the synthesized filter. If the computed PSR is smaller than some threshold ϑ_2 then that means that the current face image is not represented by the synthesized filters, so we include it in the current training set and re-synthesize or update the current filter. After each image we add to the training set, we test the quality of the updated filter as described before. If the filter has reached its maximum capacity we start building a new filter, using the same scheme. The flowchart in Fig. 11, describes this process.

While we have shown that these filters are tolerant to illumination changes, handling pose changes is a tougher problem (we are currently working on evaluating the performance with pose variations on the PIE pose dataset). These advanced correlation filters in efforts to produce sharp correlation peaks, emphasize higher spatial frequencies in the images, therefore capture the relative geometrical structure of the facial features while ignoring the lower spatial frequencies, which are mostly affected by illumination conditions. Thus expected poses of the face images, including scale changes need to be shown as examples to these filters in order to correctly classify them during verification. This is done using an instructive guide in the system, asking the user to exhibit various pose changes as the online-training algorithm is running. The final enrollment process period is then dependent on how many filters can be stored, and how much variation is exhibited by the user.

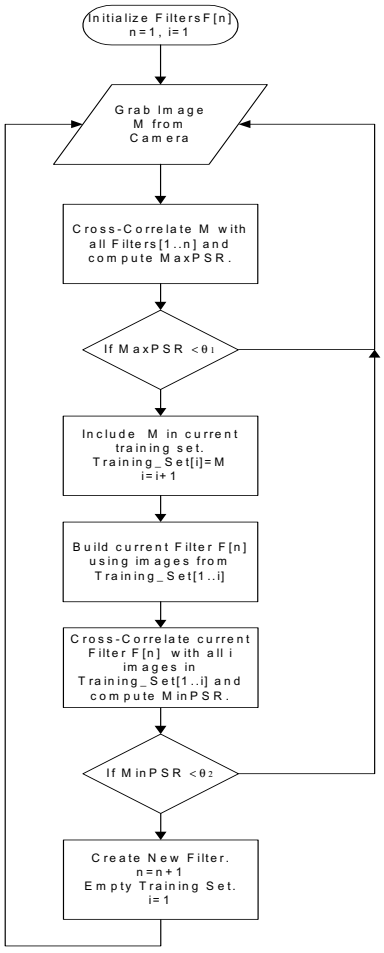

Fig. 11 Flow diagram of the online-training algorithm for synthesizing correlation filters.

6. FACE LOCALIZATION

In the authentication process, the user is asked to co-operate and place his face in front of the camera. To not constrain the user and for purposes of increasing the speed of the overall verification process, we have

implemented a face localizer which locates the face and centers it for the classification. While correlation-filters are shift invariant, we still need to provide the full face image for reliable verification. For verification purposes, correlation filters need to have a near full face image in order to perform well and achieve a PSR score above a specific threshold, this is harder to achieve especially when other distortions such as pose are present, therefore, we have added a simple pre-processing step to locate partial face in view of the camera, in order to automatically capture and process a full face image. When a face image is captured it is correlated with the stored set of correlation filters, and the maximum PSR achieved is stored along with the location of the correlation peak. This peak location tells us how much the face image is shifted, thus we can use this to select and crop out the correct face region. The capture image resolution is typically much higher than the resolution of the face images used for verification. In our system, we crop out a region of the captured scene, and downsample the image region to the resolution desired for performing face verification. There are two reasons for doing this: the downsampling process of the selected face region in scene, allows us to smooth out camera noise by form of pixel averaging (in comparison to directly capturing a very low resolution image). Also, more importantly, we can locate the position of the face in the smaller resolution image, and estimate the correct face region in the high resolution background image, shift the crop window, and downsample the estimated region containing the face, then perform verification. This is more computationally efficient than performing cross-correlation of a face template on a higher resolution background image to first locate the face, then downsample and perform verification. Fig. 12, shows example face captured on the left (the white cross denotes the position of correlation peak). The right image shows the face centered based on the peak location. Face recognition can be performed by simply enrolling many people in the verification system.

Fig. 12. (left) Captured face image with face location marked with the white cross. (right) Full face image captured based on estimated face location.

7. CONCLUSION

In this paper we have summarized recent research work on using correlation filters for face recognition. We have shown results demonstrating that these filters are tolerant to illumination changes (we are currently working on evaluating performance due to pose on the PIE pose subsets). However, in certain verification type scenarios, one can assume that the user will be co-operative and provide a suitable pose for verification. We have presented an online training algorithm that has been able to handle pose variations in such situations.

We have also shown methods to reduce the complexity (memory requirements to only 2bits/pixel in frequency domain) of these filters showing that verification performance is similar to the full complexity filters. This includes computationally efficient methods of synthesizing MACE filters on limited resource platforms. Finally we describe an online training algorithm implemented on a face verification system for synthesizing correlation filters to handle pose/scale variations as well a way to perform efficient face localization.

ACKNOWLEDGEMENTS

This research is supported in part by SONY Corporation.

7. REFERENCES

[1]. M. Savvides, B.V.K. Vijaya Kumar and P.K. Khosla, "Face verification using correlation filters," *Proc. Of Third IEEE Automatic Identification Advanced Technologie*s, Tarrytown, NY, 56-61 (2002)

[2]. B.V.K. Vijaya Kumar, M. Savvides, K. Venkataramani and C. Xie, "Spatial Frequency Domain Image Processing for Biometric Recognition," *Proc. Of Intl. Conf. on Image Processing* (ICIP), Rochester, NY, Sept. 2002.

[3]. B.V.K. Vijaya Kumar, "Tutorial survey of composite filter designs for optical correlators," Applied Optics 31 (1992)

[4]. A. Vanderlugt, "Signal detection by complex spatial filtering," *IEEE Tans. Inf. Theory* 10, 139-145 (1964).

[5]. A. Mahalanobis, B.V.K. Vijaya Kumar, and D. Casasent, "Minimum average correlation energy filters," *Appl. Opt*. 26, 3633-3630 (1987).

[6]. A. Mahalanobis, B.V.K. Vijaya Kumar, S.R.F. Sims and J.F. Epperson, "Unconstrained correlation filters," *Appl. Opt.*, Vol.33, pp. 3751-3759 (1994).

[7]. P. Refregier, "Filter Design for optical pattern recognition: Multi-criteria optimization approach", *Optics Letters*, Vol. 15, 854-856, 1990.

[8]. T. Sim, S. Baker, and M. Bsat "The CMU Pose, Illumination, and Expression (PIE) Database of Human Faces," Tech. Report CMU-RI-TR-01-02, Robotics Institute, Carnegie Mellon University, January (2001).

[9]. P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces vs Fisherfaces: Recognition Using Class Specific Linear Projection," PAMI-19(7), (1997).

[10]. A.V. Oppenheim and J. W. Lim, "The importance of phase on signals," *Proc. IEEE* 69, 529-532 (1981).

[11]. B. V. K. Vijaya Kumar, "A Tutorial Review of Partial-Information Filter Designs for Optical Correlators," *Asia-Pacific Engineering Journal (A)*, Vol. 2, No. 2 (1992) 203-215

[12]. Louis L. Scharf, "*Statistical Signal Processing-Detection, Estimation, and Time-Series Analysis*", Addison-Wesley Publishing Company, 1991.