A ROBUST FEATURE-BASED TECHNIQUE FOR WATERMARKING FRONTAL FACE IMAGES

Athanasios Nikolaidis *Ioannis Pitas*

Department of Informatics
Aristotle University of Thessaloniki
Box 451, Thessaloniki 540 06, GREECE
e-mail: {nikola, pitas}@zeus.csd.auth.gr

ABSTRACT

We introduce a novel method for embedding and detecting a chaotic watermark in the digital spatial domain of color face images, based on localizing salient facial features. Simulation results prove the robustness of the method to several kinds of attack, such as compression, filtering, scaling, cropping and rotation.

1. INTRODUCTION

A field of rapidly increasing interest during the last few years has been multimedia protection. This has been addressed by the fact that many innovative techniques for digital data transfer, storage and processing have emerged. This has facilitated malicious users with the opportunity of manipulating images, audio or video and, thus, claiming product ownership.

Many different watermarking methods for still images have been proposed, that embed a pseudo-random sequence in either the spatial or the transform image domain [1]-[4]. All of these methods consider the image as a single object in the spatial domain and embed the watermark across the entire image content without exploiting possible local salient properties. Usually, the methods are customized in such a way that no care is taken whether the watermark is resistant against all types of attacks.

The current paper presents a technique for copyright protection by watermarking a certain image class that has special interest because of the existence of large databases of such content, namely color frontal face images with uniform background, based on extracting a finite set of salient facial features. The proposed technique will be proved to provide sufficient immunity to any intended attack. Section 2 presents the color segmentation and region approximation technique, as well as the pattern matching process for localizing the salient features. In Section 3, the general class of chaotic watermarks is presented together with adaptations for digital images. Section 4 explains the connection between the extracted features and the watermark to be embedded on the image, and Section 5 presents the watermark detection procedure. Simulation results for several kinds of manipulations on the watermarked image are presented and explained in Section 6 and, finally, conclusions are drawn in Section 7.

2. FACE SEGMENTATION AND SALIENT FEATURE LOCALIZATION

The first stage in the process of embedding our watermark to a selected region of the image, is to segment the face region so that the search for salient features, namely the eyes and the mouth, is limited in this area.

In the following we propose one technique for eyes and mouth localization that is rotation, translation and scale invariant. Other techniques for this purpose have been proposed in the literature [5]-[6]. The method followed is based on exploiting color information in a similar way as in [5]. Our aim is to discriminate the skin-colored image region. The original RGB image is thus converted to its HSV representation because it is easier to perform color segmentation in this color space.

The choice of certain ranges for hue, saturation and value parameters ensures that the segmented region of interest will approximately be the same even after some manipulation.

A connected components algorithm follows thresholding in order to isolate all the compact skin-colored regions. The downsampled map is scaled to the original size of the image. In order to get a good approximation of the face region that does not contain useless areas, e.g. the neck, as well as to prevent the face region areas from getting connected to the background, we employ an $\alpha-$trimmed Mean Radial Basis Function network to get an ellipsoidal approximation of the region [7]. This technique provides the marginal median estimation for the center of each object:

$$\hat{\mu}_k = \frac{\sum_{i=\alpha_k N_k}^{N_k-\alpha_k N_k} X_{(i)}}{N_k - 2\alpha_k N_k} \quad (1)$$

where $X_{(i)}$ are the marginal data samples sorted according to their values, N_k is the total number of data samples assigned to the k-th hidden unit, and α_k is the percentage of data samples to be trimmed away. The estimate of the covariance matrix is:

$$\hat{\mathbf{\Sigma}}_k = \frac{\sum_{i=0}^{N_k-\alpha_{k,\mathcal{M}} N_k} (\mathbf{X}_{(i),\mathcal{M}} - \hat{\mu}_k)(\mathbf{X}_{(i),\mathcal{M}} - \hat{\mu}_k)^T}{N_k - \alpha_{k,\mathcal{M}} N_k} \quad (2)$$

where $\mathbf{X}_{(i),\mathcal{M}}$ is the i-th ordered data sample according to the Mahalanobis distance, and $\alpha_{k,\mathcal{M}}$ is the trimming percentage in this case.

Once the face region is segmented, the trimmed ellipsoidal approximation is known and thus its orientation [8] can be computed. The input image should be rotated according to this angle before pattern matching.

The most prominent features contained in the ellipsoidal area are the eyes and the mouth and can be approximated sufficiently well by proper geometric functions. These features are unique in such images and act as robust reference points even after some geometric distortion. Other similar approaches use 2-D sinc functions for eye modeling [6]. The eye can be regarded as a circle of low, almost constant intensity centered inside an ellipse of very bright intensity. Ideally, the eye detector is described by:

$$\int_{\mathcal{C}_{per}} w_c(x,y)I(x,y) = \int_{\mathcal{E}_{per}} w_e(x,y)I(x,y) - \int_{\mathcal{C}_{per}} w_e(x,y)I(x,y) \quad (3)$$

where $I(x,y)$ is the image intensity, $\mathcal{C}_{per}$, $\mathcal{E}_{per}$ are the sets of pixels lying on the perimeter of the circle and the ellipse, respectively, and $w_c(x,y)$ and $w_e(x,y)$ are weighting functions that compensate for the luminance differences between the two areas. The weighting functions cannot be easily estimated. However, constant values can be incorporated without significant loss of accuracy in the estimation of the feature position. In order to define a pattern matching criterion for eye detection, we discretize (3), use constant weighting functions and search for the absolute minimal response of the difference within the facial region:

$$R_{eye}(x,y) = \left| \sum_{(i,j)\in\mathcal{C}} w_c I(i,j) - \sum_{(i,j)\in\mathcal{E}-\mathcal{C}} w_e I(i,j) \right| \quad (4)$$

where $\mathcal{C}$ and $\mathcal{E}$ are the sets of points that belong to the circle and the ellipse, respectively.

To obtain a reasonable estimate of the relation between the magnitudes of the circle and ellipse axes and the weighting constants, we have to compute the integrals in (3). For simplicity, we assume that the intensity is represented by its mean value I_c in the circle area, and its mean value I_e inside the ellipse area but outside the circle area.

The search for potential left and right eye positions is performed over the upper left and upper right quarter of the rotated facial region that is covered by the ellipsoidal area, respectively. The correct eye position is the one for which the matching response $R_{eye}(x,y)$ is minimal.

A similar pattern matching technique is used for the localization of the mouth, except that the model now consists of two concentric ellipses having major semiaxes of the same magnitude and minor semiaxes of considerably different magnitudes. This pattern is again unique for the mouth, and the search is performed in the lower half of the ellipsoidal region.

3. WATERMARK CONSTRUCTION

In the previous section we developed a method of locating salient features, so that they can be used as reference points to embed our watermark. We should now define the class of watermarks that will be embedded in the spatial image domain. We choose to construct a watermark based on a chaotic trajectory [9] because of its controlled lowpass properties compared to a usual pseudorandom sequence. The first step to construct such a watermark, is to produce a sequence of real numbers by using a mapping function $\mathbf{F} : U \to U, U \subset I\!R$ of the form:

$$z(n+1) = \mathbf{F}(z(n), \lambda), \qquad z(n) \in U, \lambda \in I\!R \quad (5)$$

where $n = 0, 1, 2, ...$ denotes the current iteration and λ is a parameter that controls the chaotic behavior of the system. The number of iterations is arbitrary and can be adapted to our needs. The system theoretically produces trajectories of an infinite period. By changing the value of the parameter λ, the set of real numbers is divided in two subsets. The decision on whether the trajectory presents regular or chaotic behavior depends on the seed value $z(0)$. If $z(0) \in U_{reg}$, the produced sequence proves to be periodic, whereas if $z(0) \in U_{ch}$ it is chaotic. The values of the produced trajectory oscillate inside an interval $[z_{min}, z_{max}] \subset U$ that is related to λ. Thus, we can define a threshold level $z_{th} \in [z_{min}, z_{max}]$ in a way that, after thresholding the sequence numbers, a bipolar sequence $s(n) \in \{-1, 1\}$ is produced with approximately equal number of -1s and 1s. Parameter λ controls the frequency characteristics of the chaotic sequence, i.e. the frequency of the transitions $-1 \to 1$ and $1 \to -1$. We used a map that is based on the Rényi transformation [10] in our implementation. For $\lambda > 1$ and values close to 1, we get a chaotic watermark with low number of transitions and, thus, lowpass properties, whereas when $\lambda \simeq 2$ the transitions are very frequent, the lowpass properties degrade and the sequence is very similar to a pseudorandom one.

The sequence we produced so far is one-dimensional. To embed it in a two-dimensional signal, such as a digital image, we need to scan across the sequence in such a way that the lowpass properties are preserved. The classic raster scan is not proper for this task because the number of transitions is not any more under control in the vertical dimension. To avoid this we use Peano scan order which has better spatial properties than the raster scan. In addition, it is possible to use cellular smoothing to eliminate spontaneous transitions that emerged after the Peano scan [9]. Using this technique, the output watermark has local neighborhoods of 1s (or -1s) that are more compact.

In order to construct different watermarks we use a key K that produces the seed value $z(0)$ for the generation of a chaotic trajectory. Keys of slightly different value provide sufficiently uncorrelated trajectories, because the set **K** of possible keys is quite large. This reduces the possibility of the watermark being tampered. This also ensures non-invertibility of the watermark. Thus, the corresponding key cannot be extracted from the 2D watermark.

4. WATERMARK EMBEDDING

In this stage we make use of the extracted salient feature set and ellipsoidal region orientation, to embed the produced watermark in a specific image region that will be easy to detect even after intentional or unintentional attacks.

The prototype watermark of size $2^n \times 2^n$ is first scaled to the size of the facial area where it is going to be embedded. If A_{em} is the embedding area, its size $K_1 \times K_2$ is defined by:

$$K_1 = k(x_{(mo)} - \bar{x}_{(ey)}),\ K_2 = l(y_{(r\_ey)} - y_{(l\_ey)}) \quad (6)$$

where $x_{(\cdot)}$ and $y_{(\cdot)}$ are feature coordinates, $\bar{x}_{(\cdot)}$ are mean feature coordinates, and k, l are normalizing factors that control the size of A_{em} so that it covers at least the entire face region. Afterwards,

we rotate the prototype watermark by the angle θ of orientation that was computed in Section 2.

The scaled watermark is centered in the mass center of the feature points set $\mathcal{F} = \{F_i, i = 1, ..., M\}$:

$$(\bar{x}, \bar{y}) = \left(\frac{1}{M} \sum_{(x,y) \in \mathcal{F}} x, \frac{1}{M} \sum_{(x,y) \in \mathcal{F}} y \right) \quad (7)$$

In our case $M = 3$. The mass center is also the center of the A_{em} region. After centering the watermark in the proper image area, it is rotated by $-\theta$ with respect to the center of the image. It then covers a new area A_r. Before superimposing it on the original image, a visual masking stage is introduced. In this stage the variance is computed for every point of the original image $f(x, y)$ over a proper neighborhood $N \times N$:

$$Var_{x,y} = \frac{1}{4N(N+1)} \cdot \sum_{i=-N}^{N} \sum_{j=-N}^{N} (f(x+i, y+j) - \mu_{x,y})^2 \quad (8)$$

where $\mu_{x,y}$ is the mean value over the same neighborhood. The local variance is then normalized according to its maximum value, and is compared against a threshold T_v. If the variance exceeds this threshold, then the local neighborhood contains a big amount of texture or edge information, and the watermark can be embedded without being visually perceptible. Otherwise the region is considered to be close to uniform, like background, and is not suitable for watermark casting. The dependence of the variance threshold on the watermark power is such that if the watermark power is increased, the threshold is also increased non-linearly, so that the watermark remains imperceptible.

If w_0 is the prototype watermark, then the scaled and rotated watermark w_n of size $K_1 \times K_2$ is casted to the region A_r. The watermarked image $f_w(x, y)$ is defined as:

$$f_w(x, y) = f(x, y) + h(x, y) \cdot w_n(x, y) \quad (9)$$

where $h(x, y)$ is the watermark power that is a function of the local variance:

$$h(x, y) = h_{max} \cdot s(Var(x, y)) \quad (10)$$

where $s()$ takes values in the range $[0, 1]$. s is chosen to increase monotonically with the variance. In our case, the watermark is casted in the spatial domain and, thus, the watermark power is quantized to two integer values, 0 and h_{max}, depending on the value of the variance. Therefore, the function $s()$ that is employed is:

$$s(x) = \begin{cases} 1, & x > T_v \\ 0, & x \leq T_v \end{cases} \quad (11)$$

5. WATERMARK DETECTION

When a prototype watermark is to be detected inside a watermarked and possibly manipulated image, the image has to be first segmented, so that the feature set and orientation of the approximated face region are derived. The prototype watermark is again scaled, centered, and rotated according to the information obtained from the segmentation and feature extraction stage. The response of the correlation between the geometrically adapted watermark and the detection region A_{det} is given by:

$$R(\hat{f}_w, \hat{w}_n) = \frac{1}{N_{\mathbf{A}}} \sum_{(x,y) \in \mathbf{A}} \hat{f}_w(x, y) - \frac{1}{N_{\mathbf{B}}} \sum_{(x,y) \in \mathbf{B}} \hat{f}_w(x, y) \quad (12)$$

where $\mathbf{A} = \{(x, y) \in A_{det} | \hat{w}_n(x, y) = 1\}$ and $\mathbf{B} = \{(x, y) \in A_{det} | \hat{w}_n(x, y) = -1\}$. $N_{\mathbf{A}}$ and $N_{\mathbf{B}}$ are the number of pixels of the sets $\mathbf{A}$ and $\mathbf{B}$ respectively. In the case that the watermark is casted all over the embedding region, the detector output is assumed to follow a normal distribution with mean value $\bar{R} = 2h$, where h is the watermark power. This means that, ideally, if the watermark exists the detector output should be $R = 2h$ and otherwise $R = 0$.

We choose not to use masking in the detection stage, because the local variance may have changed significantly due to manipulations. The response is thus computed over the entire expected area of the embedded watermark. The detector output (12) must be compared against a proper threshold R_{thr} that will inform us with a satisfying certainty about the presense or the absense of the watermark.

6. EXPERIMENTAL RESULTS

In order to demonstrate the robustness of the watermarking method to various attacks, we tested it on 37 color images (of size 350×286) of the M2VTS frontal face image database. The feature extraction success rate was 84% for combined eye and mouth detection. Figure 1 shows results for feature detection and subsequent watermark detection after several attacks, for a sample image of the database. Figure (a) shows the original image and in figure (b) the detected eyes and mouth are denoted with crosses on the image. We can see that the localization is quite precise.

Figure (c) shows the watermarked image and figure (d) shows the normalized difference between the watemarked and the original image. The size of the prototype watermark is 128×128, the watermark power is $h = 3$ and the chaotic map parameter is $\lambda = 1.8$. The k and l parameters for defining the watermark spread over the face region are both fixed at value 1.25. The visual masking threshold was chosen to be $T_v = 0.002$. The threshold forced most of the prototype watermark to be casted, as depicted in figure (d).

In Figure (e) we can see that the feature detection results on the watermarked image are almost identical to the ones on the original image. Figure (f) shows the experimental distributions for 100 watermarks detected on the original image (left pdf) and the corresponding watermarked image (right pdf). The detector output is normalized according to its mean value. Figure (g) shows the features on the watermarked image, after being compressed by JPEG of ratio 1:40 approximately. We see in figure (h) that the distributions have now approached each other because of the large distortion imposed by the compression. The casting region is not very wide and a certain amount of the watermark energy is lost. This addresses the fact that, in order to retain reasonable false acceptance and false rejection rates, we should choose a detection threshold R_{thr} that is smaller than the theoretical one which, after normalization, should normally be at about 0.5. Rotation by 12^0, as depicted in figures (i) and (j), and scaling by a factor of 1.2 at each dimension, as shown in figures (k) and (l), do not impose larger degradation. Therefore, a threshold of $R_{thr} = 0.12$ can be chosen for watermark detection.

Table 1 shows the false acceptance rates (FAR) and false rejection rates (FRR) for all of the attacks depicted in figure 1, using the defined threshold. We can observe that the results are quite good, having in mind that a trade off was required in order for the detection rates to be reasonable for every attack.

The features are not expected to be localized at exactly the same positions as in the original image. This can be faced efficiently by testing the correlation output for small changes in the height, width, center and orientation of the prototype watermark. The detection is expected to give a peak for the correct height, width center and orientation of the originally embedded watermark because of the high degree of sensitivity of the watermark to geometric operations. Response to a false watermark is expected to give a different but insignificant peak for some false geometric parameters selection, because different watermarks are highly uncorrelated. The detection response in the diagrams is therefore always shown for the correct size, position and orientation of the watermark. The method is expected to perform better for larger images than the ones used in our experiments.

7. CONCLUSIONS

In the present paper we developed a method for embedding and detecting watermarks in color frontal face images. Color information was exploited in order to obtain a good approximation of the skin-colored region of the face, in which to search for salient features like the eyes and the mouth, using a geometric model matching method. The prototype watermark used for casting was chosen to be a chaotic one, modified in such a way as to retain certain low-pass properties. The watermark is geometrically adapted before embedding, using the extracted feature positions and face region orientation. A correlation detector is employed in order to decide about the presense of a possible watermark. The color segmentation and feature localization technique precedes both embedding and detection stages. A visual masking technique is added in order to avoid annoying artifacts imposed by the casted watermark. Experimental results display the robustness of the method and address the fact that the method is especially suitable for databases of large images.

8. ACKNOWLEDGEMENTS

The present work has been carried out within the framework of the EU LTR project INSPECT.

9. REFERENCES

[1] N. Nikolaidis and I. Pitas, "Copyright Protection of Images using Robust Digital Signatures", in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP '96*, May 1996, Atlanta, Georgia, vol. 4, pp. 2168-2171.

[2] W. Bender, D. Gruhl, N. Morimoto and A. Lu, "Techniques for data hiding", *IBM Systems Journal*, vol. 25, pp. 313-335, 1996.

[3] I.J. Cox, J. Killian, T. Leighton and T. Shamoon, " Secure Spread Spectrum Watermarking for Multimedia", in *IEEE Trans. on Image Processing*, **6**(12), pp. 1673-1687.

[4] A. Piva, M. Barni, F. Bartolini and V. Capellini, "DCT-based watermark recovering without resorting to the uncorrupted original image", in *Proc. IEEE Int. Conf. on Image Processing (ICIP'97)*, October 1997, Santa Barbara, California, vol. 1, pp. 520-523, 1997.

[5] K. Sobottka and I. Pitas, "A novel method for automatic face segmentation, facial feature extraction and tracking", *Signal Processing: Image Communication*, **12**(3), pp. 263-281.

[6] S. Tsekeridou and I. Pitas, "Facial feature extraction in frontal views using biometric analogies", in *Proc. of EUSIPCO '98*, September 1998, Rhodes, Greece, vol. 1, pp. 315-318.

[7] A.G. Bors and I. Pitas, "Object segmentation in 3-D images based on alpha-trimmed mean radial basis function network", in *Proc. of EUSIPCO '98*, September 1998, Rhodes, Greece, vol. 2, pp. 1093-1096.

[8] A.K. Jain, "Fundamentals of Digital Image Processing", Prentice-Hall, New Jersey, 1989.

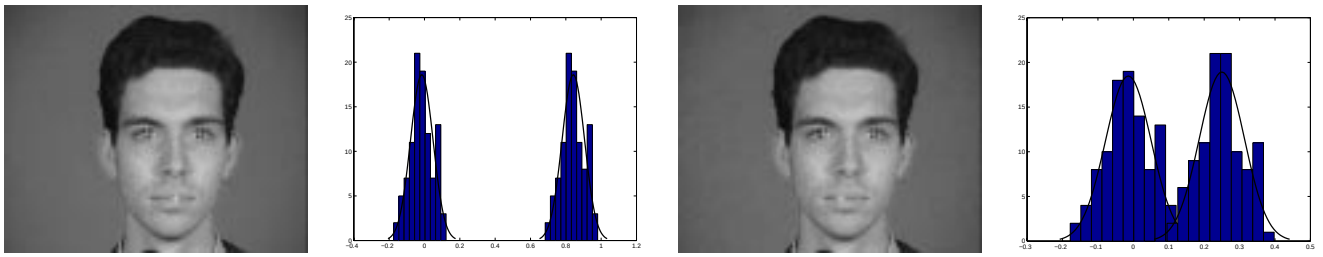
[9] G. Voyatzis and I. Pitas, "Chaotic Watermarks for Embedding in the Spatial Digital Image Domain", in *Proc. of ICIP '98*, October 1998, Chicago, Illinois, vol. 2, pp. 432-436.

[10] R.L. Devaney, "An introduction to dynamical systems", Penjamine/Cummings, 1986.

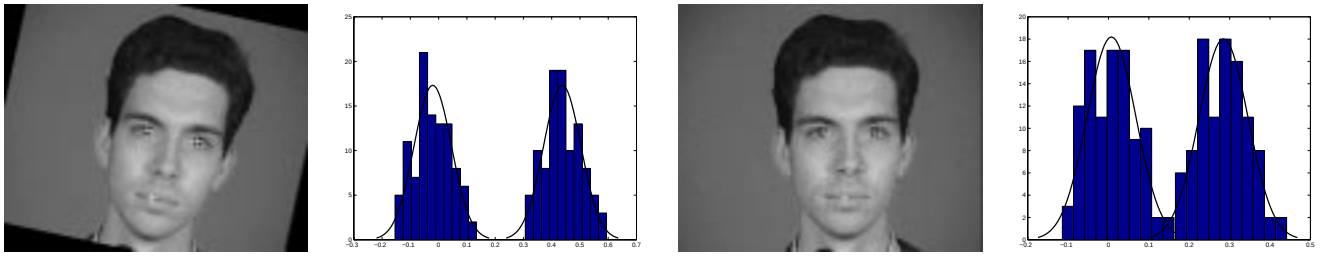
attack	FAR	FRR
no attack	0.0169	$4.052 \cdot 10^{-30}$
1:40 JPEG compression	0.018882	0.020794
rotation by 12^0	0.017046	$7.9213 \cdot 10^{-7}$
scaling by 1.2	0.031461	0.0034499

Table 1: Detector rates for several attacks.

(a) Original image (b) Detected features (c) Watermarked image (d) Difference image



(e) Features after watermarking (f) Experimental distributions (g) Features after compression (h) Experimental distributions



(i) Features after rotation (j) Experimental distributions (k) Features after scaling (l) Experimental distributions

Figure 1: Examples of salient feature extraction and watermark detection after several manipulations.