Face Detection Using Spectral Histograms and SVMs

Christopher A. Waring and Xiuwen Liu

***Abstract*—We present a face detection method using spectral histograms and support vector machines (SVMs). Each image window is represented by its spectral histogram, which is a feature vector consisting of histograms of filtered images. Using statistical sampling, we show systematically the representation groups face images together; in comparison, commonly used representations often do not exhibit this necessary and desirable property. By using an SVM trained on a set of 4500 face and 8000 nonface images, we obtain a robust classifying function for face and nonface patterns. With an effective illumination-correction algorithm, our system reliably discriminates face and nonface patterns in images under different kinds of conditions. Our method on two commonly used data sets give the best performance among recent face-detection ones. We attribute the high performance to the desirable properties of the spectral histogram representation and good generalization of SVMs. Several further improvements in computation time and in performance are discussed.**

***Index Terms*—Face detection, pattern classification, pattern recognition, special histogram representations, support vector machines.**

I. Introduction

FACE detection has gained increased interest in recent years. As computers become faster and more affordable, many applications that use face detection/localization are becoming an integral part of our life. For example, face recognition systems are being tested and installed at airports to provide a new level of security; human-computer interfaces based on facial expressions and body gestures are being exploited as ways to replace the traditional interfaces such as the mouse and the keyboard. These and other related applications all require, as an initial step, some form of face detection/localization, which can be simply defined as follows [32]: Given an image $\mathbf{I}$, find all occurrences of faces and the extent of each face in $\mathbf{I}$. This definition implies that some form of discrimination must be made between faces and all other objects. However, there are many difficulties and challenges associated with face detection. Variations in lighting conditions can make face images appear substantially different. Additives such as beards, mustaches, and glasses can augment the global structure of the face such as the jaw line and mask local features such as corners of the mouth. In addition, the large amount of intra-class variation[1] amongst all faces makes reliable face detection intrinsically difficult.

Manuscript received August 1, 2003. This work was supported in part by the NSF under Grant IIS-0 307 998 and the NGA NMA under Grant 201-01-2010. This paper was recommended by Associate Editor Bruce Draper.

The authors are with the Department of Computer Science, The Florida State University, Tallahassee, FL 32306 USA (e-mail: chwaring@cs.fsu.edu; liux@cs.fsu.edu).

Digital Object Identifier 10.1109/TSMCB.2005.846655

[1]While all faces posses to a certain degree a similar structure, location, alignment, size, and shape of facial features can vary greatly from person to person.

At the core, face detection requires an effective discrimination function between face and nonface patterns. Accordingly, existing face-detection approaches fall into one of four categories (see [32] for a recent survey): knowledge-based methods, template-based methods, feature invariant methods, or appearance-based methods. Knowledge-based methods (e.g., [9]) attempt to describe all the face patterns using rules based on human knowledge such as the fact that all faces have two eyes and a mouth. However, they are difficult to use to detect faces in real images as the translation of human knowledge into well formed rules is nontrivial. If the rules are too restrictive, many faces will be ruled out, resulting in false negatives; on the other hand, if the rules are too general, nonface patterns will be included in the face class, resulting in false positives. Template-based methods (e.g., [35]) represent the face class by (a set of) templates with allowable deformations that rely on the alignment of feature points. However, feature points can become corrupted by lighting variations, pose, additives, or facial expression changes, making the alignment of feature nodes on the template to features on the input difficult. In addition, if the amount of deformations for templates are too constrained, faces may be missed during detection; in contrast, if the amount of deformations are too flexible, false detections will be introduced. Feature-invariant methods (e.g., [34]) are hard to use in detecting faces in real images as it is difficult to find features that are truly invariant with respect to all faces and large perturbations in lighting, pose, and expressions. To overcome some of the difficulties, appearance-based methods (e.g., [8], [21], [27], [28], [33]) provide several key advantages and are widely used in face detection. Specifically, as they allow one to learn the models from training data, the large amount of intra-class variation, expression, and pose can be accounted for in training by using a large training set. However, as the training set is limited in practice and one is interested in detecting faces in images that have not been seen, generalization becomes the key issue among appearance-based methods, which is largely determined by the underlying representation and the classifying function.

In this paper, we present an appearance-based method using spectral histograms [17] as representation and support vector machines (SVMs) [26] as classifiers. Unlike some commonly used representations, the spectral histogram, which is shown through statistical sampling, gives good generalization by grouping only perceptually similar images together. With an SVM, this gives rise to a decision function that discriminates face and nonface patterns reliably in images under different kinds of conditions and results in the best performance on two commonly used face datasets.

The rest of the paper is organized as follows. Section II introduces the spectral histogram representation and shows its

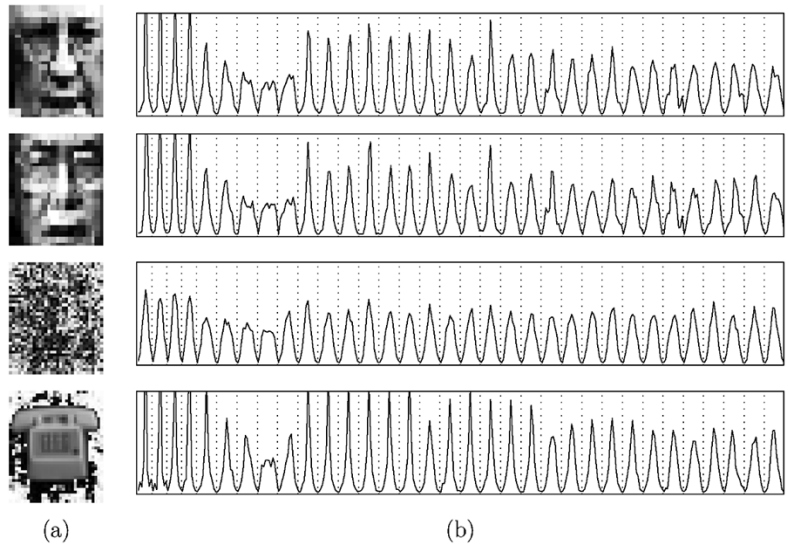


Fig. 1. Spectral histogram of four images. (a) Input images. (b) Corresponding spectral histogram using the 33 chosen filters.

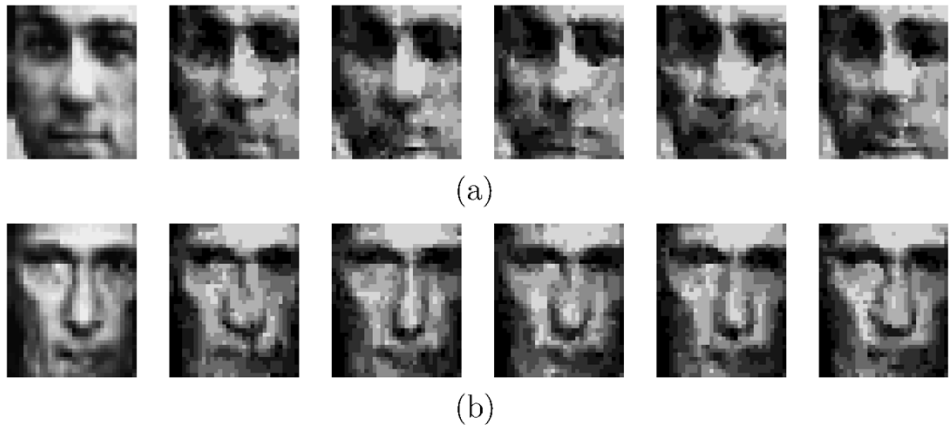


Fig. 2. Two face images along with synthesized images through sampling using the spectral histogram representation. In each subfigure, the left-most image is the original image, and the rest are typical samples of the set with similar spectral histogram representation of the given one.

sufficiency for representing faces through sampling. Section III briefly describes SVMs [26]. Section IV discusses the proposed algorithm in detail, including the preprocessing stage, the training stage, the detection stage, and the post-processing stage. Section V shows our experimental results. Section VI discusses possible further improvements in computation and performance. Section VII concludes the paper.

II. Spectral Histogram Representation

A. Definition and Properties

Given an input image $\mathbf{I}$ and a set of filters $\{F^{(\alpha)}, \alpha = 1, \ldots, K\}$, a subband image $\mathbf{I}^{(\alpha)}$ is computed through linear convolution given by $\mathbf{I}^{(\alpha)}(v) = F^{(\alpha)} * \mathbf{I}^{(\alpha)}(v) = \sum_u F(u)\mathbf{I}(v-u)$. The histogram of each subband image $\mathbf{I}^{(\alpha)}$ is given by $\mathbf{H}_{\mathbf{I}^{(\alpha)}}(z) = (1)/(|\mathbf{I}|)\sum_v \delta(z - \mathbf{I}^{(\alpha)}(v))$. The spectral histogram representation [17] of $\mathbf{I}$ with respect to the chosen filters is defined as the concatenation of $\mathbf{H}_{\mathbf{I}^{(\alpha)}}$, given by $\mathbf{H}_{\mathbf{I}} = (\mathbf{H}_{\mathbf{I}^{(1)}}, \ldots, \mathbf{H}_{\mathbf{I}^{(K)}})$. In other words, each image is represented by the histograms of filtered images. Fig. 1 shows the spectral histogram representation of several images using the 33 filters used throughout this paper (see below). These examples show that the spectral histograms of face images are similar while they are different for images from different classes.

In the spectral histogram representation, local features of an image are captured through filtering as the responses of individual filters depend on local structures, and the global structures are implicitly captured by the constraints imposed by the histograms of different filtered images. The representation is nonparametric in nature and is effective to characterize different kinds of patterns. One distinctive advantage of the spectral histogram representation is that two images do not need to be aligned in order to be compared (see Fig. 2 for examples) due to the fact that the spectral histogram representation is not sensitive to perturbations of local image features. This reduces the number of required training images for both face and nonface classes.

To specify a spectral histogram representation, one needs to choose a set of filters. Following [17] and [37], three different types of filters are used in this paper, including gradient filters, Laplacian of Gaussian filters, and Gabor filters. Gradient filters

used include $D_x = [0.0 \ -1.0 \ 1.0], D_{xx} = [-1.0 \ 2.0 \ -1.0], D_y = [0.0 \ -1.0 \ 1.0]'$, and $D_{yy} = [-1.0 \ 2.0 \ -1.0]'$, where $'$ indicates matrix transpose. Gabor Filters are defined as

$$G(x, y \,|\, T, \theta) = e^{-\frac{1}{2*T^2}(4(x\sin\theta + y\cos\theta)^2 + (-x\cos\theta + y\sin\theta)^2)} \times e^{-i\frac{2\pi}{T}(x\cos\theta + y\sin\theta)}$$

where T is the scale and θ the orientation. The Laplacian of Gaussian filters is given by $LoG(x, y, T) = (x^2 + y^2 - T^2)e^{-(x^2+y^2)/(T^2)}$, where $T = \sqrt{2\sigma}$ determines the filter's spatial scale, and σ corresponds to the variance of the Gaussian distribution. The 33 filters used in this paper are as follows:

- four Gradient filters D_x, D_y, D_{xx}, D_{yy};
- five LoG filters with $\mathrm{T} = \sqrt{2}/2, 1, 2, \sqrt{32}/3$ (expanded from $T = 1$) and $\sqrt{32}/3$ (expanded from $T = 2$);
- twenty four Gabor filters used with $\mathrm{T} = 2, 5, 12, 16$ and $\Theta = 0°, 30°, 60°, 90°, 120°, 150°$.

To characterize face patterns more effectively, we have included large kernels to impose a more rigid global structure. By expanding smaller kernels to approximate the larger ones,[2] this is achieved without increasing computation. This set of filters captures the structures that are important to face patterns and is found to be effective for face detection in images under different kinds of conditions. Note, however, that there is no optimality associated with the set. With a recent optimization technique proposed by Liu and Srivastava [15], an optimal set of filters for face detection can be learned by maximizing the performance on the training set, which needs to be further explored.

B. Face Synthesis Through Sampling

As discussed in the previous section, the underlying representation plays a critical role in generalization performance of any appearance-based face-detection system. By analyzing generalization performance, Liu *et al.* [16] showed that a necessary condition for a good representation is that it should group only images from the same class together. For face detection, as we are concerned with the set of face patterns and the set of all other images, a good representation should group face patterns together and keep all other pattern groups away from the face ones. If the chosen representation groups patterns from different classes together, the detection performance from any classifier will be limited. If the chosen representation groups nonface patterns into the face class, there would be inherently false positive detections; likewise, if the representation groups face patterns into the nonface class, there will be inherently false negatives.

One way to analyze the intrinsic generalization of a given representation is to generate random samples from the set consisting of images with identical representations [16]. Technically, given a representation $\mathcal{G}$ and an image $\mathbf{I}_{\text{obs}}$ from a known class, we want to sample from $\{\mathbf{I} | \mathcal{G}(\mathbf{I}) = \mathcal{G}(\mathbf{I}_{\text{obs}})\}$. Because of the high dimensionality of $\mathbf{I}$ [in this paper, the dimension of $\mathbf{I}$ (a 21×21 window) is 441], enumeration is computationally infeasible. A common way to overcome this problem is to use statistical sampling techniques [30], and here, we use a Gibber sampler with simulated annealing [36] on the induced Gibbs distribution given by

$$q(\mathbf{I}) = \frac{1}{Z_\Theta} \exp\left\{-\frac{\xi(\mathbf{I})}{\Theta}\right\} \tag{1}$$

where Θ denotes the temperature, and $\xi(\mathbf{I})$ is called the energy, which is given by

$$\xi(\mathbf{I}) = D(\mathcal{G}(\mathbf{I}), \mathcal{G}(\mathbf{I}_{\text{obs}})) \tag{2}$$

where D is a distance measure between the representations. Intuitively, this is achieved by updating pixels according to the conditional probability computed from (1). For an algorithmic description of the sampling process, see Zhu *et al.* [36].

To demonstrate the sufficiency of the spectral histogram representation, we use it as $\mathcal{G}$ in (2) and use the following distance measure between two images:

$$KL(\mathbf{H}_{\mathbf{I}_1}, \mathbf{H}_{\mathbf{I}_2}) = \sum_{\alpha=1}^{K} \sum_{z} \left(\mathbf{H}_{\mathbf{I}_1^{(\alpha)}}(z) - \mathbf{H}_{\mathbf{I}_2^{(\alpha)}}(z)\right) \times \log \frac{\mathbf{H}_{\mathbf{I}_1^{(\alpha)}}(z)}{\mathbf{H}_{\mathbf{I}_2^{(\alpha)}}(z)}. \tag{3}$$

Fig. 2 shows two examples of synthesized face images. In both cases, the spectral histogram representation captures the perceptual characteristics of the original images. These examples show that the spectral representation is able to capture important local features, e.g., the eyes, nose, and mouth. The global structure has also been captured as these features roughly lie in the locations where we would expect to see them in a face. Note that the variations amongst the synthesized face images are desirable as they show the spectral histogram representation groups' perceptually similar patterns together.

In comparison, we have also analyzed the intrinsic generalization of linear representations [2], [12], [19], [31]. We use the same Gibbs, sampler except that the energy function is defined as the Euclidean distance between the projections of $\mathbf{I}$ and $\mathbf{I}_{\text{obs}}$ along a given linear basis. Fig. 3 shows the results for the two images used in Fig. 2. In each subfigure, the bottom row shows the corresponding reconstructed image for the one given at the top. As expected, in each group, all the reconstructed images are identical as they all have the same linear representation. However, these images perceptually differ greatly and belong to different classes; this is because linear representations model pixels independently and, thus, allow some to change freely as long as their changes are cancelled by others. As these images have the identical representation, the use of a classifier cannot alleviate this problem. If a classifier classifies correctly the left-most face images in Fig. 3, all other images must also be classified as face patterns, resulting in false positives. On the other hand, if a classifier classifies correctly the nonface images, the face images must be classified also as nonface patterns, resulting in false negatives. Because of this, methods based on linear representations are intrinsically sensitive to different patterns in the background.

It should be noted that reconstruction and sampling are very different. Sampling is concerned with the drawing of random

[2]This is implemented by $\mathbf{I}^{(\alpha)}(v) = \sum_u F(u)\mathbf{I}(v - ku)$; we have used $k = 2, k = 3$, and $k = 4$.

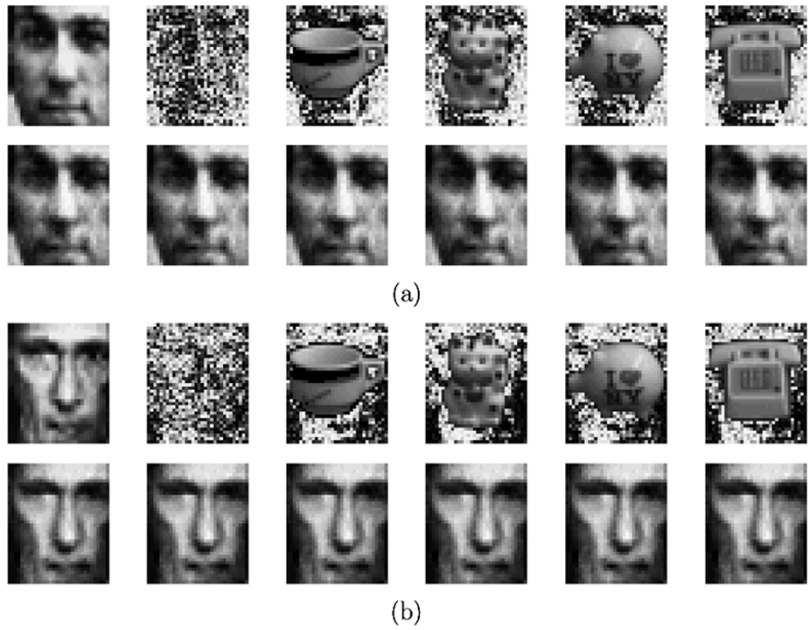


Fig. 3. Synthesized images of faces using linear representations. Here, the basis is generated using principal component analysis (PCA). In each subfigure, the top row shows the original image (the left-most one) and five typical samples from the set with the same linear representation; the bottom row shows the corresponding reconstructed image to show the fact that all of the images indeed have the same linear representation. Note that in the four synthesized images on the right, the center object is used as the boundary condition during sampling in that they are not updated.

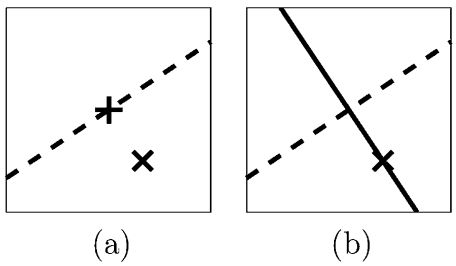


Fig. 4. Difference between sampling and reconstruction. Here, the dashed line represents a 1-D subspace in a two-dimensional space. For a given (face) image (marked as '×'), the sampling is to draw a random point along the solid line in (b), while the reconstructed image is a single point given by '+' in (a).

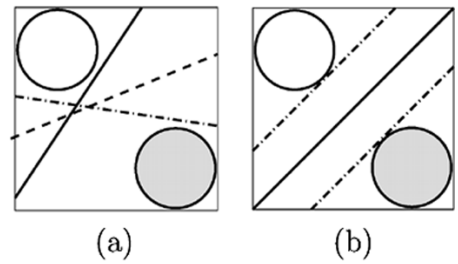


Fig. 5. Toy example that shows a small set of possible decision boundaries for an MLP and the decision boundary for an SVM. (a) Set of possible decision boundaries that could be derived by an MLP. Each line depicts a possible decision boundary. (b) Decision boundary given by an SVM. The thick line is the decision boundary, and the two dotted lines are the margins for the decision boundary.

samples from a set sharing the same representation, while reconstruction is a deterministic procedure of recovering the original image from its low-dimensional representation. To emphasize this point, Fig. 4 shows an illustration using a one-dimensional (1-D) linear representation. For the given image "×," the corresponding reconstructed image is the point "+"shown in Fig. 4(a), while the sampling can give any point along the solid line shown in Fig. 4(b). This shows an important point that the sufficiency of a representation cannot be solely justified based on the reconstruction accuracy; statistical sampling techniques must be used in order to analyze its generalization properties.

III. SVMs

In this paper, we choose the SVM [26], [20] as the classifying function. One distinctive advantage this type of classifier has over traditional neural networks is that SVMs achieve better generalization performance. While neural networks such as multiple layer perceptrons (MLPs) can produce low error rate on training data, there is no guarantee that this will translate into good performance on test data. MLPs minimize the mean squared error over the training data (*empirical risk minimization*), where SVMs use an additional principal called *structural risk minimization* [26]. The purpose of structural risk minimization is to give an upper bound on the expected generalization error.

To illustrate the difference, Fig. 5 shows a toy example with two classes. While there are numerous such decision planes that can give zero error on the training set [three are shown in Fig. 5(a)], the performance of these decision boundaries will vary when applied to unseen data. Fig. 5(b) shows the decision boundary given by the SVM, constructed as the one that separates two classes with the maximum distance between their margins. As shown by Vapnik [26], this maximal margin decision boundary can achieve optimal worst-case generalization performance. Note that SVMs are originally designed to solve problems where data can be separated by a linear decision boundary. By using kernel functions (see Osuna *et al.* [20] for details), SVMs can be used effectively to deal with problems that are not linearly separable in the original space. Some of the commonly

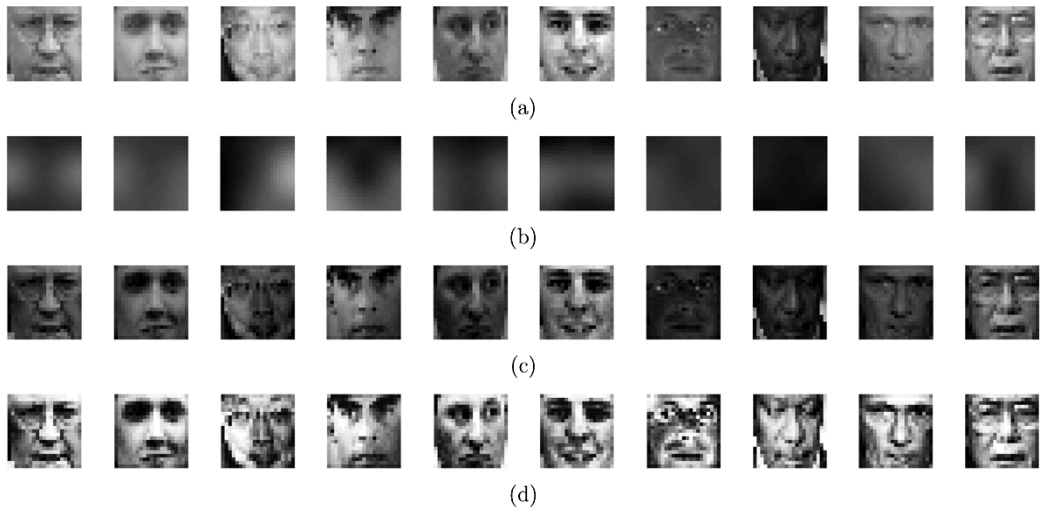


Fig. 6. Steps in the preprocessing stage. (a) Ten 21×21 images from the training set. (b) Corresponding minimal brightness plane for each face image. (c) Illumination-corrected images by subtracting the minimal brightness plane from the original image. (d) Preprocessed images.

used kernels include Gaussian Radial Basis Functions (RBFs), polynomial functions, and sigmoid polynomials. In this paper, we choose the Gaussian RBF kernel function on the spectral histogram representation of the input.[3] As the spectral histogram is naturally normalized, it simplifies the choice of parameters for the kernel function and the SVM. Our implementation of SVMs is based on the SVM-light package by T. Joachims.[4]

IV. Face Detection

This section discusses in detail each stage of the proposed method including preprocessing, training, detection, and post-processing.

A. Preprocessing

As nonuniform illumination can alter the appearance of features in images, it is imperative to correct this effect. We use a modified version of a method proposed by Sung and Poggio [25]. For each 7×7 block, the minimum value is computed, which generates a 3×3 minimal brightness plane on a 21×21 window. The brightness plane is then resized by bilinear interpolation to a 21×21 minimal brightness plane. The resulting minimal brightness planes for the images shown in Fig. 6(a) are given correspondingly in Fig. 6(b). The minimal brightness plane is then subtracted from the original image. As shown in Fig. 6(c), the procedure corrects for the effect of nonuniform illumination on the original images. Finally, the illumination-corrected images are further normalized by applying histogram equalization, and the preprocessed images are shown in Fig. 6(d). These preprocessing steps are applied to all the face and nonface training images and each window in each test image. Compared to Sung and Poggio's original iterative gradient method [25], our algorithm is computationally simpler and seems sufficient for detection using the spectral histogram representation.

[3]In a previous study [28], we used an MLP as the classifier and obtained slightly worse results; within SVMs, one can use different kernel functions, and we expect that the results would be similar.

[4]Obtained from http://svmlight.joachims.org.

B. Training

The training face images[5] were generated from real images (separate from the images used in test) by fitting a 21×21 image window around each face using an affine transform based on eye positions, as described in [25]. The original set consists of 1500 images, and here, we increased it to 4500 by rotating each image randomly by r and -r, where r ranges from $0^\circ < r \leq 15^\circ$. This increases the rotation invariance of the proposed method. A problem particular to face detection is how to select effective nonface training images. As nonface images include all kinds of images, a prohibitively large set is needed in order to be representative, which would also require an infeasible amount of computation in training. To alleviate this problem, a bootstrapping method proposed by Sung and Poggio [25] is used to incrementally build the nonface images in the training set. Starting with 1500 random nonface images, we generated in total 8000 nonface images using the bootstrapping method. All the training images (4500 face and 8000 nonface ones) are first preprocessed according to the preprocessing stage discussed previously. The spectral histogram of the illumination-corrected face and nonface images in the training set is then calculated using the 33 filters. The resulting spectral histograms are used as input to train an SVM with an RBF kernel function. The trained SVM is then used to classify image windows in test images.

C. Detection and Post-Processing

As the discrimination function is designed on a 21×21 image window, it implicitly requires that all faces lie roughly within a 21×21 window. Here, a three-level Gaussian pyramid is built by successively down sampling the test image by a factor of 1.1. Each level of the pyramid is processed independently at the detection stage. This achieves some degree of scale invariance. For the image at each level, a 21×21 image window is moved pixel by pixel, canvassing the entire image. Each test window is normalized using the method outlined in Section IV-A. The spectral histogram is then computed using the 33 chosen filters

[5]These training images, along with the test ones, were provided by H. Schneiderman, Carnegie Mellon University, and were used in [21], [22], and [25].

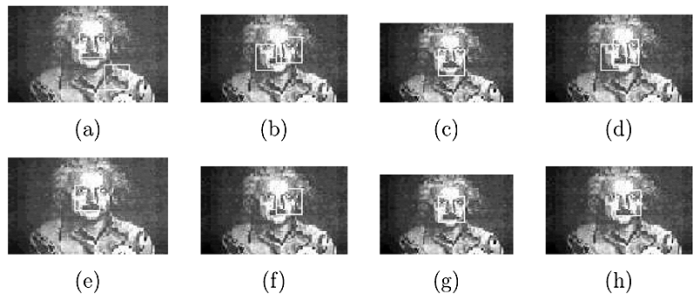


Fig. 7. Difference between no thresholding and adaptive thresholding. Images in (a)–(c) were created with no thresholding used on the region growing results, and (d) shows the final result. Images in (e)–(g) were created using an adaptive threshold of $2/3$, and (h) shows the final result.

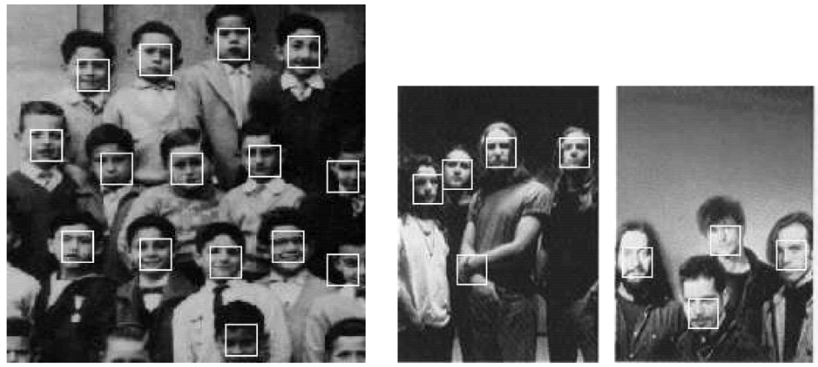


Fig. 8. Some typical results on test set 1. In each panel, the first number is the total number of faces in the image (the given ground truth), the second one the number of faces detected by the proposed algorithm, and the third one the number of false detections given by the proposed algorithm.

and stored. Once all the histograms have been computed, the histogram matrix is fed to the trained SVM for classification. If a local window is classified as a face, its center is saved. A region-growing algorithm is used to coalesce nearby positive outputs (given by the SVM) in small regions, and thresholding on the number of positive outputs is then applied to each region.

An important question is how to choose a proper threshold to achieve good performance on different kinds of images. For instance, if an arbitrary threshold is set, the generalized performance may not be good. One can test a set of images and find some average threshold [21]. However, the number of positive outputs at a face region can vary from image to image. Therefore, a threshold that changes with images is desirable. Using a set of 20 images ranging from single face images to images with multiple faces, we found empirically that in all the images, face regions possess the highest number of positive outputs per region. The number of positive outputs per region for each face α_{face_i} was found to fall within $(\alpha_{\max})/(2) < \alpha_{\text{face}_i} \leq \alpha_{\max}$, where $\alpha_{\max}$ denotes the maximum positive outputs per region for an image. Instead of using a fixed value, the threshold of positive outputs per region is given by $\delta \times \alpha_{\max}$; as $\alpha_{\max}$ can vary from image to image, we thus have the threshold. We call this adaptive thresholding. Three different values, with respect to $\alpha_{\max}, \delta = (1/2), (2/3)$, and $(3/4)$, were tested. $\delta = (2/3)$ was found to give the best overall performance. With $\delta = (1/2)$, the majority of faces were detected, but there were many false detections. Increasing δ from $(1/2)$ to $(2/3)$ only slightly decreased the number of face detections but sharply curbed the number of false detections. When δ was increased from $(2/3)$ to $(3/4)$, the number of faces detected and false detections dropped substantially. Fig. 7 shows the difference between no thresholding and adaptive thresholding. Notice that when no threshold is applied, as in Fig. 7(d), there are two final detections over the face. In contrast, when adaptive thresholding is utilized, there is only one final detection, as shown in Fig. 7(h).

Instituting adaptive threshold $\delta = (2/3)$, any region with fewer positive outputs from the SVM than $\delta \times \alpha_{\max}$ is discounted as a nonface. After region growing and thresholding, regions that are less than three pixels apart are coalesced. The centroid for each new region is computed and saved. Once the entire image pyramid has been processed, detections at each layer are examined. A detection is marked as final if and only if it is found in at least two concurrent levels of the image pyramid. A similar hierarchal approach was used by Rowley *et al.* [21] and Juell *et al.* [8]. A detection is registered as correct if it contains half or more of a face. Otherwise, it is labeled a false detection.

V. Experimental Results

The proposed face-detection algorithm was applied to two commonly used data sets that are separate from the training images. Test set 1 consists of 125 images containing 483 faces, and test set 2 consists of 23 images containing 136 faces. Test set 1 was originally used by Rowley *et al.* [21] and has become one of the standard test sets for face detection algorithms. Test set 2 was initially used by Sung and Poggio [25]. Figs. 8–11 show some of the typical results on test set 1 from the proposed algorithm. As is evident from these results, our algorithm demonstrates exceptional performance on images that range from high

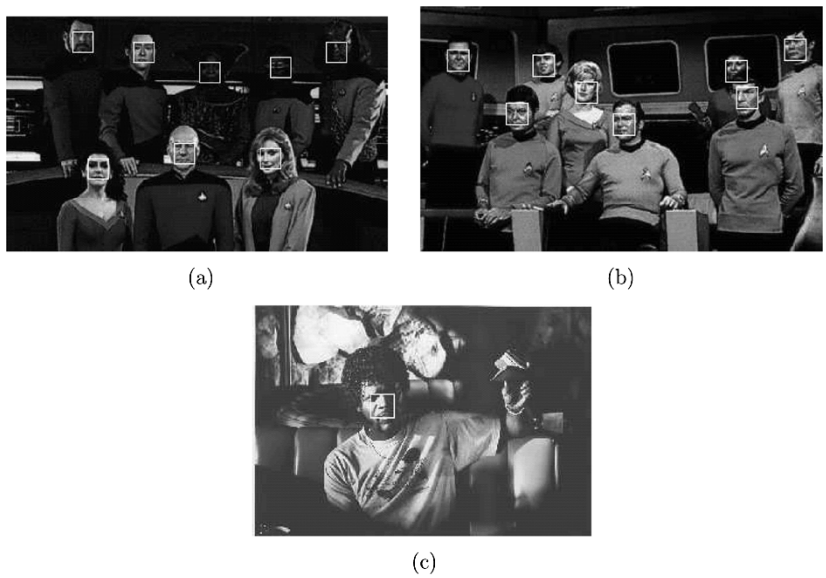
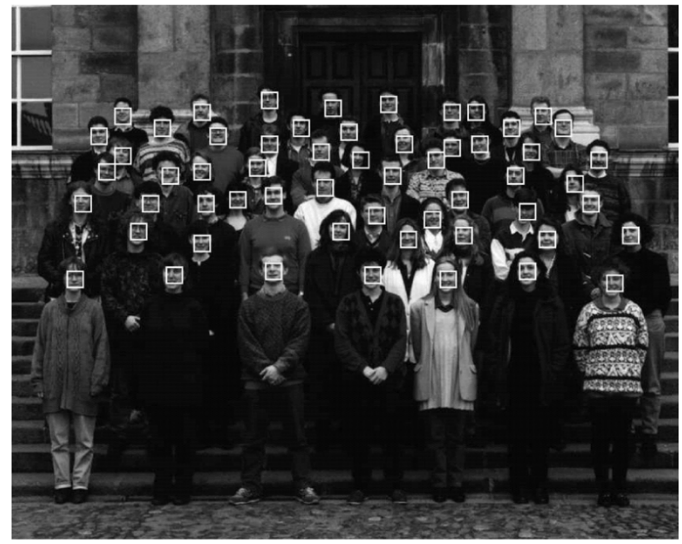


Fig. 9. Some more typical results on test set 1. See Fig. 8 for the legend. (a) $8/8/0$. (b) $8/8/0$. (c) $1/1/0$.

9 / 9 / 1

Fig. 10. Some typical results on test set 1. See Fig. 8 for the legend.



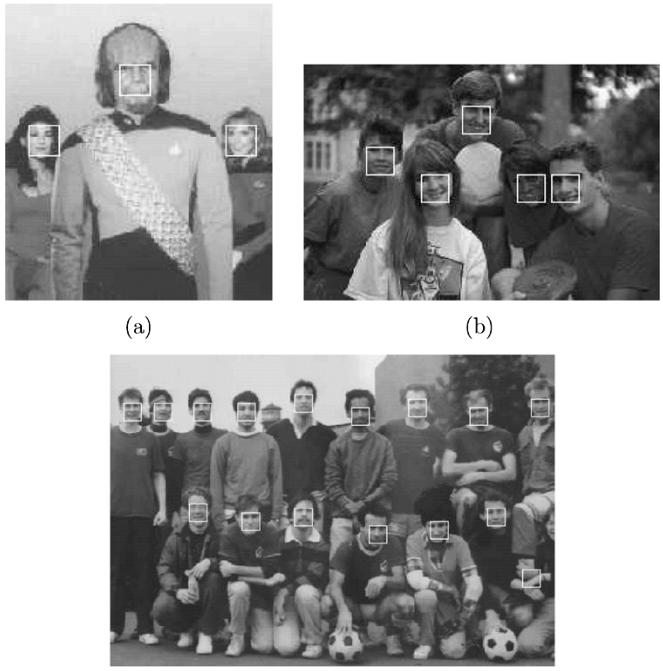
57 / 57 / 0

Fig. 11. Some more typical results on test set 1. See Fig. 8 for the legend.

quality to low quality. Fig. 8 shows two examples with relatively simple backgrounds, and the proposed algorithm detects all the faces accurately even when some of them are rotated (see Fig. 8(b)). Fig. 9 shows three examples with perfect performance. In Fig. 9(c), the man's face is detected correctly, despite the fact that his eyes are occluded by the visor that he wears and the complex background. In Fig. 10, we detect all instances of faces with one false detection located on the womans hands on the top level. In Fig. 11, we are able to detect all 57 faces with no false detections. Note that this image contains many different patterns, and faces are close to one another, making detection in this type of image difficult. Our algorithm not only detects all the instances of faces but also marks the detections accurately.

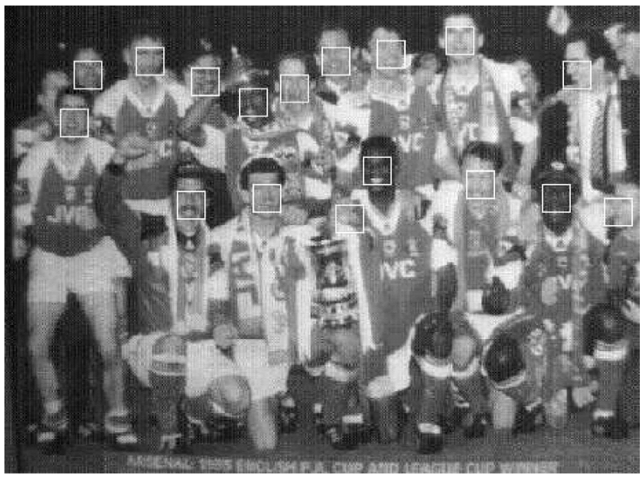
Figs. 12–14 show typical results from the proposed algorithm on test set 2. In Fig. 12, all the faces are detected correctly and accurately. Figs. 13 and 14 depict two low-quality images. In Fig. 13, there is a variety of expressions on the faces, and each face is posed differently. In Fig. 14, notice the three larger faces in the bottom right corner. These faces were too large to fit entirely in the 21×21 image window, but we are still able to detect them. As a result, our algorithm detects all the faces despite the low quality of the images in Figs. 13 and 14.

Although there are still false detections given by the proposed algorithm, unlike many other methods, the false detections in our results are localized in very specific areas and, thus, could be handled effectively. The majority of false detections are localized around the hands, such as in Figs. 8(b) and 12(c). This can be attributed to the relaxed topological ordering imposed by the spectral histogram representation. One can overcome this problem by either including images of hands in certain positions as negative training examples or imposing a stricter topological structure. The latter can be accomplished by computing



(a) (b) (c)

Fig. 12. Some more typical results on Test set 2. See Fig. 8 for legend. (a) $3/3/0$. (b) $5/5/0$ (c) $15/15/1$.



16 / 16 / 1

Fig. 13. Some more typical results on Test set 2. See Fig. 8.

the spectral histograms for specified subsections of an image window [13]. This would enforce a more rigid alignment of feature points as there would be less variation within each subregion of the window compared with the total variation within the entire window.

Table I shows a summary of our results along with that from other recent methods on the same test sets. The results from other methods are taken from Yang *et al.* [32]. As is evident from these results, our algorithm gives the best overall performance.

VI. Discussion

We have presented a face-detection method based on spectral histogram representations and SVMs and have demonstrated its performance on two commonly used data sets. The comparison shows that our method achieves the best performance with respect to both false detections and detection rate. The performance of our system can be attributed to the desirable properties of spectral histogram representations and the generalization property of SVMs. One may further improve the performance by integrating the spectral representation with other representations with more rigid topological ordering such as edge maps [5], Gabor jets [10], or templates [35], thus reducing further the number of false positives.

14 / 14 / 0

Fig. 14. Some more typical results on Test set 2. See Fig. 8.

In addition to the performance, this type of representations offers further benefits. For example, our approach is generic in nature and is applicable to other forms of object detection and recognition and not solely face detection. As shown by Liu *et al.* [14]–[16], the spectral representation can be used to enhance the performance on such tasks as face and object recognition. This could lead to the possibility of integrating the two systems as one. While several other methods (e.g., [22] and [27]) are applicable to detection of different objects, none of them has shown to be effective for recognition and classification of faces, objects, and textures. Due to the filters used, the proposed system exhibits a large degree of rotation invariance, which is demonstrated in Fig. 15. Results in Fig. 15(a) and (b) are both generated with no modification to the current system as they are tested using the same procedure for all images in the test sets. Note the faces with large rotation are detected correctly. By combing the results in (a) and (b), all the faces are detected correctly with no false detections, as shown in Fig. 15(c). This example demonstrates that our system can handle rotation up to 45°.

As the spectral histogram representation has been applied to recognition in different modalities such as infrared images [23], [14], it may also provide an effective representation to detect faces in infrared and other kinds of images. As shown by Srivastava and Liu [23], infrared imaging is not sensitive to illumination conditions, and thus, face detection based on infrared images may operate during the day and at night. A context-sensitive face-detection method may also be possible through the use of spectral histograms based on their effectiveness in clutter modeling [7], [24].

TABLE I
RESULTS FROM OUR METHOD ALONG WITH THAT FROM OTHERS.

Method	Test set 1		Test set 2	
	Detection Rate	False Detections	Detection Rate	False Detections
Waring & Liu	96.7 %	67	95.6 %	6
Yang, Ahuja, & Kriegman [31]	93.6 %	74	91.5 %	1
Yang, Ahuja, & Kriegman [31]	92.3 %	82	89.4 %	3
Yang, Roth, & Ahuja [33]	94.2 %	84	93.6 %	3
Yang, Roth & Ahuja [33]	94.8 %	78	94.1 %	3
Rowley, Baluja, & Kanade [21]	92.5 %	862	90.3 %	42
Schneiderman [22]	93.0 %	88	91.2 %	12
Colmenarez & Huang [3]	98.0 %	12758	N.A.	N.A
Sung & Poggio [25]	N.A.	N.A.	81.9 %	13

(a) (b) (c)

Fig. 15. Example that demonstrates large rotation invariance of the proposed method. (a) and (b) Results generated with the current system without modification. Note that the image in (b) is obtained by rotating that in (a) for 90°. (c) Results by combining the results from (a) and (b). See Fig. 8 for the legend. (a) $5/3/0$. (b) $5/3/0$. (c) $5/5/0$.

Compared to some recent methods (e.g., [11] and [27]), our method is significantly slower. It takes several minutes to process a typical test image. While spectral histograms of local windows can be calculated efficiently using *integral histogram images* [13], it takes time for the SVM to classify these large number of local windows. There are a few possible ways to reduce the computation time. One possibility is to use Adaboost-like algorithms [11], [27] to learn much more efficient classifiers. As these computations are intrinsically parallel, another way is to implement them on parallel machines or special-purpose chips such as field programmable gate arrays [29]. There are some other possible ways to decrease the running time. The image window could be moved every two or three pixels instead of every pixel. This would certainly lead to faster run times; however, the performance may suffer as the number of detections per region for faces may decrease. This can be further improved by moving the windows adaptively. In regions with low scores from the trained SVM, we can move more pixels; in regions with high scores, we can move detection windows pixel by pixel. This would dramatically reduce the computation and yet not reduce the number of detections in face regions. Another way is to have several face detectors that are organized hierarchically based on their computation complexity, as in [1] and [5]. A possible way to implement such a hierarchy for face detection is to use spectral histograms with a different number of filters. With fewer filters, one can obtain candidate regions faster and then apply more accurate spectral histogram models only on those candidate regions. This needs to be explored further.

VII. CONCLUSION

In this paper, we have shown that the spectral histogram representation is a good choice for face detection and yields results that are better than other recent methods both with respect to false detections and detection rate. In addition, without any modification, our system is able to achieve a respectable degree of rotation invariance. While most representations used in face detection are justified only based on empirical results, the sufficiency of the spectral histogram representation is shown systematically through statistical sampling. As our representation is generic in nature, it can be easily adapted to other forms of object detection and other tasks such as face recognition [16], object recognition [16], and texture classification [18]. With these results, we expect the spectral histogram representation to provide a unified representation for effective object detection and recognition.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers whose comments have improved the presentation of this paper significantly.

REFERENCES

[1] Y. Amit and D. Geman, "A computational model for visual selection," *Neural Comput.*, vol. 11, pp. 1691–1715, 1999.
[2] P. Belhumeur, J. Hespanha, and D. Kriegman, "Eigenfaces versus fisherfaces: Recognition using class specific linear projection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 7, pp. 711–720, Jul. 1997.
[3] A. Colmenarez and T. Huang, "Face detection with information-based maximum discrimination," in *Proc. IEEE Conf. Computer Vision Pattern Recogn.*, 1997, pp. 782–787.
[4] J. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by the two-dimensional visual cortical filters," *J. Opt. Soc. Amer. A*, vol. 2, no. 7, pp. 23–26, 1985.
[5] F. Fleuret and D. Geman, "Coarse-to-fine face detection," *Int. J. Comput. Vision*, vol. 41, pp. 85–107, 2001.
[6] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Comput.*, vol. 4, pp. 1–58, 1992.

[7] U. Grenander and A. Srivastava, "Probability models for clutter in natural images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 4, pp. 424–429, Apr. 2001.
[8] R. Juell and R. Marsh, "A hierarchal neural network for human face detection," *Pattern Recogn.*, vol. 29, pp. 781–787, 1996.
[9] C. Kotropoulos and I. Pitas, "Rule-based face detection in frontal views," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, vol. 4, 1997, pp. 2537–2540.
[10] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. von de Malsburg, R. P. Wurtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Trans. Comput.*, vol. 42, no. 3, pp. 300–311, Mar. 1993.
[11] R. Lienhart, A. Kuranov, and V. Pisarevsky, "Empirical analysis of detection cascades of boosted classifiers for rapid object detection," in *Proc. DAGM Pattern Recogn. Symp.*, 2003, pp. 297–304.
[12] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for facce recognition," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 467–476, Apr. 2002.
[13] X. Liu, "A computational framework for real-time scene interpretation," in *Proc. Applied Imagery Pattern Recogn. Workshop*, 2004.
[14] X. Liu and L. Cheng, "Independent spectral representation of images for recognition," *J. Opt. Soc. Amer. A*, vol. 20, no. 7, pp. 1271–1282, 2003.
[15] X. Liu and A. Srivastava, "Stochastic search for the optimal linear representations of images on spaces with orthogonality constraints," in *Proc. Int. Workshop Energy Minimization Methods Computer Vision Pattern Recogn.*, 2003, pp. 3–20.
[16] X. Liu, A. Srivastava, and D. Wang, "Intrinsic generalization analysis of low dimensional representations," *Neural Netw.*, vol. 16, no. 5/6, pp. 537–545, 2003.
[17] X. Liu and D. Wang, "A spectral histogram model for texton modeling and texture discrimination," *Vision Res.*, vol. 42, pp. 2617–2634, 2002.
[18] ——, "Texture classification using spectral histograms," *IEEE Trans. Image Process.*, vol. 12, no. 6, pp. 661–670, Jun. 2003.
[19] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, no. 7, pp. 696–710, Jul. 1997.
[20] E. Osuna, R. Freund, and F. Girosi, "Training support vector machines: An application to face detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, 1997, pp. 130–136.
[21] H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 1, pp. 23–38, Jan. 1998.
[22] H. Schneiderman, "A Statistical Approach to 3d Object Detection Applied to Faces and Cars," Ph.D. Dissertation, Carnegie Mellon Univ., Pittsburgh, PA, 2000.
[23] A. Srivastava and X. Liu, "Statistical hypothesis pruning for identifying faces from infrared images," *J. Image Vision Comput.*, vol. 21, no. 7, pp. 651–660, 2003.
[24] A. Srivastava, X. Liu, and U. Grenander, "Universal analytical forms for modeling image probabilities," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 9, pp. 1200–1214, Sep. 2002.
[25] K.-K. Sung and T. Poggio, "Example-based learning for view-based human face detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 1, pp. 39–51, Jan. 1998.
[26] V. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer-Verlag, 1995.
[27] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, 2004.
[28] C. A. Waring, "An Exploration of the Spectral Histogram Representation for Face Detection," Masters Thesis, Dept. Comput. Sci., Florida State Univ., Tallahassee, FL, Jul. 2003.
[29] G. Wall, X. Liu, F. Iqbal, and S. Foo, "Real-time texture classification using FPGA," in *Proc. Applied Imagery Pattern Recogn. Workshop*, 2004.
[30] G. Winkler, *Image Analysis, Random Fields and Dynamic Monte Carlo Methods: A Mathematical Introduction*, Second ed. Berlin, Germany: Springer, 2003.
[31] M. Yang, N. Ahuja, and D. Kriegman, "Mixture of linear subspaces for face detection," in *Proc. Int. Conf. Automatic Face Gesture Recogn.*, 2000, pp. 70–76.
[32] M. Yang, D. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 24, no. 1, pp. 34–58, Jan. 2002.
[33] M. Yang, N. Roth, and N. Ahuja, "A snow-based face detector," *Adv. Neural Inf. Process. Syst.*, vol. 12, pp. 855–861, 2000.
[34] K. Yow and R. Cipolla, "Feature-based human face detection," *Image Vision Comput.*, vol. 15, no. 9, pp. 713–735, 1997.
[35] A. Yuille, D. Cohen, and P. Hallinan, "Feature extraction from faces using deformable templates," in *Proc. IEEE Conf. Comput. Vision Pattern Recogn.*, 1989, pp. 104–109.
[36] S. Zhu, X. Liu, and Y. Wu, "Exploring texture ensembles by efficient markov chain monte carlo," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, no. 6, pp. 554–569, Jun. 2000.
[37] S. Zhu, Y. Wu, and D. Mumford, "Minmax entropy principle and its application to texture modeling," *Neural Comput.*, vol. 9, pp. 1627–1660, 1997.

Christopher A. Waring received the B.S. degree in pure mathematics and computer science in 2002 and the M.S. degree in computer science in 2003, both from the Florida State University, Tallahassee.

His research interests include computer vision and pattern recognition and their applications.

Xiuwen Liu received the B.Eng. degree in computer science in 1989 from Tsinghua University, Beijing, China, the M.S. degrees in geodetic science and surveying in 1995 and computer and information science in 1996, and the Ph.D. degree in computer and information science in 1999 from the Ohio State University, Columbus.

From August 1989 to February 1993, he was with Department of Computer Science and Technology, Tsinghua University. Since 2000, he has been with the Department of Computer Science, the Florida State University, Tallahassee. His current research interests include low-dimensional representations of images, statistical pattern recognition, manifold-based optimization and inference algorithms, computational modeling of visual perception and inference, and real-time vision algorithms and implementations using FPGAs.