Face Recognition using Tensors of Census Transform Histograms from Gaussian Features Maps

John A. Ruiz-Hernandez
john-alexander.ruiz-hernandez@inrialpes.fr

James L. Crowley
james.crowley@inrialpes.fr

Antoine Meler
antoine.meler@inrialpes.fr

Augustin Lux
augustin.lux@inrialpes.fr

Grenoble INP & INRIA Rhone-Alpes
Research Center
France

Abstract

This paper presents a new approach for face recognition based on the fusion of tensors of census transform histograms from Local Gaussian features maps. Local Gaussian feature maps encode the most relevant information from Gaussian derivative features. Census Transform (CT) histograms are calculated and concatenated to form a tensor for each class of Gaussian map. Multi-linear Principal Component Analysis (MPCA) is applied to each tensor to reduce the number of dimensions as well as the correlation between neighboring pixels due to the Census Transform. We then train Kernel Discriminative Common Vectors (KDCV) to generate a discriminative vector using the results of the MPCA. Results of recognition using MPCA of tensors-CT histograms from Gaussian features maps with KDCV is shown to compare favorably with competing techniques that use more complex features maps like for example Gabor features maps in the FERET and Yale datasets. Additional experiments were done in the Yale B+ extended Yale B Faces dataset to show the performance of Gaussian features map with hard illumination changes.

1 Introduction

Face recognition can be defined as the identification of individuals from images of their faces using a stored database of faces labeled with identity. Face recognition is a challenging task due to the large variety of appearance that a face may exhibit under variations in illumination and viewing position as well as variations in facial expression. It is common to consider the problem of face recognition using the relatively controlled imaging conditions, such as for passports and photo IDs, as a much easier problem then recognition from surveillance video images under a large variation in pose, aging, illumination, occlusion and facial expressions [1, 27].

Many of the more successful approaches use Gabor wavelets as an image descriptor, resulting in relatively high computational cost. For example, Zhang et al [26] propose the use of Local Gabor Binary Pattern Histogram Sequence (LGBPHS) which combine the magnitude part of Gabor feature and the LBP operator. In more recent work, Zhang et al [25] have explored encoding the face image with a Global Gabor Phase Pattern (GGPP) and a Local Gabor Phase Pattern (LGPP) derived under variations in orientation for the Gabor wavelet at a given scale (or spatial frequency). Tan and Triggs [18] use a fusion of Gabor and LBP feature sets, those feature sets are normalized in illumination and the process of recognition is improved with Kernel Discriminative Common Vectors (KDCV) [2].

In this work, we have explored the use of Gaussian derivative features calculated with a linear complexity half-octave Gaussian pyramid [6]. Derivative features have been used as a feature set for view invariant object recognition by Yokono and Poggio [23]. Gaussian features has been used with success in multidimensional receptive field histograms by Schiele and Crowley [17], invariant object recognition [8] and image descriptor [11, 19].

In this paper we associate Gaussian features by the nature of information that is encoded. Original scale space encode textural information and characteristic scale [10], first order Gaussian derivatives encode information from edges and sharp contrast changes. Second order Gaussian derivatives capture information about structure from bars, blobs and corners. We consider these different classes of Gaussian derivative features to provide complementary information.

In section 4, we propose a tensorial representation for Gaussian derivative histograms. Our method is inspired by recent work on face recognition by Rana et al [15]. This representation retains the spatial structure without loss of information due to vectorization of features as occurs in earlier approaches [25, 26]. In our approach, local difference of samples in a half-octave Gaussian pyramid are used to compute Gaussian feature maps. For each location in each Gaussian map, a Census Transform histogram is calculated and concatenated to form a local tensor. Only one tensor is computed for each class of Gaussian map. An MPCA (Multilinear Principal Component Analysis) [12] method is then applied to each tensor to reduce the dimensionality and inter-correlation due to the Census Transform. Finally we apply the KDCV (Kernel Discriminative Common Vectors) method to generate a discriminative vector as is explained in the sections 3 and 4. The experimental presented in section 5 and 6 demonstrate that this method improves recognition rates compared techniques when used with the Feret and Yale datasets.

2   Gaussian Derivative Features

This section describes how a half octave Gaussian pyramid can provide very fast computation of a local Gaussian norm jet composed of derivatives features of order 0, 1 and 2. This algorithm has a linear algorithmic complexity and is easily implemented on GPU or other SIMD style computing devices.

2.1   The Gaussian Jet

Koenderink has described how the visual appearance of a neighborhood can be represented by a local Taylor series expansion [9]. The coefficients of this Taylor series constitute a feature vector, referred to as the "Local Jet" that compactly represents the neighborhood appearance for indexing, matching and recognition. Ter Haar Romeny and others [16], have

shown that invariance to scale and orientation can be obtained when the local jet is computed using Gaussian Derivatives. The basis functions for this expansion are local derivatives computing using a Gaussian support:

$$G(x,y,\sigma) = \frac{1}{2\sigma\pi} e^{\frac{x^2+y^2}{2\sigma^2}} \tag{1}$$

Where σ is a scale factor that expresses the size of the support in terms of the second moment (or variance). This function is sometimes referred to a Gaussian Receptive Field [17]. The first and second derivatives of the Gaussian support are:

$$G_x(x,y,\sigma) = \frac{\partial G(x,y,\sigma)}{\partial x} = -\frac{x}{\sigma^2} G(x,y,\sigma) \tag{2}$$

$$G_y(x,y,\sigma) = \frac{\partial G(x,y,\sigma)}{\partial y} = -\frac{y}{\sigma^2} G(x,y,\sigma) \tag{3}$$

Gaussian derivative features capture information about changes of the surface normal and measure the intensity of edges [23]. The second order derivates are given by:

$$G_{xx}(x,y,\sigma) = \frac{\partial G(x,y,\sigma)}{\partial^2 x} = \left(\frac{x^2}{\sigma^4} - \frac{1}{\sigma^2} \right) G(x,y,\sigma) \tag{4}$$

$$G_{yy}(x,y,\sigma) = \frac{\partial G(x,y,\sigma)}{\partial^2 y} = \left(\frac{y^2}{\sigma^4} - \frac{1}{\sigma^2} \right) G(x,y,\sigma) \tag{5}$$

$$G_{xy}(x,y,\sigma) = \frac{\partial G(x,y,\sigma)}{\partial x \partial y} = \left(\frac{xy}{\sigma^4} \right) G(x,y,\sigma) \tag{6}$$

The second order derivatives are good descriptors for image features such as bars, blobs and corners. Higher order Gaussian derivatives are more sensitive to the image noise and can provide useful information in cases where the second order derivatives are stronger [8].

The use of Gaussian derivatives is motivated by their capacity to describe image neighborhoods and also by their representation of image appearance in specific orientations and frequencies [23]. This representation is well suited for object detection and learning.

2.2 The Half-Octave Gaussian Pyramid

Computing a scale invariant local Jet for an $M = N \text{x} N$ pixels image requires computing second order derivatives of the image at $\log_2 N$ scales. A linear complexity pyramid algorithm for this calculation has been known since the 1980's [4]. The result of this algorithm is a half-octave Gaussian Pyramid. An integer coefficient version of this algorithm [5] has been demonstrated using repeated convolutions of the binomial kernel $(1,2,1)$. Implementations that compute such pyramids on PAL sized images (and faster) exist for the current generation of computer stations, and can be embedded in dedicated a signal processor. The half-octave Gaussian pyramid for an $N \text{x} N$ pixels image is composed of up to $K = \log_2(N)$ images. Each image, $k \in [1,K]$ of the pyramid has been convolved with a Gaussian filter, $G(x,y,2^{\frac{k}{2}})$, and can be resampled with a sample distance of $2k/2$, resulting in a constant ratio of scale over sample distance. The resulting "pyramid" represents the original $N \text{x} N$ pixels image with a sequence of $K = 2\log_2 N$ images at a geometric progressions of scales each with half the number of samples of the previous, resulting in a total of $2 \text{x} N \text{x} N$ samples.

The Gaussian pyramid for an $M = N \mathrm{x} N$ pixels image can be computed in $O(M)$ operations using cascade convolution with resampling [3]. This algorithm involves alternatively convolving with a Gaussian support, and resampling the resulting image with a sample distance of $\sqrt{2}$. The effect of cascade convolution is to sum the variances of the filters, so that the cumulative variance is $\sigma_k^2 = 2^k$ and the resulting standard deviation is $\sigma_k = 2^{k/2}$. Interleaving resampling with convolutions decreases the number of image samples while expanding the distance between samples. This has the effect of dilating the Gaussian support without increasing the number of samples used for the Gaussian, effectively increasing the scale. Aliasing is avoided (or minimized) by the fact that the image has been low-pass filtered by previous convolutions. The result is an algorithm with linear algorithmic complexity (i.e. $O(M)$) producing a discrete representation of scale space with $2M$ total samples. The image is initially convolved with a filter of $\sigma_0 = 1$ to produce an initial image $P(x,y,0)$.

$$k = 0 : P(x,y,0) = P(x,y) * G(x,y,1) \tag{7}$$

Where " $*$ " is the convolution operator. The pyramid image $(k = 1)$ is produced by a convolution with the same low pass filter, resulting in a cumulative scale factor of $\sigma_1^2 = 4$ giving $\sigma_1 = 2$.

$$P(x,y,\sigma) = S_{\sqrt{2}^k}\{P(x,y,k-1)\} * E_{\sqrt{2}^k}\{G(x,y,1)\} \tag{8}$$

Where $S_{\sqrt{2}}\{\}$ is the "diagonal" resampling operator and $E_{\sqrt{2}^k}\{\}$ is an diagonal expansion operator defined as:

$$S_{\sqrt{2}^K}\{P(x,y)\} = \begin{cases} P(x,y) & \text{if } (x+y)^2 Mod(2^{k-1}) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$E_{\sqrt{2}^K}\{G(x,y)\} = \begin{cases} G(\frac{x+y}{2^{K/2}}, \frac{x-y}{2^{K/2}}) & \text{if } (x+y)^2 Mod(2^{k-1}) \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

The $k = 0$ image may be discarded or used for estimating a Laplacian image for $k = 1$ if required. Because the $k = 1$ image has been smoothed with a Gaussian low-pass filter of scale $\sigma = 2$, resampling with a sample distance of $\sqrt{2}$ will result in an aliasing of less than 1% of signal energy. Gaussian derivatives are easily calculated in the row and column directions from these images by differences of adjacent pixels. Assuming that the space between samples at level k is $s = 2^{(k-1)/2}$ pixels, the image differences are:

$$\frac{\partial P(x,y,k)}{\partial x} = P * G_x(x,y,2^{k/2}) \approx P(x+s,y,k) - P(x-s,y,k) \tag{11}$$

$$\frac{\partial P(x,y,k)}{\partial y} = P * G_y(x,y,2^{k/2}) \approx P(x,y+s,k) - P(x,y-s,k) \tag{12}$$

$$\frac{\partial^2 P(x,y,k)}{\partial^2 x} = P * G_{xx}(x,y,2^{k/2}) \approx P(x+s,y,k) - 2P(x,y,k) + P(x-s,y,k) \tag{13}$$

$$\frac{\partial^2 P(x,y,k)}{\partial^2 y} = P * G_{yy}(x,y,2^{k/2}) \approx P(x,y+s,k) - 2P(x,y,k) + P(x,y-s,k) \tag{14}$$

$$\frac{\partial^2 P(x,y,k)}{\partial^2 xy} = P * G_{xy}(x,y,2^{k/2})$$

$$\approx P(x+s,y+s,k) - P(x+s,y-s,k) - P(x-s,y+s,k) + P(x-s,y-s,k)$$
(15)

Derivatives along diagonal directions can be computed using diagonal differences. Freeman and Adelson [6] have shown how a basis set of Gaussian derivatives can be "steered" to a desired orientation by weighting the derivative terms with the appropriate sine and cosines terms. Using their technique it is relatively easy to define a steerable basis for the Gaussian jet. If needed, Gaussian derivative features may easily be determined for image positions between image samples using bilinear interpolation, while Gaussian derivative features for scale values between the powers or root 2 can be computed using quadratic interpolation between adjacent levels in the pyramid. However, our experience is that the information encoded by such interpolated features is redundant with the existing features and does not contribute to object recognition.

3 Census Transform of Gaussian Features Maps

Gaussian derivatives features of first and second order capture information about edges, bars and blobs [23]. In our approach Gaussian feature maps are obtained by computing Gaussian derivative features of first and second order at four different orientations $0, \pi/4, \pi/2$ and $3\pi/2$ at 6 levels from the half-octave pyramid, corresponding to $\sigma = \sqrt{2}, 2, 2\sqrt{2}, 4, 4\sqrt{2}, 8$. These values have been determined by experimentation. We used also the feature maps from the zeroth order Gaussian pyramid. This descriptor provide a textural description that complements the information given by Gaussian Derivatives Features.

The total number of Gaussian maps calculated in our approach is 66. For Gaussian derivatives features of first order we compute 24 Gaussian maps (4 orientations and 6 levels), for Gaussian derivatives features of second order we compute 30 Gaussian maps (4 orientations, 6 levels plus the maps of G_{xy}) and finally we use 6 Gaussian maps resultant from original scale space in the first six levels of the pyramid.

For each Gaussian map, we apply a Census Transform Operator (CT) [24]. This operator compares the intensity values of a pixel with their eight neighboring pixels to obtain an 8-binary string, which is converted in to a decimal number between 0 and 255.

The Census Transform operator is robust to illumination changes and gamma variations. In the case of Gaussian features maps, the Census Transform captures information about local discontinuities due to edges, bars and blobs and reinforces the textural information encoded in the scale space and the second order Gaussian Jet Norm.

4 Tensors of Census Transform Histograms

A Census Transform encodes information about local appearance using Gaussian feature maps. For face recognition, we also need to capture information about the spatial distribution of features. A number of possible solutions exist such an encoding [20, 25, 26]. One problem with most approaches is that they do not conserve spatial structure but rather concatenate the histograms to form one unique vector. To avoid this problem, we propose

a Tensors of Census Transform Histograms. Such tensors encode the spatial information within the images.

We separate the Census Transform maps by type of information in the following order: first by Gaussian derivative features of first order, second by Gaussian derivatives Features of second order, third by the original Gaussian scale space. This separation is used to conserve the individual discriminative power of each representation. We divide each CT-Gaussian Feature map into non-overlapping rectangular sub-regions with specific size and a set of histograms is computed for each sub-region. These histograms encode the most relevant textural and spatial information in manner that is robust to illumination changes. In previous work, such histograms were calculated and concatenated to form a single vector [18, 20, 25]. The inconvenience of this approach is the large size of the final vector as well as the loss of information due to concatenation in only one single vector as each histogram has been calculated in a specific position, for a specific level resulting in a loss of the natural spatial 3-D organization. To avoid this, we propose to organize the histograms in a Tensor, thereby conserving the 3-D structure of Census Transform Gaussian maps.

We show an example of such a Tensor in the figure 1 (A). In our approach we calculated a 4-D tensor for each kind of Census Transform Gaussian feature, one for the Gaussian derivatives features of first order $\mathbf{T_1}$, another for the Gaussian derivatives features of second order $\mathbf{T_2}$ and the last is calculated with the Gaussian scale space $\mathbf{T_G}$. As a result we obtain 3 tensors.

$$\mathbf{T_1} \in \Re^{N_\theta x N_{bins} x N_x x N_{ins}} \tag{16}$$

$$\mathbf{T_2} \in \Re^{N_{\theta \cap G_{xy}} x N_{bins} x N_x x N_{ins}} \tag{17}$$

$$\mathbf{T_G} \in \Re^{N_{feat} x N_{bins} x N_x x N_{ins}} \tag{18}$$

Where N_θ is the number of orientations, N_x is the number of positions in which one histogram has been calculated, N_{bins} is the number of bins used in each histogram, $N_{\theta \cap G_{xy}}$ are the same orientations of N_θ plus the mixed orientation of the census transform map from G_{xy}, N_{feat} is census features maps for G_0 and N_{ins} are the face images used to calculate the CT Gaussian features maps.

4.1   Fusing CT Tensors Histograms with MPCA

In a recent paper, Wu and Rehg [20] experimentally demonstrated the correlation in neighboring pixels for Census Transform histograms. This correlation can be interpreted as redundant information. Wu and Rehg proposed to eliminate this redundant information using Principal Component Analysis of each histogram.

In a similar manner, our tensorial representation has a large redundancy as may be seen in the correlation in both the index N_{bins} and also in the indexes in each tensor. To reduce this correlation while conserving the spatial distribution of tensors, we propose the use of Multilinear Principal Component Analysis (MPCA) as suggested by Lu et al [12]. Their method determines a multilinear projection that captures most of the original tensorial input variation and supplants existing heterogeneous solutions such as the classical PCA and its 2-D variant 2-D PCA.

We describe MPCA similar to [12] and [15]: T is an n'th order tensor and $\mathbf{T} \in \Re^{N_1 x N_2 x ... N_n}$, the MPCA is defined as the application of n-mode Single Value Decomposition.
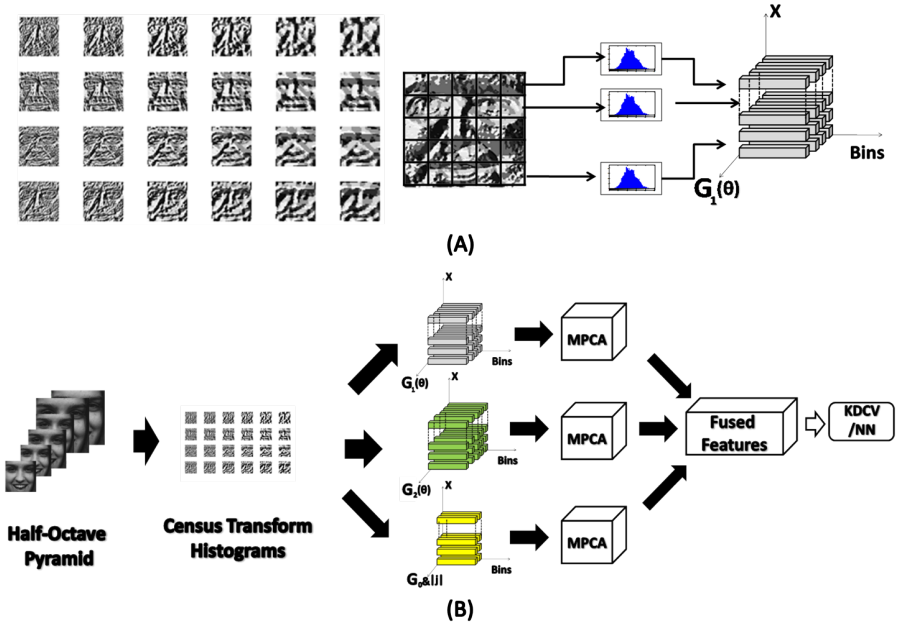
Figure 1: An example of the construction for a tensor in our method (A), Overall architecture of our face recognition system (B)

This orthogonalizes the "n" associated vector spaces of and decomposes the tensor as:

$$\mathbf{T} = \mathbf{S}\mathbf{x_1}\mathbf{U^1}\mathbf{x_2}...\mathbf{x_k}\mathbf{U^K}...\mathbf{x_n}\mathbf{U^n} \tag{19}$$

Where,$\mathbf{x_k}$ denotes mode$-k$ product and $\mathbf{U^k}$ the orthonormal matrix that contains ordered principal components for the kth mode. $\mathbf{S}$ is called the core tensor of projection tensor.

In our approach we apply MPCA to each tensor $\mathbf{T_1}$, $\mathbf{T_2}$ and $\mathbf{T_G}$. The resultant vector of features for each tensor are: $\mathbf{v_1}$, $\mathbf{v_2}$ and $\mathbf{v_3}$ respectively. We retain the first m_k components from each vector. Generally the number of retained components in a PCA approach without lost of information is:

$$m_k \leq (ins, dim(v_k)) \tag{20}$$

Where *ins* is the number of face images used to compute the tensors and $dim(v_k)$ is the dimension of each resultant vector. We then combine all of the resultant vectors to form a single normalized feature vector.

$$v_{test} = cat(v_1, v_2, v_3) \quad v_{test} \in \mathfrak{R}^{\Sigma m_k} \tag{21}$$

We have found that the discriminative power of each final vector v_{test} can be improved by projection onto an optimal discriminative space with a KDCV [7] (Kernel Discriminative Common Vector). This kernel method has been proved with success in [18]. The characteristic equation for the KDCV is:

$$l_{test} = \left(U\Lambda^{1/2}V\right)^T k_{test} \quad l_{test} \in \mathfrak{R}^t \tag{22}$$

Where Λ is the diagonal matrix with non-zero eigenvalues, $\mathbf{U}$, the associated matrix of normalized eigenvectors, $\mathbf{V}$ is the basis for the null space of the projected within-class scatter matrix and l_{test} is a vector with entries $K(v_i^t, v_{test}) = \langle \phi(v_i^t), \phi(v_{test}) \rangle$, where $\phi(v_i^t)$ are the mapped training samples in a high dimensional space and $K()$ is a typical kernel, for details see [2, 13].

The projected l_{test} is classified using the nearest neighbor rule and the cosine distance.

$$d_{cos}(l_{test}, l_{temp}) = -\frac{l_{test}^T l_{temp}}{\|l_{test}\| \|l_{temp}\|} \tag{23}$$

Figure 1 (B), summarizes the overall architecture of our face recognition system that has been described above.

5   Experimental Evaluations

In this section, we present the results of experiments designed to illustrate the effectiveness of the proposed method. We use three publicly available face databases: The FERET [14] Yale and the Yale B + Extended Yale B [7] face datasets. In all of our experiments, the images were cropped using manually located eyes positions and normalized at size of 128x128. No attempt is made to normalize for illumination. To calculate Histograms of CT Gaussian maps, we used a sub-region size of 16x16, removing two bins corresponding to CT = 0, 255 per each histogram. The others bins are grouped to form a 127 bin histogram. We used a Matlab® implementation of the MPCA algorithm provided by [12]. For each tensor, the mean tensor has been subtracted and the final concatenated vector has been normalized to a unit standard deviation. For the KDCV algorithm we used Gaussian kernels $k(x,y) = e^{-\|x-y\|^2/2\sigma^2}$ with scale parameter ó chosen experimentally to obtain the best results.

5.1   Experiments with the FERET dataset

To test the robustness of our approach against variations in facial expression, lighting and aging, we evaluated the proposed method on all four FERET probe sets: the 'fb' set contains faces with variation in expression, the 'fc' set with lighting variation and the 'dup I' and 'dupII' sets contain variation dues to aging of the subject. We used the FERET distributed training set plus the gallery images 'fa' to train the KDCV. As shown in [13], the addition of the gallery increases the dimensionality of the final discriminative space. We have compared our results with the best results in the FERET'97 test [14] and the published results in [13, 18, 25, 26]. The rank-1 recognition rates of different methods on the FERET probe sets are show in Table 1. To our knowledge, the results in [18] and [13] are the most recent state-of-the-art with the FERET database.

Our results with the FERET database are statistically equivalent (with a difference of ± 0.01) or better to the most recent state-of-the-art results on this dataset. Note that most of the methods described in [25] [26] and [13] use the Gabor wavelets to generate their maps. These wavelets have a much higher algorithmic complexity and overall computing cost that is not improvable. On the other hand, our Gaussian derivatives features calculated with the half-octave pyramid to generate feature maps [5], can be provided with a fast linear complexity algorithm, and are thus much more suitable for real applications.

Table 1: The Rank-1 Recognition Rates of Different Algorithms on the FERET Probe Sets

Method	FERET Probe Sets			
	fb	fc	Dup1	Dup1
Best Results [14]	0.96	0.82	0.59	0.52
LGBPHS_Weighted [26]	0.98	0.97	0.74	0.71
HGPP_Weighted [25]	0.97	0.99	0.80	0.78
Gabor+LBP_KDCVM [18]	0.98	0.98	0.90	0.85
GRM-Local [13]	0.98	0.98	0.80	0.84
Our Method	**0.98**	**0.98**	**0.90**	**0.85**

Table 2: The Rank-1 Recognition Rates of Diferent Algorithms on The YALE Face Dataset

Method	Recognition Accuracy
Eigenfaces [0]	76.0%
2DPCA [22]	84.2%
Fisherfaces [0]	96.0%
DCVM [0]	97.3%
Our Method	**98.2%**

5.2 Experiments on the YALE database

To evaluate robustness to variations in expression we used the YALE face database, which contains 165 images of 15 individuals under various facial expressions (with glasses, happy, without glasses, normal, sad, sleepy, surprised and wink) and lighting conditions (center, right and left light). Images with a neutral facial expression are used as gallery set. We augment the gallery set with images without glasses to train the Kernel DCV. The remaining images are used as a probe set to compare our results with the results presented in [0, 22]. The rank-1 recognition rates of different methods on the YALE dataset are shown in the Table 2. Clearly the results of the proposed method are better than the best results reported in [0] and [22], demonstrating reliability of our method under changes in facial expression.

5.3 Experiments on the Yale B + extended Yale B database

The Yale face dataset B contains images obtained from 10 individuals. Images are captured under 64 different lighting conditions from 9 pose views and are divided into 5 subsets according to the ranges of the illumination angles between the light source direction and the camera axis. The extended Yale-B dataset contains 16128 images of 28 human subjects captured under the same condition as Yale B. In our experiments, we use only the frontal face images from these two databases. The images with the most neutral illumination A+000E+00 are used as gallery set. To train the Kernel DCV we take the images in the gallery set and a randomly selected image from the first subset. We did not test our method with the challenging subset 5. The rank-1 recognition rates of different methods on the Yale B + extended Yale B dataset are show in the Table 3. In this case we outperformed the published results in [21] for the subset number 3 and we have the same results for the subsets 1 and 2. Note that while the results reported in [21] are for different methods of illumination normalization, we do not normalize the images for changes in illumination and our method does not try to specifically solve this problem.

Table 3:   The Rank-1 Recognition Rates of Different Algorithms on The YALE B+EXTENDED YALE-B Face Dataset

	Subset Number			
Method	**1**	**2**	**3**	**4**
LTV [21]	100.0%	99.8%	79.4%	76.1%
RLS(LOG-DTC) [21]	100.0%	100.0%	87.1%	87.6%
Our Method	**100.0%**	**100.0%**	**94.7%**	**60.1%**

6   Conclusion and Discussion

This paper presents a new algorithm for face recognition that uses the Census Transform histograms from Gaussian features calculated with a half octave Gaussian pyramid. This method does not require any correction for variation of illumination, and yet is robust to illumination changes. The method uses Gaussian derivatives features obtained with a fast, linear complexity pyramid algorithm, and yet obtains results that are comparable or better than more computationally expensive methods using Gabor Wavelets. We avoid the problems of loss of spatial information by using a tensorial representation for the local histograms calculated from the CT Gaussian maps. Finally Multilinear Principal Component Analysis and the Kernel Discriminative common vectors are used to lower the information correlation and improve the recognition accuracy. Experimental evaluations of the proposed approach on the FERET, Yale and Yale B + extended Yale B datasets showed the effectiveness and robustness of CT histograms from local Gaussian features to the general variations of lighting and expression.

References

[1] H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana. Discriminative common vectors for face recognition. *IEEE PAMI.*, 27(1):4–13, 2005.

[2] H. Cevikalp, M. Neamtu, and M. Wilkes. Discriminative common vector method with kernels. *IEEE TNN*, 17(6):1550–1565, 2006.

[3] J. L. Crowley and R.M. Stern. Fast computation of the difference of low-pass transform. *IEEE PAMI*, 6(2):212–222, March 1984.

[4] J.L. Crowley and A.C. Parker. A representation for shape based on peaks and ridges in thedifference of low-pass transform. *IEEE PAMI*, 6(2):156–169, March 1984.

[5] J.L. Crowley and O. Riff. Fast computation of scale normalised gaussian receptive fields. In *Proc. Scale Space Methods in Computer Vision*, pages 584–598, 2003.

[6] W. T. Freeman and E. H. Adelson Y. The design and use of steerable filters. *IEEE PAMI*, 13: 891–906, 1991.

[7] Athinodoros S. Georghiades, Peter N. Belhumeur, and David J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE PAMI*, 23 (6):643–660, 2001.

[8] D. Hall. Viewpoint independent recognition of objects from local appearance. *Phd Thesis, Institute National Polytecnique de Grenoble, Grenoble*, 2004.

[9] J.J Koenderink and A.J van Doom. Representation of local geometry in the visual system. *Biol. Cybern.*, 55(6):367–375, 1987. ISSN 0340-1200.

[10] T. Lindeberg. Scale-space theory in computer vision. *Kluwer Academic Publishers, Norwell, MA, USA*, 1994.

[11] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004.

[12] Haiping Lu, Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. Mpca: Multilinear principal component analysis of tensor objects. *IEEE TNN*, 19(1):18–39, 2008.

[13] Yui Man Lui and J. Ross Beveridge. Grassmann registration manifolds for face recognition. In *ECCV2008*, pages 44–57, 2008.

[14] P. J. Phillips, H. Moon, Syed A. Rizvi, and Patrick J. Rauss. The feret evaluation methodology for face-recognition algorithms. *IEEE PAMI*, 22(10):1090–1104, 2000.

[15] S. Rana, Wanquan Liu, M. Lazarescu, and S. Venkatesh. Recognising faces in unseen modes: A tensor based approach. *IEEE CVPR2008.*, pages 1–8, June 2008.

[16] Bart M. ter Haar Romeny, Luc Florack, Alfons H. Salden, and Max A. Viergever. Higher order differential structure of images. In *IPMI '93: Proceedings of the 13th International Conference on Information Processing in Medical Imaging*, pages 77–93, London, UK, 1993. Springer-Verlag.

[17] B. Schiele and J.L. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36:31–50, 2000.

[18] X. Tan and B. Triggs. Fusing gabor and lbp feature sets for kernel-based face recognition. In *Analysis and Modelling of Faces and Gestures*, pages 235–249, oct 2007.

[19] Engin Tola, Vincent Lepetit, and Pascal Fua. Daisy: An efficient dense descriptor applied to wide baseline stereo. *IEEE PAMI*, 99(1), 2009. ISSN 0162-8828.

[20] Jianixn W. and J.M. Rehg. Where am i: Place instance and category recognition using spatial pact. *IEEE CVPR2008.*, pages 1–8, June 2008.

[21] Xiaohua Xie, Wei-Shi Zheng, Jianhuang Lai, and P.C. Yuen. Face illumination normalization on large and small scale features. *IEEE CVPR2008.*, pages 1–8, June 2008.

[22] Jian Yang, David Zhang, Alejandro F. Frangi, and Jing yu Yang. Two-dimensional pca: A new approach to appearance-based face representation and recognition. *IEEE PAMI*, 26(1):131–137, 2004.

[23] J.J. Yokono and T. Poggio. Oriented filters for object recognition: an empirical study. In *Proceedings of the IEEE FG2004. Seoul, Korea*, page 755, 2004.

[24] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In *ECCV '94*, pages 151–158, 1994.

[25] B. Zhang, S. Shan, X. Chen, and W. Gao. Histogram of gabor phase patterns (hgpp): A novel object representation approach for face recognition. *IEEE TIP*, 16(1):57–68, 2007.

[26] W. Zhang, S. Shan, W. Gao, X. Chen, and H. Zhang. Local gabor binary pattern histogram sequence (lgbphs): A novel non-statistical model for face representation and recognition. *IEEE ICCV,*, 1:786–791, 2005.

[27] W. Zhao, R. Chellappa, A. Rosenfeld, and P. J. Phillips. Face recognition: A literature survey. *ACM Computing Surveys*, pages 399–458, 2003.