Region Covariance: A Fast Descriptor for Detection and Classification

Oncel Tuzel[1,3], Fatih Porikli[3], and Peter Meer[1,2]

[1] Computer Science Department,
[2] Electrical and Computer Engineering Department,
Rutgers University, Piscataway, NJ 08854
{otuzel, meer}@caip.rutgers.edu
[3] Mitsubishi Electric Research Laboratories,
Cambridge, MA 02139
fatih@merl.com

Abstract. We describe a new region descriptor and apply it to two problems, object detection and texture classification. The covariance of d-features, e.g., the three-dimensional color vector, the norm of first and second derivatives of intensity with respect to x and y, etc., characterizes a region of interest. We describe a fast method for computation of covariances based on *integral images*. The idea presented here is more general than the image sums or histograms, which were already published before, and with a series of integral images the covariances are obtained by a few arithmetic operations. Covariance matrices do not lie on Euclidean space, therefore we use a distance metric involving generalized eigenvalues which also follows from the Lie group structure of positive definite matrices. Feature matching is a simple nearest neighbor search under the distance metric and performed extremely rapidly using the integral images. The performance of the covariance features is superior to other methods, as it is shown, and large rotations and illumination changes are also absorbed by the covariance matrix.

1 Introduction

Feature selection is one of the most important steps for detection and classification problems. Good features should be discriminative, robust, easy to compute and efficient algorithms are needed for a variety of tasks such as recognition and tracking.

The raw pixel values of several image statistics such as color, gradient and filter responses are the simplest choice for image features, and were used for many years in computer vision, e.g., [1, 2, 3]. However, these features are not robust in the presence of illumination changes and nonrigid motion, and efficient matching algorithms are limited by the high dimensional representation. Lower dimensional projections were also used for classification [4] and tracking [5].

A natural extension of raw pixel values are via histograms where a region is represented with its nonparametric estimation of joint distribution. Following [6], histograms were widely used for nonrigid object tracking. In a recent study [7],

© Springer-Verlag Berlin Heidelberg 2006

fast histogram construction methods were explored to find a global match. Besides tracking, histograms were also used for texture representation [8, 9], matching [10] and other problems in the field of computer vision. However, the joint representation of several different features through histograms is exponential with the number features.

The *integral image* idea is first introduced in [11] for fast computation of Haar-like features. Combined with cascaded AdaBoost classifier, superior performances were reported for face detection problem, but the algorithm requires long training time to learn the object classifiers. In [12] scale space extremas are detected for keypoint localization and arrays of orientation histograms were used as keypoint descriptors. The descriptors are very effective in matching local neighborhoods but do not have global context information.

There are two main contributions within this paper. First, we propose to use the covariance of several image statistics computed inside a region of interest, as the region descriptor. Instead of the joint distribution of the image statistics, we use the covariance as our feature, so the dimensionality is much smaller. We provide a fast way of calculating covariances using the integral images and the computational cost is independent of the size of the region. Secondly, we introduce new algorithms for object detection and texture classification using the covariance features. The covariance matrices are not elements of the Euclidean space, therefore we can not use most of the classical machine learning algorithms. We propose a nearest neighbor search algorithm using a distance metric defined on the positive definite symmetric matrices for feature matching.

In Section 2 we describe the covariance features and explain the fast computation of the region covariances using integral image idea. Object detection problem is described in Section 3 and texture classification problem is described in Section 4. We demonstrate the superior performance of the algorithms based on the covariance features with detailed comparisons to previous methods and features.

2 Covariance as a Region Descriptor

Let I be a one dimensional intensity or three dimensional color image. The method also generalizes to other type of images, e.g., infrared. Let F be the $W \times H \times d$ dimensional feature image extracted from I

$$F(x, y) = \phi(I, x, y) \tag{1}$$

where the function ϕ can be any mapping such as intensity, color, gradients, filter responses, etc. For a given rectangular region $R \subset F$, let $\{\mathbf{z}_k\}_{k=1..n}$ be the d-dimensional feature points inside R. We represent the region R with the $d \times d$ covariance matrix of the feature points

$$\mathbf{C}_R = \frac{1}{n-1} \sum_{k=1}^{n} (\mathbf{z}_k - \boldsymbol{\mu})(\mathbf{z}_k - \boldsymbol{\mu})^T \tag{2}$$

where $\boldsymbol{\mu}$ is the mean of the points.

There are several advantages of using covariance matrices as region descriptors. A single covariance matrix extracted from a region is usually enough to match the region in different views and poses. In fact we assume that the covariance of a distribution is enough to discriminate it from other distributions. If two distributions only vary with their mean, our matching result produces perfect match but in real examples these cases almost never occur.

The covariance matrix proposes a natural way of fusing multiple features which might be correlated. The diagonal entries of the covariance matrix represent the variance of each feature and the nondiagonal entries represent the correlations. The noise corrupting individual samples are largely filtered out with an average filter during covariance computation.

The covariance matrices are low-dimensional compared to other region descriptors and due to symmetry $\mathbf{C}_R$ has only $(d^2+d)/2$ different values. Whereas if we represent the same region with raw values we need $n \times d$ dimensions, and if we use joint feature histograms we need b^d dimensions, where b is the number of histogram bins used for each feature.

Given a region R, its covariance $\mathbf{C}_R$ does not have any information regarding the ordering and the number of points. This implies a certain scale and rotation invariance over the regions in different images. Nevertheless, if information regarding the orientation of the points are represented, such as the norm of gradient with respect to x and y, the covariance descriptor is no longer rotationally invariant. The same argument is also correct for scale and illumination. Rotation and illumination dependent statistics are important for recognition/classification purposes and we use them in Sections 3 and 4.

2.1 Distance Calculation on Covariance Matrices

The covariance matrices do not lie on Euclidean space. For example, the space is not closed under multiplication with negative scalers. Most of the common machine learning methods work on Euclidean spaces and therefore they are not suitable for our features. The nearest neighbor algorithm which will be used in the following sections, only requires a way of computing distances between feature points. We use the distance measure proposed in [13] to measure the dissimilarity of two covariance matrices

$$\rho(\mathbf{C}_1, \mathbf{C}_2) = \sqrt{\sum_{i=1}^{n} \ln^2 \lambda_i(\mathbf{C}_1, \mathbf{C}_2)} \tag{3}$$

where $\{\lambda_i(\mathbf{C}_1, \mathbf{C}_2)\}_{i=1...n}$ are the generalized eigenvalues of $\mathbf{C}_1$ and $\mathbf{C}_2$, computed from

$$\lambda_i \mathbf{C}_1 \mathbf{x}_i - \mathbf{C}_2 \mathbf{x}_i = 0 \qquad i = 1...d \tag{4}$$

and $\mathbf{x}_i \neq 0$ are the generalized eigenvectors. The distance measure ρ satisfies the metric axioms for positive definite symmetric matrices $\mathbf{C}_1$ and $\mathbf{C}_2$

1. $\rho(\mathbf{C}_1, \mathbf{C}_2) \geq 0$ and $\rho(\mathbf{C}_1, \mathbf{C}_2) = 0$ only if $\mathbf{C}_1 = \mathbf{C}_2$,
2. $\rho(\mathbf{C}_1, \mathbf{C}_2) = \rho(\mathbf{C}_2, \mathbf{C}_1)$,
3. $\rho(\mathbf{C}_1, \mathbf{C}_2) + \rho(\mathbf{C}_1, \mathbf{C}_3) \geq \rho(\mathbf{C}_2, \mathbf{C}_3)$.

The distance measure also follows from the Lie group structure of positive definite matrices and an equivalent form can be derived from the Lie algebra of positive definite matrices. The generalized eigenvalues can be computed with $O(d^3)$ arithmetic operations using numerical methods and an additional d logarithm operations are required for distance computation, which is usually faster than comparing two histograms that grow exponentially with d. We refer the readers to [13] for a detailed discussion on the distance metric.

2.2 Integral Images for Fast Covariance Computation

Integral images are intermediate image representations used for fast calculation of region sums [11]. Each pixel of the integral image is the sum of all the pixels inside the rectangle bounded by the upper left corner of the image and the pixel of interest. For an intensity image I its integral image is defined as

$$\text{Integral Image } (x', y') = \sum_{x<x', y<y'} I(x, y). \tag{5}$$

Using this representation, any rectangular region sum can be computed in constant time. In [7], the integral images were extended to higher dimensions for fast calculation of region histograms. Here we follow a similar idea for fast calculation of region covariances.

We can write the (i, j)-th element of the covariance matrix defined in (2) as

$$C_R(i, j) = \frac{1}{n-1} \sum_{k=1}^{n} (z_k(i) - \mu(i))(z_k(j) - \mu(j)). \tag{6}$$

Expanding the mean and rearranging the terms we can write

$$C_R(i, j) = \frac{1}{n-1} \left[\sum_{k=1}^{n} z_k(i) z_k(j) - \frac{1}{n} \sum_{k=1}^{n} z_k(i) \sum_{k=1}^{n} z_k(j) \right]. \tag{7}$$

To find the covariance in a given rectangular region R, we have to compute the sum of each feature dimension, $z(i)_{i=1...n}$, as well as the sum of the multiplication of any two feature dimensions, $z(i)z(j)_{i,j=1...n}$. We construct $d + d^2$ integral images for each feature dimension $z(i)$ and multiplication of any two feature dimensions $z(i)z(j)$.

Let P be the $W \times H \times d$ tensor of the integral images

$$P(x', y', i) = \sum_{x<x', y<y'} F(x, y, i) \qquad i = 1...d \tag{8}$$

and Q be the $W \times H \times d \times d$ tensor of the second order integral images

$$Q(x', y', i, j) = \sum_{x<x', y<y'} F(x, y, i) F(x, y, j) \qquad i, j = 1...d. \tag{9}$$

In [11], it is shown that integral image can be computed in one pass over the image. In our notation, $\mathbf{p}_{x,y}$ is the d dimensional vector and $\mathbf{Q}_{x,y}$ is the $d \times d$ dimensional matrix

$$\mathbf{p}_{x,y} = [P(x,y,1)\dots P(x,y,d)]^T$$
$$\mathbf{Q}_{x,y} = \begin{pmatrix} Q(x,y,1,1) \dots Q(x,y,1,d) \\ \vdots \\ Q(x,y,d,1) \dots Q(x,y,d,d) \end{pmatrix}. \tag{10}$$

Note that $\mathbf{Q}_{x,y}$ is a symmetric matrix and $d + (d^2 + d)/2$ passes are enough to compute both P and Q. The computational complexity of constructing the integral images is $O(d^2WH)$.

Let $R(x', y'; x'', y'')$ be the rectangular region, where (x', y') is the upper left coordinate and (x'', y'') is the lower right coordinate, as shown in Figure 1. The covariance of the region bounded by $(1, 1)$ and (x', y') is

$$\mathbf{C}_{R(1,1;x',y')} = \frac{1}{n-1}\left[\mathbf{Q}_{x',y'} - \frac{1}{n}\mathbf{p}_{x',y'}\mathbf{p}^T_{x',y'}\right] \tag{11}$$

where $n = x' \cdot y'$. Similarly, after a few manipulations, the covariance of the region $R(x', y'; x'', y'')$ can be computed as

$$\mathbf{C}_{R(x',y';x'',y'')} = \frac{1}{n-1}\Big[\mathbf{Q}_{x'',y''} + \mathbf{Q}_{x',y'} - \mathbf{Q}_{x'',y'} - \mathbf{Q}_{x',y''} \tag{12}$$
$$-\frac{1}{n}\left(\mathbf{p}_{x'',y''} + \mathbf{p}_{x',y'} - \mathbf{p}_{x',y''} - \mathbf{p}_{x'',y'}\right)\left(\mathbf{p}_{x'',y''} + \mathbf{p}_{x',y'} - \mathbf{p}_{x',y''} - \mathbf{p}_{x'',y'}\right)^T\Big]$$

where $n = (x'' - x') \cdot (y'' - y')$. Therefore, after constructing integral images the covariance of any rectangular region can be computed in $O(d^2)$ time.

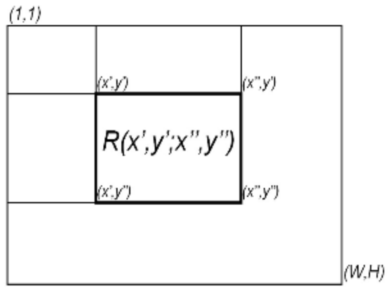

Fig. 1. Integral Image. The rectangle $R(x', y'; x'', y'')$ is defined by its upper left (x', y') and lower right (x'', y'') corners in the image, and each point is a d dimensional vector.

3 Object Detection

In object detection, given an object image, the aim is to locate the object in an arbitrary image and pose after a nonrigid transformation. We use pixel locations (x,y), color (RGB) values and the norm of the first and second order derivatives of the intensities with respect to x and y. Each pixel of the image is converted to a nine-dimensional feature vector

$$F(x,y) = \left[x \quad y \quad R(x,y) \quad G(x,y) \quad B(x,y) \right.$$
$$\left. \left|\frac{\partial I(x,y)}{\partial x}\right| \left|\frac{\partial I(x,y)}{\partial y}\right| \left|\frac{\partial^2 I(x,y)}{\partial x^2}\right| \left|\frac{\partial^2 I(x,y)}{\partial y^2}\right| \right]^T \qquad (13)$$

where R, G, B are the RGB color values, and I is the intensity. The image derivatives are calculated through the filters $[-1\ 0\ 1]^T$ and $[-1\ 2\ -1]^T$. The covariance of a region is a 9×9 matrix. Although the variance of pixel locations (x,y) is same for all the regions of the same size, they are still important since their correlation with the other features are used at the nondiagonal entries of the covariance matrix.

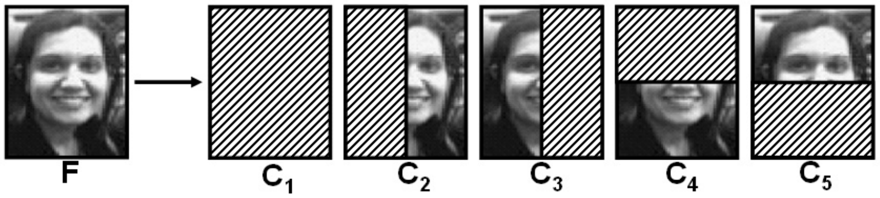

Fig. 2. Object representation. We construct five covariance matrices from overlapping regions of an object feature image. The covariances are used as the object descriptors.

We represent an object with five covariance matrices of the image features computed inside the object region, as shown in Figure 2. Initially we compute only the covariance of the whole region, $\mathbf{C}_1$, from the source image. We search the target image for a region having similar covariance matrix and the dissimilarity is measured through (3). At all the locations in the target image we analyze at nine different scales (four smaller, four larger) to find matching regions. We perform a brute force search, since we can compute the covariance of an arbitrary region very quickly. Instead of scaling the target image, we just change the size of our search window. There is a 15% scaling factor between two consecutive scales. The variance of the x and y components are not the same for regions with different sizes and we normalize the rows and columns corresponding to these features. At the smallest size of the window we jump three pixels horizontally or vertically between two search locations. For larger windows we jump 15% more and round to the next integer at each scale.

We keep the best matching 1000 locations and scales. At the second phase we repeat the search for 1000 detected locations, using the covariance matrices $\mathbf{C}_{i=1...5}$. The dissimilarity of the object model and a target region is computed

$$\rho(O,T) = \min_j \left[\sum_{i=1}^{5} \rho(\mathbf{C}_i^O, \mathbf{C}_i^T) - \rho(\mathbf{C}_j^O, \mathbf{C}_j^T) \right] \qquad (14)$$

where $\mathbf{C}_i^O$ and $\mathbf{C}_i^T$ are the object and target covariances respectively, and we ignore the least matching region covariance of the five. This increases robustness

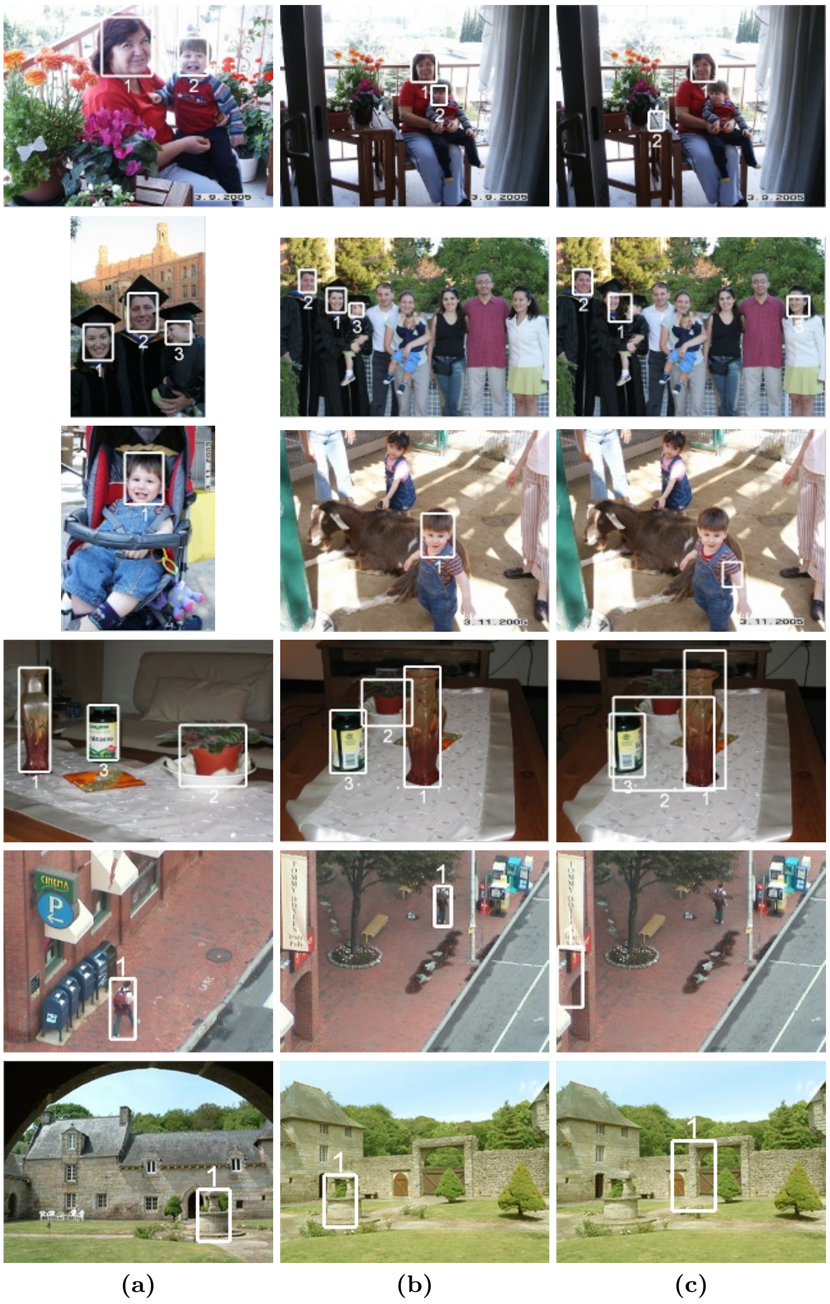

Fig. 3. Object detection. (a) Input regions. (b) Regions found via covariance features. (c) Regions found via histogram features.

towards possible occlusions and large illumination changes. The region with the smallest dissimilarity is selected as the matching region.

We present the matching results for a variety of examples in Figure 3 and compare our results with histogram features. We tested histogram features both with the RGB and HSV color spaces. With the RGB color space the results were much worse in all of the cases, therefore we did not present these results. We construct three separate 64 bin histograms for hue, saturation and value since it is not practical to construct a joint histogram. We search the target image for the same locations and sizes, and fast construction of histograms are performed through integral histograms [7]. We measure the distance between two histograms through Bhattacharyya distance [6] and sum over three color channels.

Covariance features can match all the target regions accurately whereas most of the regions found by histogram are erroneous. Even among the correctly detected regions with both methods we see that covariance features better localize the target. The examples are challenging since there are large scale, orientation and illumination changes, and some of the targets are occluded and have nonrigid motion. Almost perfect results indicate the robustness of the proposed approach. We also conclude that the covariances are very discriminative since they can match the correct target in the presence of similar objects, as seen in the face matching examples.

Covariance features are faster than the integral histograms since the dimensionality of the space is smaller. The search time of an object in a color image with size 320×240 is 6.5 seconds with a MATLAB 7 implementation. The performance can be improved by a factor of 20-30 with a C++ implementation which would yield to near real time performance.

4 Texture Classification

Currently, the most successful methods for texture classification are through textons which are cluster centers in a feature space derived from the input. The feature space is built from the output of a filter bank applied at every pixel and the methods differ only in the employed filter bank.

– **LM:** A combination of 48 anisotropic and isotropic filters were used by Leung and Malik [8]. The feature space is 48 dimensional.
– **S:** A set of 13 circular symmetric filters was used by Schmid [14]. The feature space is 13 dimensional.
– **M4**, **M8:** Both representations were proposed by Varma and Zissermann [9]. Original filters include both rotationally symmetric and oriented filters but only maximum response oriented filters are included to feature vector. The feature space is 4 and 8 dimensional respectively.

To find the textons, usually the k-means clustering algorithm is used, although it was shown that it might not be the best choice [15]. The most significant textons are aggregated into the texton library and the texton histograms are used as texture representation. The χ^2 distance [8] is used to measure the similarity of two histograms and the training image with the smallest distance from the test image determines the class of the latter. The process is computationally expensive since the images are convolved with large filter banks and in most cases requires clustering in high dimensional space.

4.1 Random Covariances for Texture Classification

We present a new approach to texture classification problem without using textons. We start with extracting several features from each pixel. For texture classification problem we use image intensities and norms of first and second order derivatives of intensities in both x and y direction. Each pixel is mapped to a $d = 5$ dimensional feature space

$$F(x,y) = \left[I(x,y) \left|\frac{\partial I(x,y)}{\partial x}\right| \left|\frac{\partial I(x,y)}{\partial y}\right| \left|\frac{\partial^2 I(x,y)}{\partial x^2}\right| \left|\frac{\partial^2 I(x,y)}{\partial y^2}\right| \right]^T . \qquad (15)$$

We sample s random square regions from each image with random sizes between 16×16 and 128×128. Using integral images we compute the covariance matrix of each region. Each texture image is then represented with s covariance matrices and we have u training texture images from each texture class, a total of $s \cdot u$ covariance matrices. Texture representation process is illustrated in Figure 4. We repeat the process for the c texture classes and construct the representation for each texture class in the same way.

Given a test image, we again extract s covariance matrices from randomly selected regions. For each covariance matrix we measure the distance (3) from all the matrices of the training set and the label is predicted according to the majority voting among the k nearest ones (kNN algorithm). This classifier performs as a weak classifier and the class of the texture is determined according to the maximum votes among the s weak classifiers.

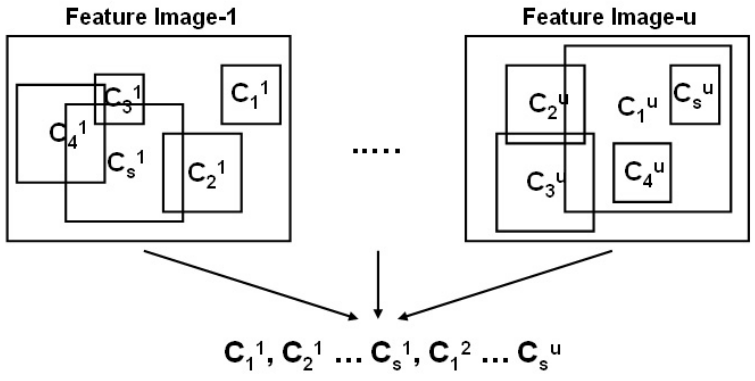

Fig. 4. Texture representation. There are u images for each texture class and we sample s regions from each image and compute covariance matrices **C**.

4.2 Texture Classification Experiments

We perform our tests on the Brodatz texture database which consists of 112 textures. Because of the nonhomogeneous textures inside the database, classification is a challenging task. We duplicate the test environment of [15]. Each 640×640 texture image is divided into four 320×320 subimages and half of the images are used for training and half for testing.

We compare our results with the results reported in [15] in Table 1. Here we present the results for k-means based clustering algorithm. The texture representation through texton histograms has 560 bins. The results vary from 85.71% to 97.32% depending on the filter bank used.

In our tests we sample $s = 100$ random covariances from each image, both for testing and training, and we used $k = 5$ for the kNN algorithm. For $d = 5$ dimensional features, the covariance matrix is 5×5 and has only 15 different values compared to 560 bins before. Our result, 97.77%, is better than all of the previous results and faster. Only 5 images out of 224 is misclassified which is close to the upper limit of the problem. We show two of the misclassified images in Figure 5 and the misclassification is usually in nonhomogeneous textures.

To make the method rotationally invariant, we used only three rotationally invariant features: intensity and the magnitude of the gradient and Laplacian. The covariance matrices are 3×3 and have only 6 different values. Even with this very simple features the classification performance is 94.20%, which is as good as or even better than other rotationally invariant methods (**M4**, **M8**, **S**) listed in Table 1. Due to random sized window selection our method is scale invariant. Although the approach is not completely illumination invariant, it is more robust than using features (intensity and gradients) directly. The variances of intensity and gradients inside regions change less than intensity and gradients themselves in illumination variations.

In the *second experiment* we compare the covariance features with other possible choices. We run the proposed texture classification algorithm with the raw intensity values and histograms extracted from random regions.

For raw intensities we normalize each random region to 16×16 square region and use Euclidean distance to compute distances for kNN classification, which is similar to [3]. The feature space is 256 dimensional. The raw intensity values are very noisy therefore only in this case we sample $s = 500$ regions from each image.

We perform two tests using histogram features: intensity only, and intensity and norms of first and second order derivatives together. In both cases the dissimilarity is measured with Bhattacharyya distance [6]. We use 256 bins for intensity only and $5 \cdot 64 = 320$ bins for intensity and norm of derivatives together. It is not practical to construct the joint intensity and norm of derivatives histograms, due to computational and memory requirement.

Table 1. Classification results for the Brodatz database

	M4	**M8**	**S**	**LM**	**Random Covariance**
Performance	85.71	94.64	93.30	97.32	97.77

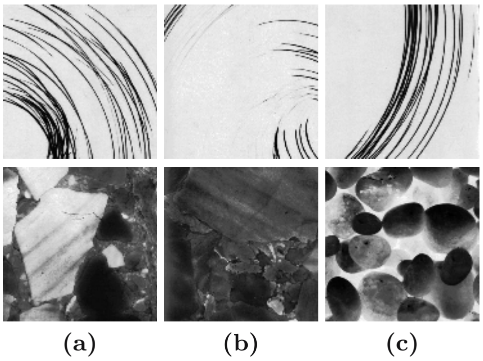

Fig. 5. Misclassified samples. (a) Test examples. (b) Samples from the same class. (c) Samples from the predicted texture class.

Table 2. Classification results for different features

	Raw Inten.	**Inten. Hist.**	**Inten./Deriv. Hist.**	**Covariance**
Performance	26.79	83.35	96.88	97.77

We sample $s = 100$ regions from each texture image. The results are shown in Table 2. The only result close to covariance is the 320 dimensional intensity and derivative histograms together. This is not surprising because our covariance features are the covariances of the joint distribution of the intensity and derivatives. But with covariance features we achieve a better performance in a much faster way.

5 Conclusion

In this paper we presented the covariance features and related algorithms for object detection and texture classification. Superior performance of the covariance features and algorithms were demonstrated on several examples with detailed comparisons to previous techniques and features. The method can be extended in several ways. For example, following automatical detection of an object in a video, it can be tracked in the following frames using this approach. As the object leaves the scene, the distance score will increase significantly which ends the tracking. Currently we are working on classification algorithms which use the Lie group structure of covariance matrices.

References

1. Rosenfeld, A., Vanderburg, G.: Coarse-fine template matching. IEEE Trans. Syst. Man. Cyb. **7** (1977) 104–107
2. Brunelli, R., Poggio, T.: Face recognition: Features versus templates. IEEE Trans. Pattern Anal. Machine Intell. **15** (1993) 1042 – 1052

3. Marée, R., Geurts, P., Piater, J., Wehenkel, L.: Random subwindows for robust image classification. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, San Diego, CA. Volume 1. (2005) 34–40
4. Turk, M., Pentland, A.: Face recognition using eigenfaces. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Maui, HI. (1991) 586–591
5. Black, M., Jepson, A.: Eigentracking: Robust matching and tracking of articulated objects using a view-based representation. Intl. J. of Comp. Vision **26** (1998) 63–84
6. Comaniciu, D., Ramesh, V., Meer, P.: Real-time tracking of non-rigid objects using mean shift. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Hilton Head, SC. Volume 1. (2000) 142–149
7. Porikli, F.: Integral histogram: A fast way to extract histograms in cartesian spaces. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, San Diego, CA. Volume 1. (2005) 829 – 836
8. Leung, T., Malik, J.: Representing and recognizing the visual appearance of materials using three-dimensional textons. Intl. J. of Comp. Vision **43** (2001) 29–44
9. Varma, M., Zisserman, A.: Statistical approaches to material classification. In: Proc. European Conf. on Computer Vision, Copehagen, Denmark. (2002)
10. Georgescu, B., Meer, P.: Point matching under large image deformations and illumination changes. IEEE Trans. Pattern Anal. Machine Intell. **26** (2004) 674–688
11. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, HI. Volume 1. (2001) 511–518
12. Lowe, D.: Distinctive image features from scale-invariant keypoints. Intl. J. of Comp. Vision **60** (2004) 91–110
13. Förstner, W., Moonen, B.: A metric for covariance matrices. Technical report, Dept. of Geodesy and Geoinformatics, Stuttgart University (1999)
14. Schmid, C.: Constructing models for content-based image retreival. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Kauai, HI. (2001) 39–45
15. Georgescu, B., Shimshoni, I., Meer, P.: Mean shift based clustering in high dimensions: A texture classification example. In: Proc. 9th Intl. Conf. on Computer Vision, Nice, France. (2003) 456–463