Sigma Set: A Small Second Order Statistical Region Descriptor

Xiaopeng Hong[1,2]   Hong Chang[2,3]   Shiguang Shan[2,3]   Xilin Chen[2,3]   Wen Gao[1,4]

[1]School of Computer Science and Technology, Harbin Institute of Technology, China

[2]Institute of Computing Technology, Chinese Academy of Sciences, China

[3]Key Laboratory of Intelligent Information Processing, Chinese Academy of Sciences, China

[4]Institute for Digital Media, Peking University, China

{xphong, hchang, sgshan, xlchen, wgao}@jdl.ac.cn

Abstract

Given an image region of pixels, second order statistics can be used to construct a descriptor for object representation. One example is the covariance matrix descriptor, which shows high discriminative power and good robustness in many computer vision applications. However, operations for the covariance matrix on Riemannian manifolds are usually computationally demanding. This paper proposes a novel second order statistics based region descriptor, named "Sigma Set", in the form of a small set of vectors, which can be uniquely constructed through Cholesky decomposition on the covariance matrix. Sigma Set is of low dimension, powerful and robust. Moreover, compared with the covariance matrix, Sigma Set is not only more efficient in distance evaluation and average calculation, but also easier to be enriched with first order statistics. Experimental results in texture classification and object tracking verify the effectiveness and efficiency of this novel object descriptor.

1. Introduction

Object descriptor plays crucial roles in computer vision (CV) community. Many low-level features, such as color, gradient, binary edges and filter responses [11], have been proposed to describe pixels of an image. In order to describe a region of pixels in higher level, one popular way is to use descriptors based on statistics, such as the histogram and covariance matrix.

A histogram [11] is a nonparametric estimation of the distribution over pixels in a region. It owns a simple form and shows good robustness against translation and rotation. It can be calculated efficiently, especially with fast algorithms like the integral histogram [17] and distributive histogram [19]. However, the dimensionality of histograms grows exponentially with the number of low-level features.

The region covariance matrix, as 2nd order statistics of an image region, is first introduced as a region descriptor by Tuzel et al. [20] and has been applied to many applications, such as texture classification [20], object detection [20],

object tracking [18] and human detection [21][25]. The covariance matrix descriptor shows good discriminative power and elegant robustness. For example, Wu and Nevatia show that the covariance matrix has the best discriminative power for human detection task [25] (compared with the HOG [7] and Edgelet [24] features). It has much lower dimensionality compared with histograms. Because covariance matrices do not lie on Euclidean space, a Riemannian manifold based framework for covariance matrices has been developed [16][18][20][21][25].

However, the main disadvantage of the covariance matrix lies in that the operations through Riemannian geometry are usually time-consuming, such as the matrix logarithm operation. In addition, it is not easy for the covariance matrix to combine with other information, such as the 1st order statistics of the region. Readers are referred to Section 2.3 for more detailed discussions.

In this paper, we propose a novel region descriptor which possesses the effectiveness of 2nd order statistics in a more efficient manner. We name this descriptor Sigma Set. The basic idea is as follows: suppose the dimensionality of the feature vector extracted from each pixel is *d*, 2nd order statistics (the covariance matrix) of a given region with any large number of pixels can be reconstructed with only *2d* points, which actually form the so-called Sigma Set. Hence Sigma Set is *equivalent* to the original object region in terms of 2nd order statistics. In this paper, it is adopted as a new region descriptor. We show that, Sigma Set is of low dimension, powerful and robust. Furthermore, since the descriptor is in the form of a set of points, the distance evaluation and average computation for Sigma Sets are simpler than those for covariance matrices through Riemannian geometry.

In some sense, this work is a revisit of the methods based on Sigma points [12][14], since our proposed Sigma Set is actually the set of Sigma points. However, there are two main differences between our method and the previous ones. First, the purposes are totally different. In [14], unscented transform uses Sigma points to make predictions of the filtering parameters (the mean and covariance matrix) easier and more accurate. In [12], Sigma points are adopted in a Monte-Carlo simulation to calculate the Kullback-Leibler Divergence between two normal

distributions. In this paper, Sigma Set, which consists of Sigma points, is introduced as an object descriptor. Second, to make Sigma Set feasible for object representation, this paper proposes algorithms for distance evaluation and average Sigma Set calculation, which are beyond the scopes of [12] and [14].

There are three main contributions in this paper. First, this paper proposes a novel object descriptor, Sigma Set, to represent 2nd order statistics of an object region in a form of a set, which brings more efficiency and flexibility than the covariance matrix. Second, to apply Sigma Set in real-world applications, such as texture classification and object tracking, we design algorithms for Sigma Set construction, distance evaluation and average calculation. Third, the proposed descriptor and corresponding algorithms extend and develop the previous framework which is based on Sigma points.

The remainder of this paper is organized as follows. In Section 2, the traditional covariance matrix descriptor with its Riemannian manifold framework is briefly reviewed to facilitate the subsequent presentations. In Section 3 and 4, we present the details of the proposed Sigma Set region descriptor, including algorithms for construction, distance evaluation and average Sigma Set computation. Experimental results in texture classification and object tracking in Section 5 show that this novel object representation method is effective, and it is simpler and more flexible than the covariance matrix descriptor. Section 6 concludes the paper.

2. Covariance matrix and its metrics

Here, we start with a brief review of the covariance matrix descriptor and its corresponding metrics.

2.1. Covariance matrix descriptor

The covariance matrix is first introduced as an object descriptor by Tuzel et al. [20]. Let I denote an image. For a given rectangular region $R \subseteq I$ with N pixels, let $\mathbf{f_i}$ be a $d \times 1$ feature vector extracted from the ith pixel in R, and $\mathbf{u}$ is the mean vector of the set of feature vectors $\{\mathbf{f_i}\}_{i=1,\cdots,N}$ in R. The bias covariance matrix $\mathbf{C}(R)$ of R can be calculated using Eqn.1:

$$\mathbf{C}(R) = \mathbf{F}_R \mathbf{F}_R{}^T , \qquad (1)$$

where $\mathbf{F}_R = \left[\widehat{\mathbf{f}}_1, \cdots, \widehat{\mathbf{f}}_N\right]_{d \times N}$ denotes the matrix of the centered vectors $\widehat{\mathbf{f}}_i = 1/\sqrt{N}\left(\mathbf{f_i} - \mathbf{u}\right)$, $i = 1, \cdots, N$. The region R is then represented by $\mathbf{C}(R)$, as illustrated in Fig.1.
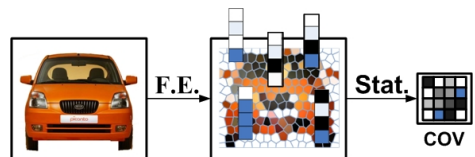

Figure 1: An image region represented by a 4×4 covariance matrix. "F.E.", "Stat." and "COV" are short for "Feature Extraction" and "Statistics" respectively.

2.2. Metric based on Riemannian manifold

Most of the covariance matrices used in real-word applications are positive definite, except the extreme case that some feature of all pixels inside a region has a constant value. It is worth noting that nonsingular covariance matrices (or symmetric positive definite (SPD) matrices) lie on a (nonlinear) Riemannian manifold but not on Euclidean space [20]. Hence, it is necessary to map covariance matrices to a linear tangent space of the manifold first, and then perform distance evaluation or classification task in the tangent space. Specifically, the mapping from a Riemannian manifold M to the tangent space T_C at one point $C \in M$ is achieved by the matrix logarithm operation $\mathrm{logm}_C : M \to T_C$. Inversely, expm, the matrix exponential operation maps points in the tangent space to the manifold [2].

Based on Riemannian geometry, the affine-invariant metric [9] and Log-Euclidean metric [1] are widely used for evaluating the distance between SPD matrices. Given T covariance matrices $\mathbf{C}_1, \cdots, \mathbf{C}_T$ and the identity matrix $\mathbf{I}$, Table 1 summarizes corresponding properties for these two metrics, where $\left\{\lambda_i(\mathbf{C}_1, \mathbf{C}_2)\right\}_{i=1,\cdots,d}$ are the generalized eigenvalues of $\mathbf{C}_1$ and $\mathbf{C}_2$, the Karcher mean is defined by [15] and $\|\bullet\|$ denotes the Frobenius norm of matrices [23]. Due to space limitation, we refer the readers to corresponding papers for more details.

2.3. Discussion

The covariance matrix descriptor with the corresponding Riemannian manifold framework is capable of capturing 2nd order statistics inside an object region. Moreover, it has high discriminative power and good robustness to the changes in illumination, view, and pose. A normalized covariance matrix, which is equivalent to a correlation matrix in the extreme situation, is even more robust in applications. Faster computation of covariance matrices in $O(d^2)$ time complexity can be achieved by the distributive histogram [19] and integral image [20].

	affine-invariant metric	Log-Euclidean metric
Distance $\mathrm{d}^2(\mathbf{C}_1,\mathbf{C}_2)$	$\sum_{i=1}^{d}\ln^2\lambda_i(\mathbf{C}_1,\mathbf{C}_2)$ (or $\left\|\mathrm{logm}_{\mathbf{I}}\left(\mathbf{C}_1^{-1/2}\mathbf{C}_2\mathbf{C}_1^{-1/2}\right)\right\|^2$)	$\left\|\mathrm{logm}_{\mathbf{I}}(\mathbf{C}_1)-\mathrm{logm}_{\mathbf{I}}(\mathbf{C}_2)\right\|^2$
Karcher mean $\mathbf{m}$	No closed form. Using a gradient descent procedure. $\mathbf{m}^{t+1}=\mathrm{expm}_{\mathbf{m}^t}\left(1/T\sum_{i=1}^{T}\mathrm{logm}_{\mathbf{m}^t}(\mathbf{C}_i)\right)$	$\mathrm{expm}_{\mathbf{I}}\left(1/T\sum_{i=1}^{T}\mathrm{logm}_{\mathbf{I}}(\mathbf{C}_i)\right)$

Table 1. Properties of two Riemannian metrics.

Because of these advantages, the covariance matrix descriptor has been successfully applied to many computer vision applications [16][18][20][21][25].

Despite its promising performance, there exist two disadvantages. First, it is not efficient to make evaluation using this descriptor. This is mainly due to inefficient computation of the matrix exponential and logarithm operations, although both of them have fairly simple expressions for SPDs [10]. Second, 2nd order statistics is the only information we can get from the covariance matrix. On one hand, there is no straightforward way to incorporate other statistics into a single descriptor; On the other hand, it is difficult to make statistical computations from multiple descriptors. For example, it is complicated to estimate the average of T covariance matrices with different weights through Riemannian metrics [1][9].

To address the above disadvantages, we propose a new and elegant region descriptor in the subsequent sections.

3. Sigma Set: essence of the original region

3.1. Basic idea

Let $\mathbf{C}(R)$ denote the covariance matrix of a given region R. This section aims at a new object descriptor which is able to capture 2nd order statistics of R more flexibly and more efficiently.

The basic idea of our method is to construct a small set of points, S which satisfies $\mathbf{C}(S)=\mathbf{C}(R)$. It means that S is *equivalent* to R in terms of 2nd order statistics. On one hand, we expect the set S to be extremely small, say $O(d)$ in its cardinality. On the other hand, we define efficient distance measures and propose a straightforward method to compute the average of the descriptors for multiple regions. We introduce S as a new region descriptor of R and name it as "Sigma Set". The points involved in S are called "Sigma points" (just as what they are named in [12] and [14]). For clarity, we use S to denote the Sigma Set descriptor in the following context.

In the next section, we will describe how to construct Sigma Set efficiently.

Input: A region R consisting of N $d\times1$ feature vectors.
Output: Sigma Set S satisfying $\mathbf{C}(S)=\mathbf{C}(R)$.
Algorithm:
1. Calculate the $d\times d$ covariance matrix $\mathbf{C}=\mathbf{C}(R)$ of R.
2. Perform Cholesky decomposition of $\mathbf{C}$, $\mathbf{C}=\mathbf{L}\mathbf{L}^T$, where $\mathbf{L}$ is a lower triangular matrix.
3. Multiple $\mathbf{L}$ by a scalar $\sqrt{d}$, that is $\mathbf{L}=\sqrt{d}\times\mathbf{L}$.
4. $S=\left\{\mathbf{L}_1,\cdots,\mathbf{L}_d,(-\mathbf{L}_1),\cdots,(-\mathbf{L}_d)\right\}$, where $\mathbf{L}_i$ is the ith column of $\mathbf{L}$.

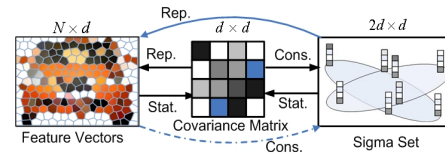Figure 2: Algorithm for Sigma Set construction.

Figure 3: Relationships among the covariance matrix, Sigma Set and the original region, where "Rep.", "Stat." and "Cons." are short for "Representation", "Statistics" and "Construction" respectively.

3.2. Sigma Set construction

Obviously, from Eqn.1, for any matrix $\mathbf{A}$ that satisfies $\mathbf{C}(R)=\mathbf{A}\mathbf{A}^T$, the set of columns of $\mathbf{A}$ has the same 2nd order statistics as R. More specifically, we can construct the Sigma Set descriptor for region R through Cholesky decomposition (CD) as summarized in Fig.2.

From Appendix A, it is easy to see that $\mathbf{C}(S)=\mathbf{C}(R)$. Furthermore, the cardinality of S is *2d*. Therefore, S is the small substitute set that we expect in Section 3.1. For representation convenience, we arrange the indices of elements in S constructed by Fig.2 and denote the set as $S=\left\{\mathbf{L}_1,\cdots,\mathbf{L}_d,\mathbf{L}_{(d+1)},\cdots,\mathbf{L}_{2d}\right\}$ hereinafter.

Cholesky Decomposition for SPD matrices exists uniquely [10]. Therefore, it can be easily drawn the existence and uniqueness of the Sigma Set constructed using the algorithm in Fig.2. Relationships among the original feature vectors, the covariance matrix, and Sigma Set for a region R are illustrated in Fig.3.

It is worth noting that other matrix operations, such as the matrix square root (SQ) operator $\mathbf{C}^{1/2}$ and singular value decomposition (SVD), can also lead to a set of points of size $O(d)$. The reason of choosing Cholesky

decomposition mainly lies in two folds. First, although the computational complexities of all methods mentioned above are $O(d^3)$, only $d^3/3$ atomic operations are needed for CD. Hence it is much less than the cost of SQ or SVD in practice (We will discuss the computational cost in detail in Section 4.4). Second, the special form of Sigma points obtained by CD can be adopted to construct point correspondence between Sigma Sets. Based on this correspondence, we can simplify the calculation of the distance between Sigma Sets and define the average of Sigma Sets for multiple object regions.

3.3. Properties of Sigma Set

Since Sigma Set is derived from the covariance matrix uniquely, the robustness of covariance matrix is obviously inherited by Sigma Set. More specifically, the robustness property may come from the following aspects: First, disregarding the order and the number of points in a region implies certain invariance under scale and rotation changes. Second, robustness holds by choosing proper features. For example, if features not related to orientation of the points is extracted, such as the intensity and the magnitude of gradient or Laplacian, it is rotationally invariant. Third, the normalized covariance matrix with unit standard deviation leads to an invariant property to some extent.

Besides, Sigma Set can be enriched to incorporate 1st order statistics of the object region. To incorporate the (2nd order) Sigma Set S with the (1st order) mean vector $\mathbf{u}$ of the region R, we introduce the set:

$$S_{+1} = \left\{ (\mathbf{L}_1 + \mathbf{u}), \cdots, (\mathbf{L}_d + \mathbf{u}), (\mathbf{L}_{(d+1)} + \mathbf{u}), \cdots, (\mathbf{L}_{2d} + \mathbf{u}) \right\}. \quad (2)$$

Similar to Appendix A, it can be easily seen that S_{+1} has the same 1st and 2nd statistics as R. Therefore, S_{+1} forms a new Sigma Sets in terms of both 1st and 2nd order statistics.

4. Distance and average model

4.1. Distance between Sigma Sets

After introducing Sigma Set as a region descriptor, it is necessary to define a way of measuring the dissimilarity between Sigma Sets. Considering its special form of a set with a limited number of points, the distance between Sigma Sets can be evaluated by any between-set distance metric.

The modified Hausdorff distance (MHD) [8] is widely used as a distance metric over closed and bounded sets. MHD has several desirable properties [8]. Hence MHD is introduced to evaluate Sigma Sets in this paper.

Given two Sigma Sets S_A and S_B both constructed by the algorithm in Fig.2 for two regions R_A and R_B respectively, the MHD $\mathrm{H}: S \times S \rightarrow R^+$ between them is defined as:

$$\mathrm{H}(S_A, S_B) = \max\{\mathrm{h}(S_A, S_B), \mathrm{h}(S_B, S_A)\}, \quad (3)$$

$$\mathrm{h}(S_A, S_B) = 1/2d \sum_{\mathbf{a} \in S_A} \min_{\mathbf{b} \in S_B} (\mathrm{d}_{\mathrm{E}}(\mathbf{a}, \mathbf{b})), \quad (4)$$

where $\mathrm{d}_{\mathrm{E}}(\bullet)$ in Eqn.4 can be any common distance metric defined in R^d, such as the Manhattan distance (L1 Norm) and Euclidean distance (L2 Norm).

Thanks to the symmetric property, Eqn.4 can be further simplified as:

$$\mathrm{h}(S_A, S_B) = \frac{1}{d} \sum_{i=1}^{d} \min_{j=1}^{d} \left(\mathrm{d}_{\mathrm{E}}(\mathbf{L}_i^A, \mathbf{L}_j^B) \right), \quad (5)$$

which halves the computational complexity, where $\mathbf{L}_i^A$ and $\mathbf{L}_i^B$, $i = 1, \cdots, d$, denote the ith and jth points whose signs are both positive in S_A and S_B respectively.

4.2. Point correspondence and fast approximation of distance calculation

In this section, we construct the point correspondence betweens Sigma Sets and adopt it to efficiently approximate the calculation of Eqn.5. For description clarity, we only focus on the Sigma points with positive signs. It can be easily extended to those Sigma points with negative signs.

Let $\mathbf{C}_A$ and $\mathbf{C}_B$ be the covariance matrices for two regions R_A and R_B respectively. $\mathbf{C}(k,l)$ denotes the (k,l)th entry of a covariance matrix $\mathbf{C}$, $k, l = 1, \cdots, d$. Since Sigma points come from a lower triangular matrix, the top $(i-1)$ elements in $\mathbf{L}_i$ are constantly zero for $i = 1, \cdots, d$. Here, we refer to the number of elements *not* constantly equal to 0 in column $\mathbf{L}_i$ as the *degree* of $\mathbf{L}_i$. Obviously, the *degree* of $\mathbf{L}_i$ is $(d+1-i)$.

From the procedure of Cholesky decomposition, it is obvious that if $\mathbf{C}_A(k,l)$ and $\mathbf{C}_B(k,l)$ are different, $\mathbf{L}_i^A$ and $\mathbf{L}_i^B$, the corresponding elements of the same *degree* in S_A and S_B would be different for $i = l, \cdots, d$. It implies that the difference between two Sigma Sets can be reflected by the (accumulation of) differences between Sigma points of the same *degree*. This situation motivates us to construct the point correspondence, which can greatly simplify the distance calculation. We formalize this empirical idea as:

$$d\left(\mathbf{L}_i^A, \mathbf{L}_j^B\right) = \begin{cases} d_E\left(\mathbf{L}_i^A, \mathbf{L}_j^B\right) & i = j \\ \infty & \text{others} \end{cases}. \quad (6)$$
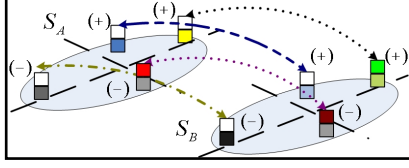
Figure 4: Point correspondence between two Sigma Sets.

Eqn.6 shows that only Sigma points of the same *degree* have a meaningful distance (or relationship). Therefore, the distance between $\mathbf{L}_i^A$ and $\mathbf{L}_i^B$ defined by Eqn.6 is referred as Restricted Point Distance. A simple case in two-dimensional space (d=2) is illustrated in Fig.4.

We replace $d_E(\bullet)$ in Eqn.5 with $d(\bullet)$ in Eqn.6 and get

$$h(S_A, S_B) = 1/d \sum_{i=1}^{d} d_E(\mathbf{L}_i^A, \mathbf{L}_i^B). \qquad (7)$$

Eqn.3 and Eqn.7 form a fast approximation of MHD, which is named Point Restricted Modified Hausdorff distance (PRMHD) of Sigma Sets. It is obvious that PRMHD satisfies the metric axioms [23].We will show in Section 5.1 that PRMHD achieves similar performance to that of MHD in texture analysis. Considering its higher efficiency, we use PRMHD instead of MHD in object tracking in Section 5.2 and obtain encouraging performance.

It is worth noting that the Restricted Point Distance in Eqn.6 brings many benefits. Besides the fast calculation of the distance, it can be adopted to define an average Sigma Set.

4.3. Average of multiple Sigma Sets

Many applications usually require calculating an average model for objects. For Sigma Set, the element correspondence defined in Section 4.2 can be easily extended to the case of multiple objects, which enables us to define S_M, the mean of T Sigma Sets $\{S_1, \cdots, S_T\}$ as:

$$S_M = \left\{ u\left(\mathbf{L}_i^1, \cdots, \mathbf{L}_i^T\right) \mid i = 1, \cdots, 2d \right\}, \qquad (8)$$

where $\mathbf{L}_i^t$ denotes the ith element in the tth Sigma Set, $t = 1, \cdots, T$ and $u(\bullet)$ can be any "mean" defined in R^d. In this paper, the arithmetic mean is adopted as a model update strategy in Section 5.2.

4.4. Computational analysis

As a low-dimensional representation, only *2d* points of dimensionality *d* are needed to form Sigma Set for a region R ($|R| = N, d$ is always much less than N). Because of the symmetric property (on signs of Sigma points) and the form of a triangular matrix, the space complexity of Sigma Set is $(d^2 + d)/2$, the same as the covariance matrix.

We compare time complexities using Sigma Set and the covariance matrix with the affine-invariant metric (It is similar for the Log-Euclidean metric) in Table 2. At the first glance, the construction cost of Sigma Sets is $O(d^3)$, larger than $O(d^2)$ for the covariance matrix. However, this is only one-sided view. On one hand, in most applications, the frequency of distance evaluation is always higher than that of descriptor construction. On the other hand, as we will explain soon, even in the "One Construction One Evaluation" mode (i.e. in single object tracking), the actual cost of Sigma Sets is less than that of the covariance matrix.
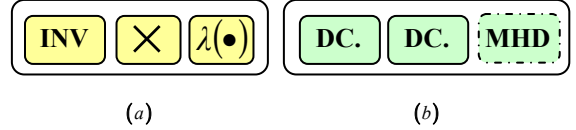

(*a*)   (*b*)

Figure 5: Dominating computational costs for (a) covariance matrices and (b) Sigma Sets. *Dashed rectangle* is only for MHD.

	Sigma Set PRMHD	Sigma Set MHD	Cov.
Construction	$O(d^3)$	$O(d^3)$	$O(d^2)$
Matching	$O(d^2)$	$O(d^3)$	$O(d^3)$
1st order statistics	$O(Td^2)$	N/A	$O(Td^3)$

Table 2. Comparisons of time complexity.

Because the construction of the covariance matrix is shared by these two methods, we compare the actual cost of "the covariance matrix evaluation" with that of "Sigma Set construction + evaluation" in the situation of "One Construction One Evaluation". The overall computation complexity of both descriptors is $O(d^3)$. However, the actual costs are very different.

The affine-invariant distance in Table 1 between two covariance matrices $\mathbf{C}_A$ and $\mathbf{C}_B$ can be calculated as:

$$d^2(\mathbf{C}_A, \mathbf{C}_B) = \sum_{i=1}^{d} \ln^2 \lambda_i(\mathbf{C}_A \mathbf{C}_B^{-1}). \qquad (9)$$

Main computational costs of the two descriptors are illustrated in Fig.5. From Fig.5 (a), the cost for covariance matrices is dominated by the matrix inverse, multiplication and eigenvalue calculation (always solved by the Jacobi algorithm). As for Sigma Set, the main cost consists of two times Cholesky decomposition (and additional MHD calculation when using the MHD distance metric).

As mentioned in Section 3.2, the cost of CD is about $1/d^3$ atomic operations, which is less than those of matrix inverse and multiplication and much less than those of the Jacobi algorithm. Even in the case of Sigma Set with the MHD metric, exact d^3 atomic operations are necessary for the additional MHD calculation. Therefore, we conclude that the actual cost of Sigma Sets is much less than that of covariance matrices. Experimental results in Section 5.3 will verify this point.

5. Applications

To demonstrate the effectiveness of Sigma Set, we introduce it in two applications: texture classification (which focuses on the distance metric) and target tracking (which is mainly for "mean" calculation). All experiments are implemented in Visual C++ 6.0 associated with OPENCV 1.0 Library and Matlab C++ Math Library.

5.1. Texture classification

We compare our Sigma Set based method with the random covariance method [20] in this subsection. For fair comparison, we duplicate the experimental configuration in [20], using the same Brodatz texture database [3], which contains $NC = 111$ texture classes[1]. The left-top and the right-bottom quarters of each 640×640 texture image are used for training while the rest parts for testing. s ($s = 100$ in our experiments) square regions with sizes between 16×16 and 128×128 are randomly sampled from each image in the training set. Thus there are 200 image patches for each texture class.

As in [20], a 5-dimensional feature vector

$$\mathbf{f}(x,y) = \left[\mathrm{I}(x,y), |\mathrm{I}_x|, |\mathrm{I}_y|, |\mathrm{I}_{xx}|, |\mathrm{I}_{yy}| \right]^T \qquad (10)$$

is extracted from each pixel, where $\mathrm{I}(x,y)$, $\mathrm{I}_x / \mathrm{I}_y$ and $\mathrm{I}_{xx} / \mathrm{I}_{yy}$ represent the intensity value, the 1st order and 2nd order derivatives at the position (x,y) respectively. A covariance matrix is then computed according to Eqn.1 for each image patch.

Input: an image I for testing.
Output: the class label that I belongs to.
Algorithm:
1. Reset a counter vector $Count[1,\cdots,NC]$.
2. Sample s square patches randomly $\{p_i\}$, $i = 1,\cdots,s$ with sizes between 16×16 and 128×128.
3. For each sample p_i, do
4.   Label p_i as class c_t, $t = 1,\cdots,NC$ from k nearest patches from the training set using **kNN** Algorithm.
5. $Count[t]++$.
6. End For
7. Return $id = \arg\max_t (Count[t])$.

Figure 6: Algorithm for texture classification using KNN.

	Cov.	MHD-L2	PRD-L1
Errors	5	5	6

Table 3. Classification results for Brodatz database.

---

[1] Although 112 classes are tested in [20], only 111 classes are available now since the 14th class cannot be found in [3] any more.
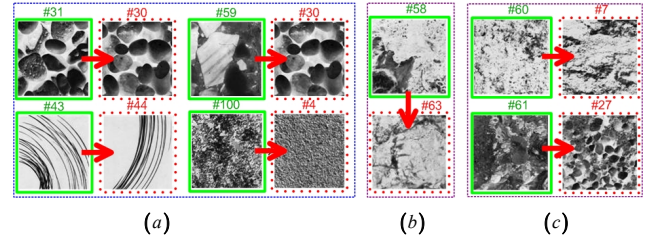

(a)   (b)   (c)

Figure 7: Misclassified samples in Brodatz database.

Given a test image, we execute the texture classification algorithm shown in Fig.6 to make a decision which class this image belongs to. In our experiments, we compare 3 different metrics for kNN classification (Step 4 in Fig.6): the PRMHD with L1 Norm $\mathrm{d}_E(\bullet)$ and MHD with L2 Norm $\mathrm{d}_E(\bullet)$ for Sigma Sets and the affine-invariant metric for covariance matrices. For clarity, we denote PRMHD with L1 norm as PRD-L1 and denote the traditional MHD with L2 norm as MHD-L2 hereinafter.

The classification performance is evaluated by the number of error decisions (of the 222 samples), as shown in Table 3, where the left column is for the covariance matrix and the other 2 columns are for Sigma Set. Fig.7 (a) shows 4 samples that were misclassified by all three methods. Both the covariance matrix and Sigma Set with MHD-L2 metric misclassified the 58th sample, as shown in Fig.7 (b). Two other samples misclassified by Sigma Set with the PRD-L1 metric are illustrated in Fig.7 (c). It can be seen that Sigma Set with the MHD-L2 metric achieved consistent performance as that of the covariance matrix (which is consistent with the result reported in [20]). Even the simplified PRD-L1 metric got comparable results.

5.2. Target tracking

We further apply Sigma Set to the visual tracking task. 10 public sequences, including 8 vehicles sequences from VIVID airborne sensor dataset [5] and 2 pedestrian sequences from CAVIAR database [4] are selected for evaluation. These sequences contain outliers and variations in shapes, motion, occlusions, viewpoints and illuminations. Most of them are captured by non-stationary cameras and some are infrared.

We adopt a unified feature vector as:

$$\mathbf{f}(x,y) = \left[x, y, \mathrm{I}(x,y), |\mathrm{I}_x|, |\mathrm{I}_y| \right]^T, \qquad (11)$$

which is the same as the configuration for the non-stationary and non-colorful setups in [18].
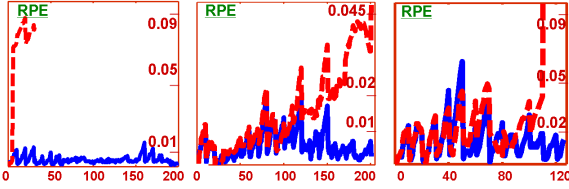
Figure 8: Curves of relative position error versus frame in *"Egtest02", "Egtest03"* and *"Pktest01"* (left to right). *Red dash lines* are for the covariance matrix and *blue real lines* for Sigma Set.

Given an initial target, the covariance matrix and Sigma Set[2] are constructed according to Eqn.1 and the algorithm in Fig.2, respectively. Then, similar to [18], the tracking task is considered as finding the most similar candidates, i.e., the nearest neighbor, in the following sequence. For fair comparison, no tracking techniques (such as Mean-Shift [6], Particle Filter [13] and Data Association [22]) are used. Instead, the most similar candidate is searched in a brute forth manner (similar to scan-window based object detection more or less). For conciseness, only PRD-L1 metric is tested for Sigma Set.

Since non-rigid and moving objects undergo shape, size, and appearance transformations in time, it is necessary to adapt to these variations. Therefore, we keep a set of T previous "best" candidates. For the covariance matrix, Karcher mean under affine-invariant metric (in Table 1) is adopted. For Sigma Set, an average of them is computed as an updated model using Eqn.8 with the arithmetic mean. As the flexible setting in [18], T is set to 5 in our experiments.

Detection rate is used to evaluate the tracking performance. It is defined as follows: if the overlap area between the tracking result and the ground truth (Gt.) are larger than half size of the Gt., the result is considered to be correct. It is looser than the one used in [18]. That's mainly due to the difficulties of the test sequences selected here. From Table 4, our method outperforms (or equals to) the covariance matrix descriptor on 8 sequences and get worse results in the other 2 sequences. We also illustrate the Relative Position Error (position error divided by the size of the target) curves of 3 sequences in Fig.8 and show the tracking results over other 3 sequences in Fig.9 to Fig.11. In the sequence *egtest01,* Sigma Set was able to capture the object in the respective frames, while covariance tracking suffered from a similar car. In Fig.10, due to the variation of the person, covariance tracking failed after the 60th frame, while our proposed method succeeded in tracking the target. For the sequence *redteam*, both of the two methods were able to locate the object in the first 250 frames. However, Sigma Set got a better detection rate (from Table.4). It can be easily seen that our method achieved better results.

Figure 9: Tracking results on *"egtest01"*. *Red rectangles* are for the covariance matrix and *blue circles* for Sigma Set.

Figure 10: Tracking results on *"walk1"*. *Red rectangles* are for the covariance matrix and *blue circles* for Sigma Set.
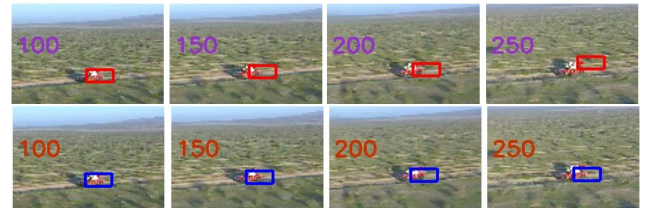
Figure 11: Tracking results on *"redteam"*. *Top row* is for the covariance matrix and *bottom row* for Sigma Set.

Sequence	Sigma Set	Cov.	Range[3]
Egtest01	.71	.27	# 1-217
Egtest02	1.0	.03	# 1-202
Egtest03	.97	.64	# 1-201
Egtest04	.33	.33	# 1-119
Pktest01	.94	.74	# 1-126
Pktest02	*.70*	*.83*	# 1-83
Pktest03	.26	.19	# 1-185
Redteam	.46	.26	# 1-400
Walk1	1.0	.45	# 38-165
OneStop.	*.51*	*.70*	# 706-755

Table 4. Detection rate for 10 selected sequences.

5.3. Speed comparison

Here, we evaluate the difference of actual costs between Sigma Set using PRD-L1 and covariance matrices with the affine-invariant metric in the "One Construction and One Matching" mode (in the tracking environment) using a 1.7G Hz Pentium-M laptop with 1GB memory. We implemented both frameworks using Matlab C++ Math Library, where most of the key functions, such as Cholesky decomposition and the calculation of General Eigenvalues (of two matrices) are provided. We recorded the time elapsed after a large amount (95,000 times) of pairs of "one evaluation one matching" operations for more than 50 times. The average time which Sigma Set taken was about 8

---

[2] For fair comparison, we do not adopt Sigma Set "with the 1st order statistics of region features" defined in Eqn.2 here, because it is not easy for the original covariance matrix [18] to include this extra information.
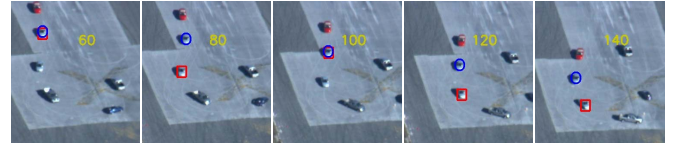
[3] Due to the naïve form of our implementation, it is of course not possible to fulfill the tracking task in all frames. Therefore only the first several hundred frames containing the target are tested.

seconds (about 0.08 milliseconds per pair of operations). The cost of the covariance matrix was nearly 20 seconds (about 0.21 milliseconds per operation pair), which was over 2 times slower than the proposed framework.

6. Conclusion

This paper has proposed an effective and efficient object descriptor, Sigma Set. Its main advantages can be summarized as follows:

- Sigma Set possesses the second order statistics of an object region in the form of a set of vectors.
- Sigma Set inherits the merits of the covariance matrix descriptor, including low-dimensionality, good robustness against noise and illumination changes and high discriminative power.
- Compared with the covariance matrix descriptor, the calculation of the distance between Sigma Sets and average Sigma Set are more efficient. It is also easier to be enriched with first order statistics.

In future work, we will pursue the possibilities of extending Sigma Set descriptor in other applications.

Appendix A

Proof: Considering the set R constructed by the algorithm in Fig.2, it is obvious that the mean vector of the features satisfies:

$$\mathrm{u}(S) = 1/2d \sum\nolimits_{i=1}^{d} \left(\sqrt{d}\mathbf{L_i} + \left(-\sqrt{d}\mathbf{L_i} \right) \right) = \mathbf{0} .$$

Thus, the 2nd order statistics covariance matrix of S is:

$$\mathbf{C}(S) = \frac{1}{2d} \sum\nolimits_{i=1}^{d} \left(\sqrt{d}\mathbf{L_i}\sqrt{d}\mathbf{L_i}^T + \left(-\sqrt{d}\mathbf{L_i} \right)\left(-\sqrt{d}\mathbf{L_i} \right)^T \right).$$

$$\therefore \mathbf{C}(S) = \sum\nolimits_{i=1}^{d} \left(\mathbf{L_i}\mathbf{L_i}^T \right) = \mathbf{C}(R) .$$

Acknowledgements

References

[1] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. In SIAM JMAA, 2006.
[2] W. Boothby. An Introduction to differentiable manifolds and Riemannian geometry, Academic Press, 2002.
[3] http://www.ux.his.no/~tranden/brodatz.html
[4] http://groups.inf.ed.ac.uk/vision/CAVIAR/
[5] R. Collins, X. Zhou, and S. Teh. An open source tracking testbed and evaluation web site. In PETS, 2005.
[6] D.Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-rigid objects using mean shift. In CVPR, 2000.
[7] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In CVPR, 2005.
[8] M. Dubuisson and A. Jain. A modified Hausdorff distance for object matching. In ICPR, 1994.
[9] W. Förstner and B. Moonen. A metric for covariance matrices. In Technical report, Stuttgart Univ., 1999.
[10] G. Golub and C. Van Loan. Matrix computations (3rd ed.), Section 4.2, Johns Hopkins University Press, 1996.
[11] R. Gonzalez and R. Woods. Digital image processing. Addison-Wesley Longman Publishing Co., Inc., 2001.
[12] J. Hershey and P. Olsen. Approximating the Kullback Leibler divergence between Gaussian Mixture Models. In ICASSP, 2007.
[13] M. Isard and A. Blake. CONDENSATION - Conditional density propagation for visual tracking. In IJCV, 29: 5–28, 1998.
[14] S. Julier and J. Uhlmann. A new extension of the Kalman filter to nonlinear systems. In Proc. AeroSense: 11th Int. Symp. Aerospace/Defence Sensing, Simulation and Controls, 1997.
[15] H. Karcher. Riemannian center of mass and mollifier smoothing. In Comm. on Pure and Appl. Math., 30: 509–541, 1977.
[16] X. Li, W. Hu, Z. Zhang, X. Zhang, M. Zhu, and J. Cheng. Visual tracking via incremental Log-Euclidean Riemannian subspace learning. In CVPR, 2008.
[17] F. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In CVPR, 2005.
[18] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on Lie Algebra. In CVPR, 2006.
[19] M. Sizintsev, K. Derpanis, and A. Hogu. Histogram-based search: a comparative study. In CVPR, 2008.
[20] O. Tuzel, F. Porikli, and P. Meer. Region covariance: a fast descriptor for detection and classification. In ECCV, 2006.
[21] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on Riemannian Manifolds. In TPAMI, 30: 1713–1727, 2008.
[22] C. Veenman, M. Reinders, and E. Backer. Resolving motion correspondence for densely moving points. In TPAMI, 3: 54–72, 2001.
[23] http://en.wikipedia.org/wiki/
[24] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by Bayesian combination of Edgelet part detectors. In ICCV, 2005.
[25] B. Wu and R. Nevatia. Optimizing discrimination efficiency tradeoff in integrating heterogeneous local features for object detection. In CVPR, 2008.