Beyond Covariance: Feature Representation with Nonlinear Kernel Matrices

Lei Wang[1], Jianjia Zhang[1], Luping Zhou[1], Chang Tang[1,2], Wanqing Li[1]
School of Computing and Information Technology, University of Wollongong, Australia[1]
School of Electronic Information Engineering, Tianjin University, China[2]

Abstract

Covariance matrix has recently received increasing attention in computer vision by leveraging Riemannian geometry of symmetric positive-definite (SPD) matrices. Originally proposed as a region descriptor, it has now been used as a generic representation in various recognition tasks. However, covariance matrix has shortcomings such as being prone to be singular, limited capability in modeling complicated feature relationship, and having a fixed form of representation. This paper argues that more appropriate SPD-matrix-based representations shall be explored to achieve better recognition. It proposes an open framework to use the kernel matrix over feature dimensions as a generic representation and discusses its properties and advantages. The proposed framework significantly elevates covariance representation to the unlimited opportunities provided by this new representation. Experimental study shows that this representation consistently outperforms its covariance counterpart on various visual recognition tasks. In particular, it achieves significant improvement on skeleton-based human action recognition, demonstrating the state-of-the-art performance over both the covariance and the existing non-covariance representations.

1. Introduction

Covariance matrix was proposed as a region descriptor around ten years ago[1] [22]. It can effectively fuse multiple features and be efficiently calculated via integral images. Also, it is partially invariant to rotation and scale changes and robust against outliers. Due to these properties, region covariance descriptor has shown promising performance in object detection, recognition and tracking [22, 18, 23].

The past several years have seen an expansion of covariance matrix in vision applications. In addition to a region descriptor, it has now been used as a generic feature representation and applied to various tasks including pedestrian detection [23], face recognition [15], action recognition [30, 5, 10], image set classification [27], shape retrieval [20], etc. A driving force to this trend is the powerful mathematical theory of Riemannian manifold of symmetric positive-definite (SPD) matrices.

The transition from region descriptor to generic feature representation brings forth new issues. Firstly, the dimensions of covariance matrix become higher, while the number of samples available for covariance estimation becomes smaller, as observed in human action recognition and image set classification. This results in unreliable or even singular covariance estimate, and the Riemannian metrics for SPD matrix cannot be directly applied. Secondly, covariance matrix only evaluates linear correlation of features. This might bring the advantages of simplicity and efficiency, when it is used as a region descriptor. Nevertheless, as a generic representation, the capability of modeling nonlinear feature relationship becomes essential. Last but not least, covariance matrix has a single, fixed form. It cannot be conveniently altered to model different feature relationships.

To address these issues, we propose to use kernel matrix as a generic feature representation. Each of its entries evaluates a kernel function between a pair of *feature dimensions* (rather than between a pair of *samples*, as we usually do in kernel methods). As will be shown, for a large set of kernel functions, the kernel matrix is guaranteed to be nonsingular, even if samples are scarce, which ensures Riemannian metrics to be readily applicable. More importantly, kernel matrix gives us unlimited opportunities to model nonlinear feature relationship in an efficient manner. Extracting different relationship is just a matter of changing the kernel function. In addition, this new representation can work well with covariance representation by providing complementary information. Combining them together can effectively improve recognition performance.

This paper first describes the background on covariance matrix and then discusses its newly encountered issues. After that, we propose kernel matrix as a generic feature representation and elaborate its properties and advantages. Following that, a framework of combining different representations is presented. At last, we discuss computational issues and the differences of our work from the existing ones. Ex-

[1] As a basic statistical concept, covariance matrix has long been used in all sorts of areas of computer vision, which is not the focus of this paper.

tensive experimental study is conducted on skeleton-based human action recognition, image set classification, and the classification tasks commonly applied with covariance matrix. As shown, the proposed new representation consistently achieves improved performance on these tasks. In particular, it demonstrates the state-of-the-art performance on skeleton-based human action recognition.

2. Background

Let $\mathbf{x}$ ($\mathbf{x} \in \mathbb{R}^d$) be a d-dimensional feature vector, and $\mathbf{D}_{d\times n} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$ denotes a data matrix. A sample-based covariance matrix $\mathbf{C}$ is defined as

$$\mathbf{C} = \frac{1}{n-1}\sum_{i=1}^{n}(\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top, \tag{1}$$

where $\boldsymbol{\mu}$ is the sample mean. In relation to feature representation, covariance matrix was initially proposed as a region descriptor [22]. Given an image region, a feature vector, $\mathbf{x}$, is extracted from each pixel to describe its location, colour, gradient, filter response, etc. With these feature vectors, covariance matrix is computed to characterize this region. As a region descriptor, it has the following merits: being a natural manner to fuse different features; being robust against illumination change and outliers; allowing two regions of different sizes to be compared; having rotation invariance when rotation-independent features are used; and fast computation via integral images.

A (nonsingular) covariance matrix belongs to the set of symmetric positive-definite (SPD) matrices, which forms a connected Riemannian manifold [23]. This non-Euclidean geometric property has been effectively considered in its distance measures including the generalized-eigenvalue-based metric [22], affine-invariant Riemannian metrics [16], log-Euclidean distance [1], Stein divergence [19], etc.

We categorize the applications of covariance matrix to visual recognition into two classes:

i) *As a region descriptor.* This dominates the initial applications. The superiority of region covariance descriptor is firstly shown on object detection and texture classification [22] and then on object tracking [18]. It is further applied to pedestrian detection [23], face recognition [15], and shape retrieval [20]. Two characteristics can be observed on these applications: 1) fast computation of region covariance descriptors is highly essential, especially for the tasks like object detection and tracking; 2) the dimensions of covariance matrix are usually low (e.g., 5×5 or 8×8).

ii) *As a general representation.* This has recently been seen in an increasing number of tasks. For human action recognition, a representation Cov3DJ is proposed to model a sequence of skeletal joint motions over time [10, 7]. In image set classification [27], a feature vector is extracted from every image in a given set and its covariance matrix is then computed to represent this image set. A similar case is observed in gesture recognition, where the covariance matrix of frame-based features is used to represent a video sequence. Two new characteristics have been observed.

1) The wider range of applications poses a challenge on covariance matrix with respect to its effectiveness as a generic representation. The requirement on extensively modeling sophisticated feature relationships becomes evident. As a result, new SPD-matrix-based representations with more expressive power are highly desired.

2) Features are not pixel-based anymore and often have higher dimensions. In an action recognition data set in the experiment, the dimensions are as high as 120, while the number of feature vectors per action instance only ranges from 40 to 500, far from being enough to estimate a reliable covariance matrix. A worse case is in image set classification. The dimensions could be as high as 400 (when reshaping a 20×20 object image), while there are only 41 images in a set [27]. This not only results in unreliable estimate but also incurs the singularity of covariance matrix.

3. Proposed method

To keep brevity, we use "covariance representation" and "kernel representation" as the short names of covariance-matrix-based and kernel-matrix-based representations.

3.1. Issues of covariance representation

Under the above new characteristics, covariance matrix as generic feature presentation has the following drawbacks.

Firstly, covariance matrix only describes linear correlation of features. Recall that $\mathbf{D}_{d\times n} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$ is a data matrix. Let $\mathbf{f}_i^\top$ ($i = 1, \cdots, d$) be the ith row of $\mathbf{D}$, standing for the realization of the ith feature. After centering, it can be written as $\bar{\mathbf{f}}_i = \mathbf{f}_i - \mu_i\mathbf{1}$, where μ_i is the sample mean while $\mathbf{1}$ is a column vector of "1"s. It is trivial to show that the (i, j)th entry of covariance matrix $\mathbf{C}$ is

$$c_{ij} = \left\langle \frac{\bar{\mathbf{f}}_i}{\sqrt{n-1}}, \frac{\bar{\mathbf{f}}_j}{\sqrt{n-1}} \right\rangle, \tag{2}$$

where $\langle\cdot,\cdot\rangle$ denotes an inner product. In other words, covariance matrix essentially implements a *linear* kernel function over scaled $\bar{\mathbf{f}}_i$ and $\bar{\mathbf{f}}_j$. When fast computation of a region descriptor is necessary, such linearity brings conceptual simplicity and computational efficiency. Nevertheless, from the perspective of generic representation, modeling only linear relationship significantly constrains its expressive power and in turn affects recognition performance. For example, for action recognition, it is certainly not sufficient to only consider the linear correlation of skeleton joints to model and differentiate various action patterns [21].

Secondly, the rank of covariance matrix obeys the rule that $\mathrm{rank}(\mathbf{C}) \leq \min(d, n-1)$. When $\mathbf{C}$ is used as a region

descriptor, the number of feature vectors extracted from an image region, n, is usually much larger than the dimensions, d. This ensures $\mathbf{C}$ to be nonsingular and allows it to be reliably estimated. However, this situation has changed in recent applications, and singularity could occur. In that case, in order to utilize Riemannian metrics, a small scaled identity matrix has to be appended [27].

3.2. Kernel matrix as feature representation

We propose to use a kernel matrix, $\mathbf{M}$, as a generic feature representation. The (i,j)th entry of $\mathbf{M}$ is defined as

$$k_{ij} = \langle \phi(\mathbf{f}_i), \phi(\mathbf{f}_j) \rangle = \kappa(\mathbf{f}_i, \mathbf{f}_j), \tag{3}$$

where $\phi(\cdot)$ is an implicit nonlinear mapping and $\kappa(\cdot,\cdot)$ is the induced kernel function. Covariance matrix corresponds to a special case in which $\phi(\mathbf{f}_i) = (\mathbf{f}_i - \mu_i \mathbf{1})/\sqrt{n-1}$. **Note that** the mapping $\phi(\cdot)$ is applied to each *feature* $\mathbf{f}_i$, rather than to each *sample* $\mathbf{x}_i$ as usually seen in kernel-based learning methods. The most significant advantage of using $\mathbf{M}$ lies at that with it, we can have much more flexibility to efficiently model the nonlinear relationship among features.

i) For example, we can evaluate the similarity of feature distributions, by applying the Bhattacharyya kernel [11]

$$\kappa(\mathbf{f}_i, \mathbf{f}_j) = \int \sqrt{p_i(z)} \sqrt{p_j(z)} dz, \tag{4}$$

where $p_i(z)$ and $p_j(z)$ denote two univariate distributions estimated from $\mathbf{f}_i$ and $\mathbf{f}_j$. When the two distributions are assumed to be Gaussian, denoted by $\mathcal{N}(\mu_i, \sigma_i)$ and $\mathcal{N}(\mu_j, \sigma_j)$, this kernel has a closed form as

$$\kappa(\mathbf{f}_i, \mathbf{f}_j) = \sqrt{\frac{2\sigma_i\sigma_j}{\sigma_i^2 + \sigma_j^2}} \exp\left[-\frac{1}{4}\frac{(\mu_i - \mu_j)^2}{\sigma_i^2 + \sigma_j^2}\right]. \tag{5}$$

ii) We can also model the interaction among samples with respect to a feature. Recall that $\mathbf{f}_j$ is a column vector $(x_{1j}, x_{2j}, \cdots, x_{nj})^\top$, where x_{ij} is the jth feature of the ith sample, $\mathbf{x}_i$. All the p-order "interaction" of samples can be exhaustively generated by mapping $\mathbf{f}_j$ (or $\bar{\mathbf{f}}_j$) as follows

$$\phi(\mathbf{f}_j) = \left\{ \sqrt{\left(\frac{p!}{r_1! r_2! \cdots r_n!}\right)} x_{1j}^{r_1} x_{2j}^{r_2} \cdots x_{nj}^{r_n} \right\}, \ \forall r_1, \cdots, r_n; \tag{6}$$

where $\sum_{i=1}^{n} r_i = p$ and $r_i \geq 0$. Introducing these features could be beneficial. For example, for skeleton-based action recognition, they consider all the p-order interactions of a given feature over the n frames of an action instance. This mapping induces a simple homogeneous polynomial kernel $\kappa(\mathbf{f}_i, \mathbf{f}_j) = \langle \phi(\mathbf{f}_i), \phi(\mathbf{f}_j) \rangle = (\langle \mathbf{f}_i, \mathbf{f}_j \rangle)^p$, where p is the degree of this kernel [2]. Therefore, with the proposed kernel representation, the relationship between a pair of high-order sample interactions can be conveniently evaluated.

iii) In practice, applying a kernel representation could be even easier, when we do not know beforehand (or are not particularly interested in) what kind of nonlinear relationship shall be modeled. In this case, any general-purpose kernel, such as the Gaussian RBF kernel $\kappa(\mathbf{f}_i, \mathbf{f}_j) = \exp(-\beta \|\mathbf{f}_i - \mathbf{f}_j\|^2)$, can be employed. Also, once it becomes necessary, users are free to design new, specific kernels to serve their goals. Such flexibility is clearly an advantage brought by using a kernel matrix as feature representation.

In relation to the singularity issue, kernel matrix is also a better choice than covariance matrix. When $d \geq n$ is true, covariance matrix is bound to be singular. In contrast, the situation is more favorable for kernel matrix. A direct application of Micchelli's Theorem (1986) [9] gives the following result for our case.

Theorem 1. *Let $\mathbf{f}_1, \mathbf{f}_2, \cdots, \mathbf{f}_d$ be a set of different n-dimensional vectors. The matrix $\mathbf{M}_{d\times d}$ computed with a RBF kernel $\kappa(\mathbf{f}_i, \mathbf{f}_j) = \exp(-\beta \|\mathbf{f}_i - \mathbf{f}_j\|^2)$ is guaranteed to be nonsingular, no matter what values d and n are.*

According to Micchelli's Theorem, the inverse multiquadric kernel $\kappa(\mathbf{f}_i, \mathbf{f}_j) = (\|\mathbf{f}_i - \mathbf{f}_j\|^2 + \beta^2)^{-1/2}$ also satisfies the above theorem. Actually, as pointed out in [4], in addition to these two kernels, there are a large set of various kernels holding this nice property, including radial kernels, translation invariant kernels, multiscale kernels, power series kernels, etc. The presence of these kernels provides us great freedom to choose the most appropriate one for a kernel representation. Lastly, in case we cannot be sure about the nonsingularity for a kernel matrix, we can always analyze it with the definition of positive definiteness and/or append a regularizer to this matrix as a preemptive measure.

3.3. Combining different feature representations

Kernel- and covariance-representations extract different feature relationships, and can therefore be combined. Let $\mathcal{D}$ denote a set of m training samples for classification. Let's assume that there are q representations in total, and the ith training sample is represented by $\mathbf{M}_{i1}, \mathbf{M}_{i2}, \cdots, \mathbf{M}_{iq}$. The class label of the ith training sample is denoted by y_i. Two combination ways are proposed as follows.

One way directly combines the q representations by weight $\mathbf{w} = (w_1, \cdots, w_q)^\top$, called "early fusion" in this work. This gives a combined representation as

$$\mathbf{M}_i(\mathbf{w}) = \sum_{j=1}^{q} w_j \mathbf{M}_{ij}. \tag{7}$$

Applying a kernel function $k(\mathbf{M}_i(\mathbf{w}), \mathbf{M}_j(\mathbf{w}))$[2], we can obtain another kernel matrix computed over the whole training set $\mathcal{D}$, denoted by $\mathbf{G}(\mathbf{w})$. Viewing the weight $\mathbf{w}$ as the

[2]Note that two different kernels are involved: $\kappa(\mathbf{f}_i, \mathbf{f}_j)$ is used to compute the proposed kernel representation $\mathbf{M}$ for each training sample; $k(\mathbf{M}_i, \mathbf{M}_j)$ is used to compute the kernel between two training samples, as what we usually do in kernel-based learning methods.

parameter of the kernel k, we can learn its optimal value by following the kernel parameter tuning approach commonly used in SVMs [3], that is, minimizing the "inverse of the margin". This leads to the following optimization,

$$\min_{\mathbf{w}\in\mathbb{R}^q}\left\{\max_{\boldsymbol{\alpha}\in\mathbb{R}^m}\left[\boldsymbol{\alpha}^\top\mathbf{1}-\frac{1}{2}(\boldsymbol{\alpha}\odot\mathbf{y})^\top\left(\mathbf{G}(\mathbf{w})+\frac{\mathbf{I}}{C}\right)(\boldsymbol{\alpha}\odot\mathbf{y})\right]\right\} \quad (8)$$
$$s.t.\ \mathbf{w}^\top\mathbf{1}=1;\ \ \mathbf{w}\geq\mathbf{0};\ \ \boldsymbol{\alpha}^\top\mathbf{y}=0;\ \ \boldsymbol{\alpha}\geq\mathbf{0},$$

where $\boldsymbol{\alpha}$ consists of m multipliers in SVMs; $\mathbf{y}$ is a vector of $y_1,\cdots,y_m$ (binary classification with $y_i=\pm1$ is assumed); and "$\odot$" denotes component-wise multiplication. Note that an ℓ_2-norm soft-margin SVM is used, where C is its regularization parameter and $\mathbf{I}$ is an identity matrix. This optimization problem can be solved by alternately optimizing $\boldsymbol{\alpha}$ and $\mathbf{w}$, and the details can be found in [3].

The other way fuses the q representations by multiple kernel learning (MKL), called "late fusion" in this work. A kernel function k is applied to each representation, resulting q kernel matrices $\mathbf{G}_1,\mathbf{G}_2,\cdots,\mathbf{G}_q$. With weight $\mathbf{w}$, the combined kernel matrix on the whole training set $\mathcal{D}$ is

$$\mathbf{G}'(\mathbf{w})=\sum_{j=1}^{q}w_j\mathbf{G}_j;\ s.t.\ \mathbf{w}^\top\mathbf{1}=1;\ \ \mathbf{w}\geq\mathbf{0}. \quad (9)$$

Replacing $\mathbf{G}(\mathbf{w})$ in Eq (8) with $\mathbf{G}'(\mathbf{w})$ and solving the resulting optimization problem will give rise to the optimal weight. This problem can be solved by the off-the-shelf MKL packages such as SimpleMKL.

3.4. Differences from existing work

Improving the effectiveness of covariance representation has been studied in the literature. Existing work in this aspect can generally be categorized into three approaches.

The first approach improves the quality of visual features. For example, considering that Gabor features could extract more important information, they are used to replace the first- and second-order gradients at each pixel to compute covariance matrix in face recognition [15]. The second approach reduces the interference from the background within an image region. In object tracking [28], pixels are weighted in computing covariance matrix, and the farther a pixel is from the center of a region, the lower is its weight. The third approach, which may be most related to ours, considers to model high-order statistics of features [14, 7]. It maps $\mathbf{x}_1,\cdots,\mathbf{x}_n$ (Eq (1)) to a feature space by a kernel function. This results in a potentially *infinite-dimensional* covariance matrix (defined in the kernel-induced feature space) as feature representation. Because these covariance matrices cannot be explicitly computed, a special measure has to be derived to evaluate their similarity. The computational complexity of this measure for a pair of covariance matrices is $\mathcal{O}(n^3)$ due to the need of eigen-decomposition [7]. This becomes computationally expensive when n is large.

Evidently, our approach is different from all the above three ones. Ours is orthogonal to the first approach since it can work with any feature set. Also, in contrast to the first and second ones, we use a kernel matrix rather than a covariance matrix as feature representation. Compared with the third approach, we apply the kernel mapping to each *feature* $\mathbf{f}_1,\cdots,\mathbf{f}_d$, instead of each *sample* $\mathbf{x}_1,\cdots,\mathbf{x}_n$. The resulting representation still maintains the same dimensions ($d\times d$) as the original covariance representation, and does not need any special measures to be designed. Moreover, as will be demonstrated, our approach consistently achieves better recognition performance than the third one. Also, it runs as efficiently as the original covariance representation, and could be much faster than the third approach in [14, 7]. In addition, note that our approach aims to utilize a kernel matrix to represent an individual feature set. This is different from existing work developing a kernel function to measure the similarity of two feature sets [17].

3.5. Computational issues

Without loss of generality, we use the commonly used RBF kernel as an example. Given n d-dimensional vectors, $\mathbf{x}_1,\cdots,\mathbf{x}_n$ (Eq (1)), computing all the entries $\|\mathbf{f}_i-\mathbf{f}_j\|^2$ $(i,j=1,\cdots,d)$ has the complexity of $\mathcal{O}(nd^2)$, same as computing a covariance matrix. Certainly, RBF kernel has an $\exp(\cdot)$ operation and needs a bit more time. In addition, although the case of region descriptor is not our focus, we show that the proposed kernel representation could still be quickly computed via integral images. Noting that $\|\mathbf{f}_i-\mathbf{f}_j\|^2=\mathbf{f}_i^\top\mathbf{f}_i-2\mathbf{f}_i^\top\mathbf{f}_j+\mathbf{f}_j^\top\mathbf{f}_j$, we can precompute d^2 integral images for the inner product of any two feature dimensions. It then becomes trivial to compute $\|\mathbf{f}_i-\mathbf{f}_j\|^2$ for any rectangular regions by following [22]. This result is also valid for the polynomial kernel which computes $\mathbf{f}_i^\top\mathbf{f}_j$.

Generally, the availability of more samples makes kernel evaluation more reliable. Take the Bhattacharyya kernel (Eq (4)) as an example. More samples make the estimates μ and σ converge towards their true values. This in turn helps the kernel evaluation to converge towards its true value. Certainly, in practice we are constrained by the number of available training samples. Also, recall that the proposed kernel matrix has a fixed size ($d\times d$), independent of the number of samples (n) in a set. Due to this, the kernel-based representations obtained from two different-sized sets can be directly compared. At the same time, considering that n affects the lengths of $\mathbf{f}_i$ and $\mathbf{f}_j$, we scale them accordingly to reduce the impact of n. For example, we divide $\|\mathbf{f}_i-\mathbf{f}_j\|$ by the average pairwise Euclidean distance over a training set, when the RBF kernel is used.

4. Experimental result

The proposed kernel representation (Ker-RP in short) is compared with covariance representation (Cov-RP) on three

types of recognition tasks. The first two are human action recognition and image set classification, which use covariance as generic feature representation. The third one includes the tasks of face, texture, and object recognition traditionally used when covariance acts as a region descriptor.

All the three kernels in Section 3.2 are involved. In specific, the representations generated by the Gaussian radial basis function kernel (RBF in short) and the polynomial kernel (POL) are compared with covariance representation to verify their advantages. The representation generated by the Bhattacharyya kernel (BHA) will be combined with covariance representation to demonstrate the benefit of combining two complementary representations.

A nonlinear SVM classifier is used in all experiments. The log-Euclidean kernel, a commonly used kernel function on SPD matrices[3], is employed for the SVM. To ensure fair comparison, all algorithmic parameters, including the regularization parameter in SVM, β in the log-Euclidean kernel, and the parameters in the RBF and POL kernels are tuned by multi-fold cross-validation on the training set only.

4.1. Comparison on human action recognition

Four benchmark data sets are used, including MSR-Action3D, MSR-DailyActivity3D, MSRC-12, and HDM05. For all of them, we only use the *skeleton* data while the other data (e.g., depth maps or RGB videos) are not included. The data set information is in Table 1.

Table 1. Feature dimensions of four action recognition data sets.

Data set	#Dim. (d)	#frames per instance (n)	Features
MSR-Action3D	120	$40 \sim 60$	Velocity
MSR-DailyActivity3D	120	$125 \sim 500$	Velocity
MSRC-12	60	$50 \sim 300$	Coordinates
HDM05	93	$30 \sim 700$	Coordinates

For MSR-Action3D and MSR-DailyActivity3D, we use velocity as the frame features [31] by calculating the coordinate differences of 3D skeleton joints between a frame and its two (before and after) neighboring frames. Each frame feature vector has 120 ($2 \times 3 \times 20$ joints) dimensions. For MSRC-12 and HDM05, the 3D coordinates of each joint are used as the frame features. As seen in Table 1, for each data set, the number of frames per instance, n, could be smaller than $(d+1)$, which causes singularity when computing a covariance matrix. In this case, we append a small regularizer $\lambda \mathbf{I}$ (e.g., $\lambda = 10^{-7}$) to the matrix as in the literature [27].

To facilitate comparison with the state-of-the-art methods, the training and test sets of these data sets are partitioned by following the literature. For MSR-Action3D, MSR-DailyActivity3D and MSRC-12, the cross-subject test setting [12] is used, in which the odd-indexed subjects are used for training and the even-indexed ones are for test. For HDM05, we used the instances of two subjects for training and those from the remaining three subjects for test [6].

[3]The log-Euclidean kernel function is defined as $k(\mathbf{X}, \mathbf{Y}) = \exp\left(-\beta \| \log(\mathbf{X}) - \log(\mathbf{Y}) \|_F^2\right)$, where $\mathbf{X}$ and $\mathbf{Y}$ are two SPD matrices and $\log(\cdot)$ denotes the matrix logarithm.

4.1.1 Result on MSR-Action3D data set

MSR-Action3D contains 20 actions performed by ten subjects. Each action is done two or three times by each subject. The number of frames in each action instance is $40 \sim 60$, which is smaller than the feature dimensions, 120. The classification accuracy is compared in Table 2.

The upper portion of this table quotes the state-of-the-art results in the last two years, while the lower portion lists the results of the methods implemented by this work. Cov-RP is the method using covariance representation. Cov-$J_{\mathcal{H}}$-SVM is the method in [7] which uses an infinite-dimensional covariance matrix in a kernel-induced feature space as representation. Ker-RP-POL and Ker-RP-RBF are the proposed methods, in which polynomial and RBF kernels are used to compute the kernel representation. As seen, Cov-RP performs poorly when compared with the quoted state-of-the-art methods. Its performance is probably affected by the insufficient number of frames for covariance matrix estimation. Cov-$J_{\mathcal{H}}$-SVM well improves over Cov-RP. However, it is still inferior to the quoted state-of-the-art ones. In contrast, the proposed methods significantly outperform Cov-RP, obtaining an improvement over 20 percentage points. Also, both methods outperform Cov-$J_{\mathcal{H}}$-SVM by a large margin, and even win these state-of-the-art methods which use complex feature representation (e.g., sparse coding [29]) or multiple forms of data such as depth maps and skeleton [32]. This result is significant and encouraging, indicating the efficacy of the kernel representation. With it, classification accuracy on this action data set is boosted from 94.3% [32] further to 96.9%.

Table 2. Comparison on MSR-Action3D data set.

Methods in comparison	Accuracy
Pose Set [25]	90.0
Hierarchy of Cov3DJs [10]	90.5
Moving Pose [31]	91.7
Lie Group [24]	92.5
SNV [29]	93.1
Spatiotemp. Features Fusing [32]	94.3
Cov-RP [22]	74.0
Cov-$J_{\mathcal{H}}$-SVM [7]	80.4
Ker-RP-POL (proposed)	96.2
Ker-RP-RBF (proposed)	**96.9**

4.1.2 Result on MSR-DailyActivity3D data set

MSR-DailyActivity3D is a challenging data set, because the extracted skeletons are noisy and most activities involve

human-object interactions such as *drink*, *eat*, *read book*, etc. Table 3 shows the comparison result. As previous, some state-of-the-art results are quoted in the upper portion, followed by the results of the methods implemented by this work. On this data set, Cov-RP becomes better and close to the state-of-the-art ones. However, Cov-$J_{\mathcal{H}}$-SVM does not improve over Cov-RP but shows a degraded performance. The two proposed methods once again demonstrate significant improvement over all the other methods. In specific, Ker-RP-POL yields the highest accuracy 96.9%. It wins the best state-of-the-art method (SNV [29]) by more than ten percentage points. Ker-RP-RBF also achieves an excellent result of 96.3%, close to Ker-RP-POL and outperforms the other ones by a large margin. Note that in these state-of-the-art methods, depth map is used to extract features in [13, 29], and local occupancy patterns are used in [26] to process human-object interaction cases. We compute the kernel representation using the skeleton data only. In addition, for this data set, the number of frames in each action instance is generally larger than the feature dimensions, making covariance estimation free of the singularity issue. Nevertheless, the result shows that the proposed kernel representation still has an advantage over covariance representation in this case. We attribute this advantage to its capability in modeling nonlinear feature relationship.

Table 3. Comparison on MSR-DailyActivity3D data set.

Methods in comparison	Accuracy
Moving Pose [31]	73.8
Local HON4D [13]	80.0
Actionlet Ensemble [26]	86.0
SNV [29]	86.3
Cov-RP [22]	85.0
Cov-$J_{\mathcal{H}}$-SVM [7]	75.0
Ker-RP-POL (proposed)	**96.9**
Ker-RP-RBF (proposed)	96.3

4.1.3 Result on HDM05 data set

HDM05 consists of around 1500 instances from over 100 motion classes. Most classes have 10 to 50 realizations of five actors named "bd", "bk", "dg", "mm" and "tr". We use two subjects "bd" and "mm" for training and the remaining three for test [6]. To compare with the literature, we conduct two experiments. Firstly, we use 14 classes[4] of this data set, and the result is in the left column of Table 4. Cov-RP shows quite competitive performance and outperforms the quoted methods. However, Cov-$J_{\mathcal{H}}$-SVM shows a degraded performance again. Ker-RP-RBF and Ker-RP-POL are still significantly better than Cov-RP, Cov-$J_{\mathcal{H}}$-SVM, and the other methods. The highest classification accuracy 96.8% is obtained by Ker-RP-RBF. Note that all the quoted methods use covariance-based representation, but sparse coding or dimensionality reduction is additionally applied to improve the performance. To further verify their effectiveness, we conduct comparison on all the classes. As shown in the right column of Table 4, although the significant increase on the number of action classes reduces the overall classification accuracy, the proposed methods still outperform the other ones in comparison.

[4]They are 'clap above head', 'deposit floor', 'elbow to knee', 'grab high', 'hop both legs', 'jog', 'kick forward', 'lie down floor', 'rotate both arms backward', 'sit down chair', 'sneak', 'squat', 'stand up lie' and 'throw basketball'.

Table 4. Comparison on HDM05 data set (Two experiments).

Methods in comparison	14 classes Accuracy	All classes Accuracy
CDL [27]	79.8	Not reported
RSR [8]	76.1	Not reported
RSR-ML [6]	81.9	40.0
Cov-RP [22]	91.5	58.9
Cov-$J_{\mathcal{H}}$-SVM [7]	82.5	-
Ker-RP-POL (proposed)	93.6	64.3
Ker-RP-RBF (proposed)	**96.8**	**66.2**

⋆The result of Cov-$J_{\mathcal{H}}$-SVM [7] is not obtained in 35 hours.

4.1.4 Result on MSRC-12 data set

MSRC-12 is a large data set, containing the performance of 12 gestures by 30 subjects. As shown in Table 5, Ker-RP-RBF again obtains the best classification result, outperforming Cov-RP and the other methods including Cov-$J_{\mathcal{H}}$-SVM. Ker-RP-POL's performance is a bit lower than that of Ker-RP-RBF. This may indicate that the RBF kernel fits better the action data in this data set. Nevertheless, Ker-RP-POL is still higher than Cov-RP and Cov-$J_{\mathcal{H}}$-SVM. Note that the method in [10] uses a hierarchy of multiple covariance matrices to capture the temporal order of motion. For each instance, the covariance matrix at the top level is computed over the whole sequence, while those at the lower levels are computed over a series of sub-sequences in order. We believe that our methods can be further improved if working in that manner.

Table 5. Comparison on MSRC-12 data set.

Methods in comparison	Accuracy
Hierarchy of Cov3DJs [10]	91.7
Cov-RP [22]	89.2
Cov-$J_{\mathcal{H}}$-SVM [7]	89.2
Ker-RP-POL (proposed)	90.5
Ker-RP-RBF (proposed)	**92.3**

4.2. Result on image set classification

An image set is a collection of images belonging to the same class but with variation, for example, images of the same object under different views. It is the image set, rather than an individual image, that will be classified. Covariance matrix has been used to model an image set [27]. Now we

compare it with the proposed kernel representation. Three data sets used by [27] are tested, including ETH80, CMU MoBo, and YouTube Celebrities. ETH80 has eight categories, with ten objects per category. For each object, there are 41 images showing different views. CMU MoBo has 96 video sequences of 24 subjects. YouTube Celebrities consists of 1910 video clips from 47 subjects. These data sets are preprocessed by [27] as follows. For YouTube and CMU MoBo, face images of each subject are collected by face detectors and resized to 20×20 pixels. Pixel intensities are used as features, leading to a 400-dimensional vector per image. The object images in ETH80 are also resized to 20×20 and pixel intensities are used as features. These data sets are downloaded from [27].

Training and test sets are created as follows. For CMU MoBo, all face images detected from the same video sequence form an image set. One image set is randomly selected from each subject for training, and the remaining image sets are for test. For YouTube, three image sets are randomly chosen from each subject for training, and another six sets are randomly chosen for test. In ETH80, the ten objects in a category are randomly halved into training and test sets. For each object, the 41 images of different views form an image set. The kernel- and covariance-representations are used to represent each image set. In total, 100 training and test pairs are created for each data set.

Following [27], we use Partial Least Squares (PLS) for classification and the code is downloaded from that work[5]. Table 6 reports the average results. Ker-RP-RBF achieves the best classification performance on ETH80, outperforming Cov-RP by 3.2 percentage points and Cov-$J_{\mathcal{H}}$-SVM by 3.5 percentage points. On CMU MoBo, it still significantly improves over Cov-RP and is comparable to Cov-$J_{\mathcal{H}}$-SVM. On YouTube, Ker-RP-RBF performs slightly worse than Cov-RP by 0.8 percentage point but clearly outperforms Cov-$J_{\mathcal{H}}$-SVM. Also, Ker-RP-POL performs better than Cov-RP on ETH80 by 2.1 percentage points, while worse on the other two data sets. This result reflects the importance of choosing an appropriate kernel function for the kernel representation. Also, compared with all the other methods, the RBF-kernel representation shows overall best classification performance over the three data sets.

Table 6. Comparison on three image set classification data sets.

Methods	ETH80	CMU MoBo	YouTube
Cov-RP (CDL [27])	92.7	83.9	**61.2**
Cov-$J_{\mathcal{H}}$-SVM [7]	92.4	**88.9**	54.4
Ker-RP-POL (proposed)	94.8	75.3	57.3
Ker-RP-RBF (proposed)	**95.9**	88.4	60.4

[5]The work [27] also investigates Linear Discriminant Analysis. However, PLS always outperforms LDA as shown in that work.

4.3. Result on object classification

We further investigate the effectiveness of the proposed kernel representation on the tasks traditionally applied with covariance matrix as a region descriptor. For them, the feature dimensions are usually lower and a larger number of feature vectors are available for covariance estimation. We use three data sets, including Brodatz for texture classification, FERET for face recognition, and ETH80 for object categorization. Brodatz contains 112 textured images. Following the literature [8], each image is partitioned into 64 non-overlapping sub-images. All sub-images from the same image form one texture class, and these sub-images are classified. For FERET, we use the "b" subset of 198 subjects. Each has 10 images with various poses and illumination conditions. ETH80 was used for image set classification in Section 4.2, but here each image is considered as a training or test sample and classified.

For all three data sets, every image/sub-image is scaled to a uniform size of 64×64 and a 43-dimensional feature vector is extracted at each pixel, including its intensity, x and y coordinates, and a set of Gabor features (8 orientations and 5 scales). For each experiment, we randomly halve the data set into training and test subsets. This is repeated 20 times to obtain average classification performance. As seen in Table 7, Ker-RP-RBF again outperforms Cov-RP, by 3.7 and 4.4 percentage points on Brodatz and FERET. This indicates the effectiveness of the proposed kernel representation even when it acts as a region descriptor. Note that Cov-$J_{\mathcal{H}}$-SVM is not included because it becomes time-consuming when the number of feature vectors, n, is large. As reported in Table 9, we cannot obtain its result even after 35 hours.

In addition, since the number of feature vectors (4096 per image) is now adequately larger than feature dimensions (43), we can compare the sensitivity of the two representations against the number of feature vectors. Brodatz data set and the RBF kernel are used. In Figure 1, the x-axis is the ratio of the number of feature vectors used to compute the kernel- or covariance-representations. The y-axis is the classification accuracy corresponding to the resulting representation. As shown, Ker-RP-RBF consistently outperforms Cov-RP, although both of them degrade with the decreasing ratio. In particular, the margin between them becomes even larger when the ratio is lower than $1/75$ (about 55 feature vectors), indicating the more significant advantage of Ker-RP-RBF when feature vectors are scarce. This suggests that modeling nonlinear feature relationship enhances the expressive power of SPD-matrix-based representation and benefits classification, especially in the case of a small number of feature vectors available.

Table 7. Comparison on object classification data sets.

Methods	Brodatz (texture)	FERET (face)	ETH80 (object)
Cov-RP [22]	81.2	81.0	94.0
Ker-RP-POL (proposed)	77.9	82.4	93.8
Ker-RP-RBF (proposed)	**84.9**	**85.4**	**94.8**

⋆The result of Cov-$J_{\mathcal{H}}$-SVM [7] is not obtained in 35 hours.

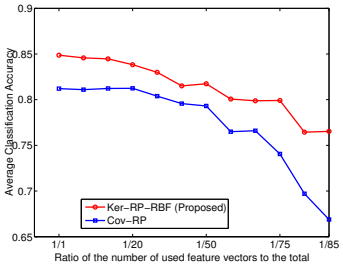

Figure 1. Comparison of the sensitivity (in terms of classification accuracy) of the kernel- and covariance-representation with respect to the number of feature vectors used to compute them.

4.4. Combining different representations

We combine Cov-RP and Ker-RP-BHA. The BHA denotes the Bhattacharyya kernel in Eq (5). It evaluates the similarity of the distributions of different features, but loses the information on the co-occurrences of features from the same feature vector. On the other hand, such information is preserved in covariance representation via the inner product (Eq (2)), although it only models the linear relationship. Due to their complementariness, we combine the two representations to show the benefit. MSR-Action3D and MSR-DailyActivity3D are used. The result is in Table 8. Two combination ways, early fusion and later fusion, are tested. As seen, both Cov-RP and Ker-RP-BHA show less promising performance on the two data sets (Ker-RP-BHA is still slightly better than Cov-RP). However, once combined, they obtain significant improvement. With early fusion, the improvements are more than 15 and 10 percentage points on the two data sets. Early fusion performs better than later fusion. In addition, we tried combining Cov-RP and Ker-RP-RBF and observed that the performance remains same as the latter. This may be because Ker-RP-RBF has been expressive enough and Cov-RP does not provide much complementary information. Nevertheless, we are confident that Ker-RP-RBF will be outperformed by combining well-designed and complementary representations.

Table 8. The result of combining complementary representations.

	MSR-A-3D	MSR-DA-3D
Cov-RP	74.0	85.0
Ker-RP-BHA	75.8	85.6
Cov-RP + Ker-RP-BHA (early fusion, proposed)	**91.2**	**96.2**
Cov-RP + Ker-RP-BHA (late fusion)	82.1	87.0

4.5. Computation time

Table 9 compares the computation time of Cov-RP, Cov-$J_{\mathcal{H}}$-SVM, and Ker-RP-RBF on all the data sets. A desktop computer with 3.6 GHz CPU and 32GB memory is used. Recall that Cov-$J_{\mathcal{H}}$-SVM does not have an explicit representation. To make fair comparison, we compare the time for computing the whole kernel matrix $\mathbf{G}$ (defined after Eq (7)) for a pair of training and test sets, which is needed by SVM classification. The value in brackets shows the time for computing the covariance or kernel representation. As seen, our kernel representation only slightly increases computation time (e.g., from 0.1 to 0.2 second), which is insignificant compared to the total time for computing $\mathbf{G}$. However, Cov-$J_{\mathcal{H}}$-SVM incurs much higher computational load, except on ETH80 which has a small number of samples, 41. In addition, on four data sets, we cannot obtain the matrix $\mathbf{G}$ by Cov-$J_{\mathcal{H}}$-SVM for a single pair of training and test sets even after 35 hours (and therefore the respective classification performance is not provided). This shows the computational efficiency of our kernel representation.

Table 9. Comparison of the time for computing the whole kernel matrix $\mathbf{G}$ used for SVM classifier training and test. The value in brackets is the time used to compute the covariance or the proposed kernel representation. (Unit: second)

Data set	Cov-RP	Cov-$J_{\mathcal{H}}$-SVM [7]	Ker-RP-RBF (Proposed)
MSR-A-3D	61.4 (0.1)	349	65.9 (0.2)
MSR-DA-3D	20.6 (0.2)	6.4×10^3	22.5 (0.3)
MSRC-12	1.3×10^3 (0.6)	3.3×10^4	1.3×10^3 (1.2)
HDM05(14)	11.4 (0.1)	1.8×10^3	15.6 (0.2)
HDM05	884.6 (0.9)	$>$ 35 hours	1037(1.6)
ETH80	28.0 (0.1)	6.5	27.9 (0.2)
CMU MoBo	21.6 (0.2)	74.0	20.4 (0.3)
YouTube	549 .0 (0.7)	898	546.9 (1.3)
Brodatz	1.4×10^3 (6.5)	$>$ 35 hours	1.4×10^3 (22.1)
FERET	109.6 (1.8)	$>$ 35 hours	110.7 (5.5)
ETH80	299.8 (2.9)	$>$ 35 hours	302.3 (9.1)

5. Conclusion

To address the new issues encountered by covariance representation, we propose to use kernel matrix as a generic feature representation. This new representation models more sophisticated feature relationship, is more robust against sample scarcity, and maintains computational efficiency. The significant improvement achieved by this representation is verified on a variety of tasks. The future work will gain more insight on the learned representations, for example, by visualizing them, and analyze the sensitivity of this representation to the number of samples in depth. Also, with the verified performance, several research issues on this new representation are worth exploring, including automatically choosing and designing appropriate kernels, its unsupervised learning methods, and the applications to more visual tasks.

References

[1] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache. Geometric means in a novel vector space structure on symmetric positive-definite matrices. *SIAM J. Matrix Analysis Applications*, 29(1):328–347, 2006.

[2] C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, 1998.

[3] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1-3):131–159, 2002.

[4] G. E. Fasshauer. Positive definite kernels: past, present and future. *Dolomites Research Notes on Approximation*, 4(Special Issue on Kernel Functions and Meshless Methods):21–63, 2011.

[5] K. Guo, P. Ishwar, and J. Konrad. Action recognition using sparse representation on covariance manifolds of optical flow. In *AVSS, Boston, MA, USA, August 29 - September 1, 2010*, pages 188–195. IEEE, 2010.

[6] M. T. Harandi, M. Salzmann, and R. Hartley. From manifold to manifold: geometry-aware dimensionality reduction for spd matrices. In *Computer Vision–ECCV 2014*, pages 17–32. Springer, 2014.

[7] M. T. Harandi, M. Salzmann, and F. M. Porikli. Bregman divergences for infinite dimensional covariance matrices. In *CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 1003–1010. IEEE, 2014.

[8] M. T. Harandi, C. Sanderson, R. Hartley, and B. C. Lovell. Sparse coding and dictionary learning for symmetric positive definite matrices: A kernel approach. In *ECCV*, pages 216–229. Springer, 2012.

[9] S. Haykin. *Neural Networks: A Comprehensive Foundation (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2007.

[10] M. E. Hussein, M. Torki, M. A. Gowayyed, and M. El-Saban. Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In *IJCAI 2013, Beijing, China, August 3-9, 2013*, 2013.

[11] R. Kondor and T. Jebara. A kernel between sets of vectors. In *ICML, August 21-24, 2003, Washington, DC, USA*, pages 361–368, 2003.

[12] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *CVPRW*, pages 9–14. IEEE, 2010.

[13] O. Oreifej and Z. Liu. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *CVPR*, pages 716–723. IEEE, 2013.

[14] Y. Pang, Y. Yuan, and X. Li. Effective feature extraction in high-dimensional space. *IEEE TSMC, Part B*, 38(6):1652–1656, 2008.

[15] Y. Pang, Y. Yuan, and X. Li. Gabor-based region covariance matrices for face recognition. *IEEE TCSVT*, 18(7):989–993, 2008.

[16] X. Pennec, P. Fillard, and N. Ayache. A riemannian framework for tensor computing. *IJCV*, 66(1):41–66, 2006.

[17] B. Póczos, L. Xiong, D. J. Sutherland, and J. G. Schneider. Nonparametric kernel estimators for image classification. In *CVPR 2012, Providence, RI, USA, June 16-21, 2012*, pages 2989–2996, 2012.

[18] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. In *CVPR, 17-22 June 2006, New York, NY, USA*, pages 728–735. IEEE, 2006.

[19] S. Sra. Positive definite matrices and the s-divergence. *arXiv:1110.1773 [math.FA]*.

[20] H. Tabia, H. Laga, D. Picard, and P. H. Gosselin. Covariance descriptors for 3d shape matching and retrieval. In *CVPR, Columbus, OH, USA, June 23-28, 2014*, pages 4185–4192. IEEE, 2014.

[21] G. W. Taylor, G. E. Hinton, and S. T. Roweis. Modeling human motion using binary latent variables. In *NIPS 2006*, pages 1345–1352, 2006.

[22] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *ECCV 2006, Part II*, pages 589–600, 2006.

[23] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *CVPR, 18-23 June 2007, Minneapolis, Minnesota, USA*. IEEE, 2007.

[24] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, pages 588–595. IEEE, 2014.

[25] C. Wang, Y. Wang, and A. L. Yuille. An approach to pose-based action recognition. In *CVPR*, pages 915–922. IEEE, 2013.

[26] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Learning actionlet ensemble for 3d human action recognition. *IEEE TPAMI*, 36(5):914–927, 2014.

[27] R. Wang, H. Guo, L. S. Davis, and Q. Dai. Covariance discriminative learning: A natural and efficient approach to image set classification. In *CVPR, RI, USA, June 16-21, 2012*, pages 2496–2503. IEEE, 2012.

[28] Y. Wu, B. Ma, and Y. Jia. Differential tracking with a kernel-based region covariance descriptor. *Pattern Anal. Appl.*, 18(1):45–59, 2015.

[29] X. Yang and Y. Tian. Super normal vector for activity recognition using depth sequences. In *CVPR*, pages 804–811. IEEE, 2014.

[30] C. Yuan, W. Hu, X. Li, S. J. Maybank, and G. Luo. Human action recognition under log-euclidean riemannian metric. In *ACCV 2009, Part I*, pages 343–353, 2009.

[31] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *ICCV*, pages 2752–2759. IEEE, Dec 2013.

[32] Y. Zhu, W. Chen, and G. Guo. Fusing spatiotemporal features and joints for 3d action recognition. In *CVPRW*, pages 486–491. IEEE, 2013.