Fast Calculation of Covariance Matrices for Arbitrary Size Cuboids

Jian Wang, Chenbin Zhang, Zhiling Wang, Junping Xiang, Zonghai Chen
Department of Automation, University of Science and Technology of China
Hefei, China
wj0910@mail.ustc.edu.cn, {zhangchb, zlwang3}@ustc.edu.cn,
junping@mail.ustc.edu.cn, chenzh@ustc.edu.cn

***Abstract*—We propose a fast calculation method to extract feature covariance matrix of cuboid of arbitrary size and arbitrary location within a given video. Every pixel in a video has several features, such as coordinates, color, gray value, gradients and orientation. Feature covariance matrix can be used to describe a video cuboid. By taking advantage of the spatio-temporal arrangement of pixels using integral video, any cuboid sum can be computed in constant time. We follow a similar idea for fast calculation of cuboid's covariance matrices. We construct integral videos for all separate features as well as integral videos of the multiplication of any two feature combinations. Using this set of integral videos we can expedite the search process more than hundreds of times in comparison to the existing conventional approach.**

Keywords- fast algorithm; feature covariance matrix; integral videos; video analysis

I. INTRODUCTION

Covariance is an essential statistical measure of how much the deviation of some variables match. If two variables both increase, it is positive; if one variable increases while the other decreases, it is negative, and if they are independent of each other, it is zero. In an image every pixel has some features, such as coordinates, RGB/gray values, gradients and orientations, and optical flow. Tuzel et al. [1] proposed a fast algorithm about construction of covariance matrices for arbitrary size image windows. References [1-3] used the feature covariance matrices as region descriptors to recognize objects, classify textures, track, detect pedestrians, etc., and get good results.

In video analysis area, there exists some cuboid descriptors [4]. Yuan et al. [5] divided the current descriptors into four kinds—(1) flattening [6], (2) global histogramming [7, 8], (3) local histogramming, e.g. HOG3D [9], 3DSIFT [10], 3DSURF [11] etc., or (4) feature covariance matrix [5]. The feature covariance matrix descriptor is significantly different from other descriptors because it utilizes the statistical property of the cuboid.

There are several advantages of using covariance matrices as descriptors. (1) The covariance matrix not only contains the statistical measure of one feature itself but also contains the correlation measures of any two features. (2) The noise is largely filtered out by the average filter during covariance computation. (3) The size of the covariance matrix is only dependent on the number of selected features and has nothing to do with the size of the image region. Thus it enables comparing regions of any size. (4) Its dimension is low.

This work is partially supported by National Natural Science Foundation of China (Grant No. 61005091) and Specialized Research Fund for the Doctoral Program of Higher Education (Grant No. 20093402110014).

In spite of its advantages, computation of the covariance matrices using a conventional approach (as given in Alg. 1) for all possible cuboids within a given video is computationally prohibitive. However, there exist a high number of overlaps between those cuboids and the statistical moments extracted from such overlapping areas can be used to reduce the computational load.

Algorithm 1: Conventional approach to compute the cuboids' covariance matrices

for each possible cuboid
  for each feature *i*
    -Compute the mean;
    -Compute the difference between the mean and each feature value *i*;
  for each feature *m* and *n*
    -Multiply two differences of each point;
    -Compute the mean of the products of all points to get the covariance coefficient (*m*, *n*).

In the 2D image analysis area, Tuzel et al. [1] took advantage of the spatial arrangement of image points using integral images and significantly improved the speed of the covariance computation. Here, we extend their work to the 3D video analysis as many cuboid descriptors [6-11] are extended from image descriptors. We propose a computationally superior method for extracting the covariance matrices of all possible cuboids based on integral video formulation.

In the next section, we describe the cuboid descriptor. In Section Ⅲ, we discuss the integral video based covariance matrices extraction formulation in detail. Then, we give a computational complexity analysis in Section Ⅳ and experimental results by considering different scenarios in Section Ⅴ. Finally we present a conclusion in Section Ⅵ.

II. COVARIANCE AS A CUBOID DESCRIPTOR

A. Covariance Matrix

We denote the observed video as V_{origin}, where it might be one dimensional intensity video or three dimensional color video, or four dimensional combination of color and infrared videos, etc. Let F be the $W \times H \times T \times d$ dimensional feature video extracted from V_{origin}

$$F(x, y, t) = \Phi(V_{origin}, x, y, t) \quad (1)$$

where the function Φ can be any mapping such as color, gradients and edge magnitude.

For a given $w \times h \times t$ cuboid $V \subset F$, let $\{f_k\}_{k=1...n}$ be the d-dimensional feature vectors inside V. We construct the feature vector f_k using two types of maps: spatial-temporal attributes that are obtained from pixel coordinate values, and appearance attributes, i.e., color, gray value, gradient, optical flow, etc. These features may be associated directly to the pixel coordinates

$$f_k = [\, x \;\; y \;\; t \;\; I \;\; I_x \;\; I_y \;\; I_t \;\; \ldots \,]. \tag{2}$$

Alternatively, they can be arranged in radially symmetric relationship i.e. we use the distance from the cuboid center as a feature replacing the coordinate feature (x, y, t).

We represent V with a $d \times d$ covariance matrix C_V of the feature points as

$$C_V = \frac{1}{n-1}\sum_{k=1}^{n}(f_k - \mu_V)(f_k - \mu_V)^T \tag{3}$$

where $n = w \times h \times t$ and $\mu_V = \frac{1}{n}\sum_{k=1}^{n} f_k$. The covariance matrix is a symmetric matrix where its diagonal entries represent the variance of each feature and the non-diagonal entries represent their respective correlations and only the upper triangular part is needed to be stored.

B. *Distance Measure*

Förstner [12] and Arsigny [13] proposed two different approaches to measure the dissimilarity of two covariance matrices. Förstner's measure is:

$$\rho_1(C_1, C_2) = \sqrt{\sum_{i=1}^{n} \ln^2 \lambda_i(C_1, C_2)} \tag{4}$$

where $\{\lambda_i\,(\,C_1, C_2\,)\}_{i=1...n}$ are the generalized eigen values of C_1 and C_2, computed from

$$\lambda_i C_1 x_i = C_2 x_i\,, \quad i = 1...d \tag{5}$$

and $x_i \neq 0$ are the generalized eigenvectors. While Arsigny's measure is

$$\rho_2(C_1, C_2) = \sqrt{Trace(\{\log(C_1) - \log(C_2)\}^2)}\ . \tag{6}$$

Let $\Sigma = UDU^T$ be the eigen value decomposition of a symmetric matrix. The logarithm in (6) is given by

$$\log(\Sigma) = \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k}(\Sigma - I)^k = U \log(D) U^T\ . \tag{7}$$

Förstner and Arsigny also gave the related approaches about calculating the mean of a set of covariance matrices.

C. *Application*

In this part we show the efficiency of the covariance matrix as a cuboid descriptor. We selected a video clip from the Weizmann human action dataset [14] (illustrated in Fig. 1). The cuboid with red lines was selected as the sample. We used feature covariance matrix to describe the cuboids. The selected features were coordinates, gray value, gradients, namely

$$f_k = [\, x \;\; y \;\; t \;\; I \;\; I_x \;\; I_y \;\; I_t \,]. \tag{8}$$

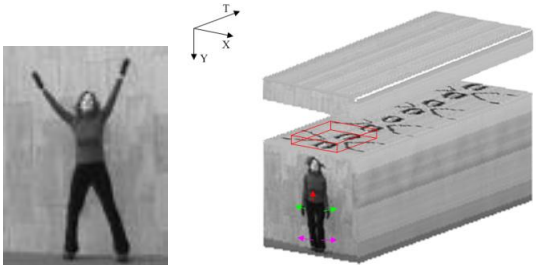


Figure 1. The image on left is a frame within a video in which a woman was jumping jack; the right one is a video solid. The colored arrows indicate the action directions of her body parts.

The size of the sample was 39×11×23. We calculated the covariance matrices of each cuboid of the same size within the video solid and measured the distances to the sample's covariance matrix using equation (4) and (6).

Fig. 2(b) showed the distances between the sample and the cuboids whose projections from XYT space to XY plane were location a~ f (illustrated in Fig. 2(a)), and the cuboids whose projections were location a, were measured by equation (4) and (6) and the others by equation (4). The experimental results showed that the distances between the sample and cuboids whose projections were location b, c, d, were at least 1.3. And the distances between the sample and cuboids whose projections were location a, were smaller (curve "a by [12]" and curve "a by [13]" in Fig. 2(b)). As the woman did the action for four periods, curve "a by [12]" had four valley floors and every time when the curve reached the valley the woman was doing the same action primitive as the sample cuboid represented. It showed that the covariance matrix can act as a cuboid descriptor well.

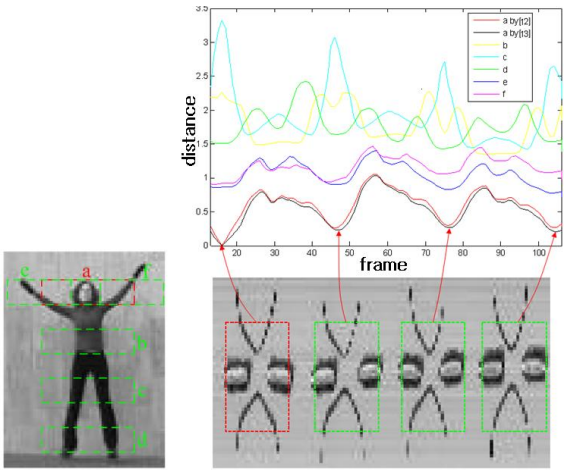


Figure 2. (a) "a"~"f" are the projections of cuboids to be compared with the sample from XYT space to XY plane. The sample's projection is location a. (b) Comparison of different cuboids to the sample one by measuring their covariance matrices. The low image is in the X-T cutting plane.

III. Integral Videos For Fast Covariance Calculation

Viola and Jones [15] showed that it was possible to calculate the sum of the values within rectangular windows in

linear time by integral image. Later, this idea was extended to fast calculation of region histograms [16] and fast construction of covariance matrices for image windows [1]. We use an "integral video" data structure to expedite calculation of covariance matrices of cuboids. It can be seemed as a direct spatio-temporal generalization of the integral image based approach described in [1].

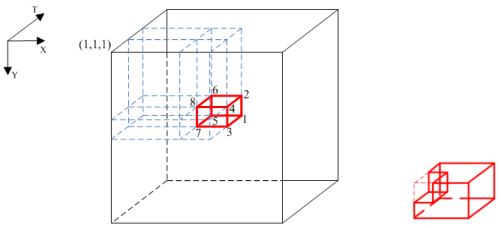
Let $F_{feature}$ be the $W\times H\times T\times 1$ dimensional feature video extracted from F. An integral video at pixel location (x, y, t) is defined as the sum of all pixels at location less than or equal to (x, y, t):

$$IntF_{feature}(x,y,t)=\sum_{x'\le x}\sum_{y'\le y}\sum_{t'\le t}F_{feature}(x',y',t') \qquad (9)$$

where $F_{feature}(x', y', t')$ is the pixel feature value at location (x', y', t'). $IntF_{feature}$ can be easily computed using the following recurrences:

$$\begin{cases} s_1(x,y,t)=s_1(x-1,y,t)+F_{feature}(x,y,t) \\ s_2(x,y,t)=s_2(x,y-1,t)+s_1(x,y,t) \\ IntF_{feature}(x,y,t)=IntF_{feature}(x,y,t-1)+s_2(x,y,t) \end{cases} \qquad (10)$$

where $s_1(x, y, t)$ is the cumulative row sum, $s_2(x, y, t)$ is the integral image at frame t, and $s_1(0, y, t)= s_2(x, 0, t)= IntF_{feature}(x, y, 0) \equiv 0$. The integral video is illustrated in Fig. 3.



(a) (b)

Figure 3. (a) Integral video. The sum within red cuboid can be computed from the integral video with eight array references: 1-(2+3+5)+4+6+7-8. If the computed volume is a cuboid like (b), the sum is the difference between two cuboids.

We can rewrite the (i, j)th element of the covariance matrix defined in (3) as

$$\begin{aligned} C_V(i,j)&=\frac{1}{n-1}\sum_{k=1}^{n}(f_k(i)-\mu_V(i))(f_k(j)-\mu_V(j)) \\ &=\frac{1}{n-1}[\sum_{k=1}^{n}f_k(i)f_k(j)-n\mu_V(i)\mu_V(j)] \\ &=\frac{1}{n-1}[\sum_{k=1}^{n}f_k(i)f_k(j)-\frac{1}{n}\sum_{k=1}^{n}f_k(i)\sum_{k=1}^{n}f_k(j)] \end{aligned} \qquad (11)$$

To find the covariance matrix in a given cuboid V, we have to compute the sum of each feature dimension, $f(i)_{i=1\ldots d}$, as well as the sum of the multiplication of any two feature dimensions, $f(i)f(j)_{i,\ j=1\ldots d}$. It is possible to compute these sums with a few arithmetic operations using a series of integral videos. We construct integral videos for each feature dimension $f(i)_{i=1\ldots d}$ and multiplication of any two feature dimensions $f(i)f(j)_{i,\ j=1\ldots d}$. As a result we construct $d+C_{d+1}^2$ integral videos. Take (8) for example: in (8) the number of selected features is seven, so we construct seven integral videos for each feature dimension and C_8^2 integral videos for multiplication of any two feature dimensions.

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

$$P(x',y',t',i)=\sum_{x\le x',y\le y',t\le t'}F(x,y,t,i) \qquad i=1\ldots d \qquad (12)$$

and Q be the $W\times H\times T\times d\times d$ tensor of the second-order integral videos

$$Q(x',y',t',i,j)=\sum_{x\le x',y\le y',t\le t'}F(x,y,t,i)F(x,y,t,j) \qquad i,j=1\ldots d \cdot \qquad (13)$$

In (10) it shows that the integral video can be computed in one pass over the video. Next we represent $[P(x,y,t,1) \ldots P(x,y,t,d)]^T$ as $P_{x,y,t}$ and represent

$$\begin{pmatrix} Q(x,y,t,1,1) & \ldots & Q(x,y,t,1,d) \\ & \vdots & \\ Q(x,y,t,d,1) & \ldots & Q(x,y,t,d,d) \end{pmatrix} \text{ as } Q_{x,y,t}.$$

Let $V(x', y', t'; x'', y'', t'')$ be the cuboid, where (x', y', t') is point 8's coordinate and (x'', y'', t'') is the point 1's, as shown in Fig. 3. The covariance of the cuboid $V(x', y', t'; x'', y'', t'')$ can be computed as

$$\begin{aligned} C_{V(x',y',t';x'',y'',t'')}=&\frac{1}{n-1}[Q_{x'',y'',t''}-Q_{x'',y'',t'-1}-Q_{x'',y'-1,t''}-Q_{x'-1,y'',t''} \\ &+Q_{x'-1,y'-1,t''}+Q_{x'-1,y'',t'-1}+Q_{x'',y'-1,t'-1}-Q_{x'-1,y'-1,t'-1} \\ &-\frac{1}{n}(P_{x'',y'',t''}-P_{x'',y'',t'-1}-P_{x'',y'-1,t''}-P_{x'-1,y'',t''} \\ &+P_{x'-1,y'-1,t''}+P_{x'-1,y'',t'-1}+P_{x'',y'-1,t'-1}-P_{x'-1,y'-1,t'-1}) \\ &\cdot(P_{x'',y'',t''}-P_{x'',y'',t'-1}-P_{x'',y'-1,t''}-P_{x'-1,y'',t''} \\ &+P_{x'-1,y'-1,t''}+P_{x'-1,y'',t'-1}+P_{x'',y'-1,t'-1}-P_{x'-1,y'-1,t'-1})^T] \end{aligned} \qquad (14)$$

where $n=(x''-x'+1)(y''-y'+1)(t''-t'+1)$ and $P_{0,y,t}=P_{x,0,t}=P_{x,y,0}\equiv[\underbrace{0 \ \ldots \ 0}_{d}]$ and $Q_{0,y,t}=Q_{x,0,t}=Q_{x,y,0}\equiv\begin{pmatrix} 0 & \ldots & 0 \\ & \vdots & \\ 0 & \ldots & 0 \end{pmatrix}_{d\times d}$. We give the integral video based covariance computation in Alg. 2.

Algorithm 2: Integral video based approach

for each feature i
  -Accumulate integral video $P(i)$
for each feature i and j
  -Accumulate integral video $Q(i, j)$
for each possible cuboid
  -Get the eight corners of the cuboid;
  -Apply Eq. 14.

IV. COMPLEXITY ANALYSIS

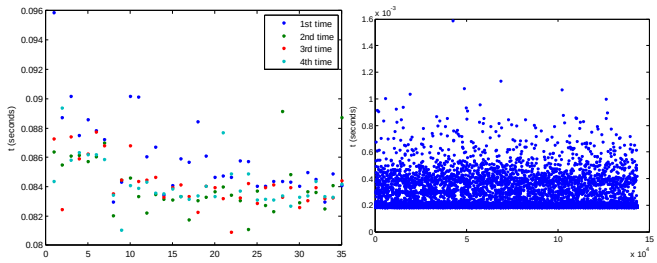
The covariance matrices are low dimensional compared to other cuboid descriptors and due to symmetry C_V has only $(d^2+d)/2$ distinct values. Whereas if we represent the same region with the feature values we need nd values, where $n=w\times h\times t$ is the number of pixels inside the cuboid. A conventional histogram representation with h-quantization levels per feature channel would require h^d-bins. Usually it is the case that $n \gg h \gg d$.

As mentioned before, it is possible to reduce the complexity of covariance computation using integral video techniques. The computational complexity of constructing integral videos is $O(d^2WHT)$. Using the constructed representation, the covariance matrix of any cuboid can be computed using $O(d^2)$ arithmetic operations.

V. EXPERIMENTS

Here we compare the conventional method and the integral video based method by their speed. The run-time estimates were obtained on an AMD Athlon (tm) Ⅱ X4 640 core PC with 3.01 GHz processors and 1.75GB RAM with a MATLAB implementation.

In Section Ⅱ.C, we needed to calculate 35 integral videos in all, sizes of which were 66×88×117. Fig. 4(a) shows the CPU times for the construction (4 times). We can see that constructing an integral video with the size 66×88×117 can take 0.08~0.1 seconds. For the reason that the constructions of each integral video are independent of each other, it will take less time to compute them by parallel computing. After calculation of the integral videos, calculation of covariance matrices for arbitrary size cuboids only takes several milliseconds (Fig. 4(b)).

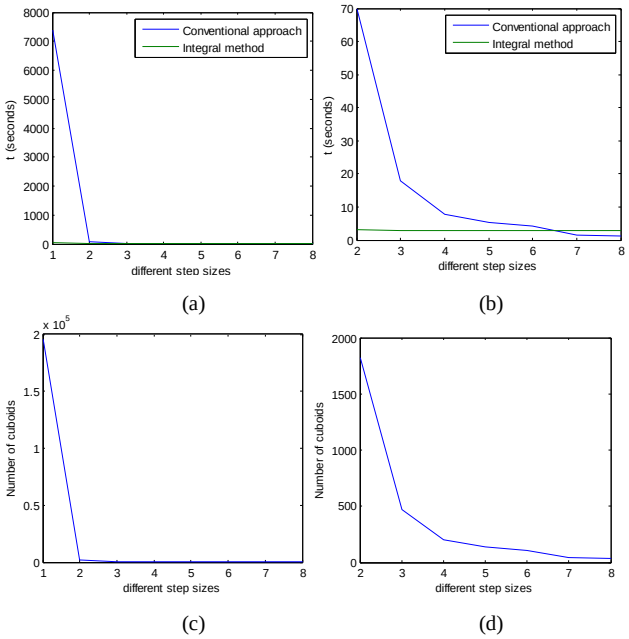


(a) (b)

Figure 4. (a) Computation times for 35 integral videos sizes of which are 66×88×117. (b) Computation times for covariance matrices of arbitrary size cuboids (143190 times).

A. Experiment 1

We fixed the size of cuboids to 39×11×23 and the size of video to 66×88×117. Computation times for different step sizes (1-1-1, 5-5-5, 10-6-8, 15-7-11, 20-8-14, 25-9-17, 30-10-20 and 39-11-23) of two methods are shown in Fig. 5(a). We observed that when the step size was set to 1-1-1namely grid search, the proposed method took 38.35s while the conventional method took 7399s (about two hours). When the step size was set to 5-5-5, the proposed method took 3.24s while the conventional method took 69.16s. In fact, in most circumstances the search size of step is not larger than 5-5-5. We can also find that the run-times of conventional approach are directly proportional to the number of computed cuboids. Note that when the step size was set to 30-10-20 or larger the conventional approach ran faster than integral approach and at this time the number of computed cuboids was smaller than 40.



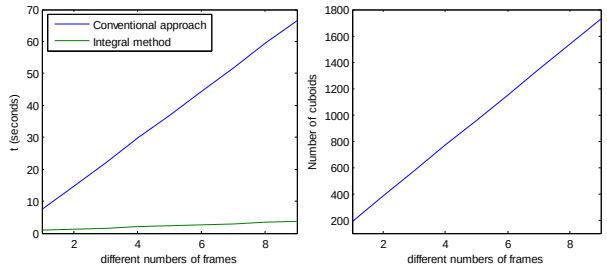
(a) (b)

(c) (d)

Figure 5. (a) Comparison with size of cuboids and size of video fixed but with different step sizes; (b) the partially zoomed graph of (a); (c) Number of computed cuboids changes with step size; (d) the partially zoomed graph of (c).

B. Experiment 2

We fixed the size of cuboids to 39×11×23 and the size of step to 5-5-5. Computation times for different numbers of frames (30, 40, 50, 60, 70, 80, 90, 100 and 110) of two methods are shown in Fig. 6(a). Note that the height and length of the frames of the videos were set to 66 and 88 respectively. We observed that when the number of frames was set to 110, the proposed method took 3.61s while the conventional method took 66.59s. Note the run-times of conventional approach are proportional to the number of computed cuboids.

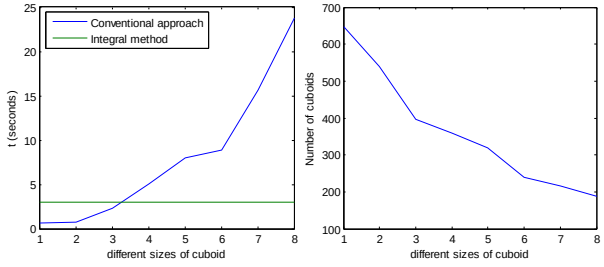


(a) (b)

Figure 6. (a) Comparison with size of cuboids and size of step fixed but with different number of frames; (b) Number of computed cuboids changes with numbers of frames.

C. *Experiment 3*

We fixed the size of video to 66×88×117 and the size of step to 10×10×10. Computation times for different sizes of cuboids (15×3×5, 21×3×5, 27×5×11, 31×7×17, 35×9×21, 39×11×23, 41×15×31 and 45×21×35) of two methods are shown in Fig. 7(a). We observed that when the size of cuboid was set to larger than 27×5×11, the proposed method ran faster than the conventional approach. Note that although the number of cuboids descended, the run-time of conventional approach arose. The larger the size of cuboid is, the more time the conventional method will take.

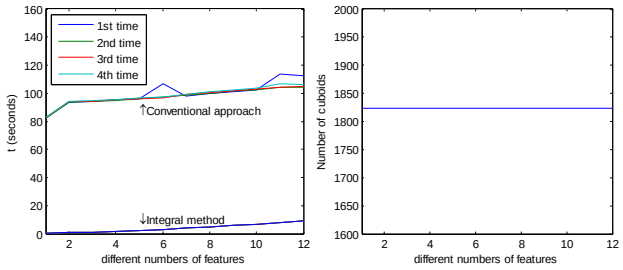


(a) (b)

Figure 7. (a) Comparison with number of frames and size of step fixed but with different sizes of cuboids; (b) Number of computed cuboids changes with sizes of cuboid.

D. *Experiment 4*

We fixed the size of video to 66×88×117, the size of step to 5×5×5 and the size of cuboids to 39×11×23 but changed the numbers of features, from 1 to 13. Computation times for different numbers of features of two methods are shown in Fig. 8(a). We observed that when the number of features increased, both methods would take more time, but the proposed method's run-time was comparatively smaller than the conventional methods. Generally the number of selected features is smaller than eleven, e.g. Porikli [2] selected five features for tracking, Tuzel [1] used five features for texture analysis and nine features for object detection, Yuan [5] used eight features for human action recognition and Tuzel [3] used nine features for pedestrian detection.



(a) (b)

Figure 8. (a) Comparison with the numbers of features changed (4 times); (b) Number of computed cuboids changes with the numbers of features.

VI. CONCLUSION

We presented a novel and fast algorithm to compute the covariance matrices of all possible cuboids in a video. Our experimental results show that the integral video based method can expedite the search process more than several hundreds of times in comparison to the existing conventional approaches. We argue that the merit of integral video based approach to video analysis is more than integral image based approach's to image analysis to some degree since videos' data are much larger than images generally. Additionally it makes greedy search or point-wise search possible.

ACKNOWLEDGMENT

The authors would like to thank National Natural Science Foundation of China and Specialized Research Fund for the Doctoral Program of Higher Education for supporting this study.

REFERENCES

[1] O. Tuzel, F. Porikli, and P. Meer, "Region covariance: A fast descriptor for detection and classification," in *The 9th European Conference on Computer Vision*, 2006, pp. 589-600.

[2] F. Porikli, O. Tuzel, and P. Meer, "Covariance tracking using model update based on lie algebra," in *19th IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 728-735.

[3] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on riemannian manifolds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, pp. 1713-1727, 2008.

[4] H. Wang, M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *The 20th British Machine Vision Conference* 2009, pp. 1-11.

[5] C. Yuan, W. Hu, X. Li, S. Maybank, and G. Luo, "Human action recognition under log-euclidean riemannian metric," in *The 9th Asian Conference on Computer Vision*, 2009, pp. 343-353.

[6] J. C. Niebles, H. C. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *International Journal of Computer Vision*, vol. 79, pp. 299-318, Sep 2008.

[7] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *2nd Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2006, pp. 65-72.

[8] I. Laptev and T. Lindeberg, "Local descriptors for spatio-temporal recognition," *Spatial Coherence for Visual Motion Analysis*, pp. 91-103, 2006.

[9] A. Kläser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *The 19th British Machine Vision Conference*, 2008, pp. 995-1004.

[10] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *The 15th ACM International Conference on Multimedia*, 2007, pp. 357-360.

[11] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *The 10th European Conference on Computer Vision*, 2008, pp. 650-663.

[12] W. Förstner and B. Moonen, "A metric for covariance matrices," Dept. of Geodesy and Geoinformatics, Stuttgart Univ.1999.

[13] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Geometric means in a novel vector space structure on symmetric positive-definite matrices," *SIAM Journal on Matrix Analysis and Applications*, vol. 29, pp. 328-347, 2006.

[14] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *10th IEEE International Conference on Computer Vision*, 2005, pp. 1395-1402.

[15] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2001, pp. 511-518.

[16] F. Porikli, "Integral histogram: A fast way to extract histograms in cartesian spaces," in *18th IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 829-836.