Position-Based Action Recognition Using High Dimension Index Tree

Qian Xiao
Shenzhen Institutes of Advanced Technology, CAS
The Chinese University of Hong Kong
Shenzhen, P. R. China
E-mail: qian.xiao@siat.ac.cn

Jun Cheng
Shenzhen Institutes of Advanced Technology, CAS
The Chinese University of Hong Kong
Guangdong Provincial Key Laboratory of Robotics and Intelligent System
Shenzhen, P. R. China
E-mail: jun.cheng@siat.ac.cn

Jun Jiang[1,2,3], *Wei Feng*[1]
[1]Shenzhen Institutes of Advanced Technology, CAS
[2]The Chinese University of Hong Kong
[3]The Shenzhen Key Laboratory of Computer Vision and Pattern Recognition
Shenzhen, P. R. China
E-mail: {jun.jiang, wei.feng}@siat.ac.cn

***Abstract* – Most current approaches in action recognition face difficulties that cannot handle recognition of multiple actions, fusion of multiple features, and recognition of action in frame by frame model, incremental learning of new action samples and application of position information of space-time interest points to improve performance simultaneously. In this paper, we propose a novel approach based on Position-Tree that takes advantage of the relationship of the position of joints and interest points. The normalized position of interest points indicates where the movement of body part has occurred. The extraction of local feature encodes the shape of the body part when performing action, justifying body movements. Additionally, we propose a new local descriptor calculating the local energy map from spatial-temporal cuboids around interest point.**

In our method, there are three steps to recognize an action: (1) extract the skeleton point and space-time interest point, calculating the normalized position according to their relationships with joint position; (2) extract the LEM (Local Energy Map) descriptor around interest point; (3) recognize these local features through non-parametric nearest neighbor and label an action by voting those local features. The proposed approach is tested on publicly available MSRAction3D dataset, demonstrating the advantages and the state-of-art performance of the proposed method.

Index Terms - Action Recognition, Depth Maps, Feature Fusion, Incremental Recognition

I. INTRODUCTION

Action recognition is a challenging area in computer vision due to its difficulty of handling large variations in body movement, posture and body size within the same action class. Traditional action researches are mainly based on appearances facing dilemmas such as cluttered background, illumination change and body tracking. The release of kinect renders the possibility of providing real-time depth images that is not very sensitive to background and illumination. Very recently, [1] provided a practical human motion capturing technique that outputs 3D joint positions of human skeleton, which is significant for action recognition, however, traditional RGB camera is hard to provide it. Fig. 1 illustrates the depth maps of MSRAction3D dataset for various actions including Tennis serve and Pickup & throw.

Most available approaches are not yet applicable to real problems. One reason is that they are based on a static training dataset, therefore not scalable. Another reason is

(a) Tennis serve

(b) Pickup & throw

Fig. 1. Examples of the sequences of MSRAction3D dataset: (a) Tennis serve and (b) Pickup & throw

that they are unable to recognize multiple actions accurately. In this paper, we propose a novel method for action recognition which is motivated by the success of using index trees to index the bag of features. Reddy [2] first used SR-tree to index large scale motion feature showing many advantages over bag-of-words model. Due to the ability to extend the tree structure, it's more appropriate for incremental action recognition. Compared to updating the visual vocabulary and computing histograms for bag-of-words method, the method of feature-tree need not time-consuming training stage and can effectively and efficiently integrate indexing and recognition stage. However, there are some factors that Reddy's [2] work cannot handle successfully, which includes (1) the inability to utilize the position of interest point to improve the recognition performance; (2) the inability to fuse multiple features; (3) the difficulty of recognizing multiple actions accurately since they only simply rank the feature queries and compare the votes to a predefined threshold which is difficult to decide.

To overcome these problems, we propose a new approach that uses the Position-Tree to query where the movement occurs and recognize body part movements. For the movement within the same action class, the position of interrelated interest point generally occurs in the similar area. In order to distinguish different actions, we need to encode the variations of the movement. Therefore, we propose a new local descriptor that accumulates the energy of space-time cuboid. The local energy map encodes the variations of adjacent frame and calculates the HOG descriptor from energy map to form the descriptor vector. The experiment results

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.753

show our descriptor's exceptional performance of computational complexity and average accuracy.

II. Related word

Actions can be considered as spatial-temporal pattern. There are two crucial issues in action recognition: discriminate and robust feature and effective recognition framework to recognize actions.

In the literature, some approaches model the dynamical patterns using finite state models [3] or Hidden Markov model [4] [5] for a number of static features, such as silhouettes [6] , edges [7] . However, a lot of training data are needed to build the complex generative model since the limited amount of training data for model generation is easy to overfit.

Other approaches apply space-time interest points and their trajectories for action recognition [8] [9] [10] [11] [12]. In these methods, actions are usually represented as a set of visual words that is the cluster of space-time features. Based on bag-of-word model, there are mainly two-stage procedures that contain vocabulary construction and category model. The advantages of bag-of-word model are the rather simple framework of modelling and the fast recognition process. The drawbacks of this model are the inability for incremental action recognition and the intensive training stage.

More recently, as RGBD sensors become available, research of human action recognition based on depth map has raised interest. The first work using depth camera for action recognition is described in [13] where Li efficiently sampled a bag of 3D points with three view from the depth maps and used the Gaussian mixture models to model human postures. Motivated by [1] that outputs 3D joint positions of human skeleton, many researches began to utilize the 3D joint which is hard to provide in RGB camera. Yang [14] proposed a type of features based on position difference of joints which combines action information of static posture, motion and offset to recognize actions. Xia [15] used histograms of 3D joint locations as a representation of postures to recognize actions. Wang [16] utilized a sparse coding to encode random occupancy pattern features that employ a sampling scheme which can effectively explore an extremely large sampling space. Yang [17] generated HOG descriptor from Depth Motion Maps that can effectively encode global motion information.

In this paper, we propose a recognition method that uses the Position-tree to query nearest normalized interest point and compare the local spatial-temporal features to recognize an action. The key contributions of this paper can be summarized as follows: (1) Intuitively, in order to recognize an action, we should firstly know where the movement has happened. So we propose a new tree structure called Position-tree consisted of the normalized position of interest points. There are some advantages of our approach, which includes:

1. fusion of multiple features.
2. recognition of multiple actions.
3. incremental learning of actions.
4. recognition of action frame by frame.

(2) We propose a simple but effective local space-time feature that accumulates the energy of adjacent frames. Its computational complexity is low. Our method makes full use of the position of interest point and its local features to improve the accuracy of recognition.

The remainder of the paper is organized as follows. In section 2 we introduce the extraction and representation of space-time feature. In section 3 we introduce our position-tree structure and recognition method. In section 4 the experiment results and evaluation are presented. In section 5, conclusion remark and future work are provided.

III. Feature Extraction and Representation

A. *Spatial-Temporal Features*

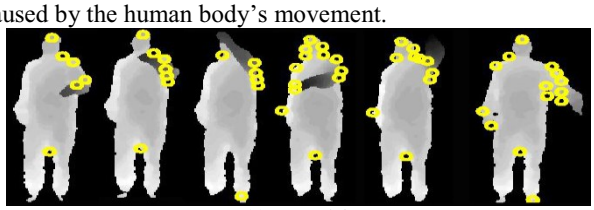
We use the interest point detector proposed by Laptev [8] which computes a spatial-temporal second-moment matrix at each point, defined as

$$\mu(.;\sigma;\tau) = g(.;s\sigma,s\tau) * (\nabla L(.;\sigma,\tau)(\nabla L(.;\sigma,\tau))^T), \quad (1)$$

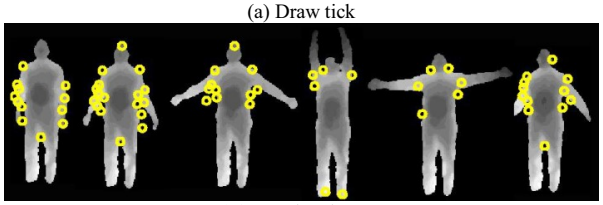
where σ, τ denote independent spatial and temporal scale values respectively, g denotes a separable Gaussian smoothing function, and ∇L denotes space-time gradients. The final locations of space-time interest points are given by local maxima of H defined as

$$H = \det(\mu) - k * trace^3(\mu), H > 0. \quad (2)$$

Our results of the detected interest points are shown in fig. 2. Though the sparse distribution of the interest point, it can actually find the local maximum of the changes that are caused by the human body's movement.



(a) Draw tick



(B) Two hand wave

Fig. 2. Detected interest points of some sample actions: (a) Draw tick, (B) Two hand wave

B. *Normalization of the interest point position*

In [1], a method to extract body joint locations from single depth image is provided. 20 skeletal joints containing hip center, spine, shoulder, center, head, L/R shoulder, L/R elbow, L/R wrist, L/R hand, L/R knee, L/R angle and L/R foot are retrieved.

Suppose the 3D coordinates of K joints are available in each frame: $J = \{J_1, J_2, ..., J_K\}$, and the position of interest point represented as P, the normalized position of that interest point N can be described as

$$N = \{\frac{P - J_i}{L} \mid i = 1,2,...,K\},$$

Where P denotes interest point's position and J_i denotes the skeletal point. L denotes body length and it can be described as:

$$L = \mid J_{HF} - J_{HipF} \mid,$$

Where J_{HF}, J_{HipF} represents the head and hip center respectively. L can be considered as the distance between human body's head and hip center when the subject stands straightly.

C. Local Spatio-Temporal Energy Map

Each spatial-temporal cuboid around interest point can be represented as $d(x, y, z, t)$. Unlike in RGB cameras, it can be projected onto three orthogonal Cartesian planes. Each 3D depth spatial-temporal cuboid can generate three 2D maps on the basis of front, side, top view. In order to get the local energy map, we compute and threshold the difference between consecutive frames, which can be represented as

$$LEM_v = \sum_{i=1}^{T-1} \left(\mid cuboid_v^{i+1} - cuboid_v^i \mid > \varepsilon \right)$$

where $V \in \{f, s, t\}$ denotes the projection view, $cuboid_v^i$ is the projected map of the ith frame under projection view V, T is the number of frames for cuboid, $\mid cuboid_v^{i+1} - cuboid_v^i \mid > \varepsilon$ is the binary map of motion energy and ε is the threshold.

We follow [17] to calculate HOG descriptor to describe the motion map which characterizes the accumulated motion distribution and intensity of this cuboid.

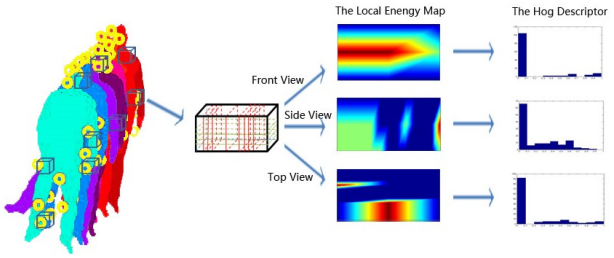


Fig. 3 Illustration of Extracting The Local Energy Map

IV. Constructing Position-Tree and Recognition

A. Constructing Position-Tree

We follow [2] to use the RS-tree to construct the Position-Tree. Note that we utilize the normalized position of interest point instead of space-time features to construct the Feature-tree. Obviously, we can query the nearest neighbor faster than [2] where the dimension of spatio-temporal feature vector is larger than our normalized position point. Additionally, we are able to fuse multiple local features around the normalized

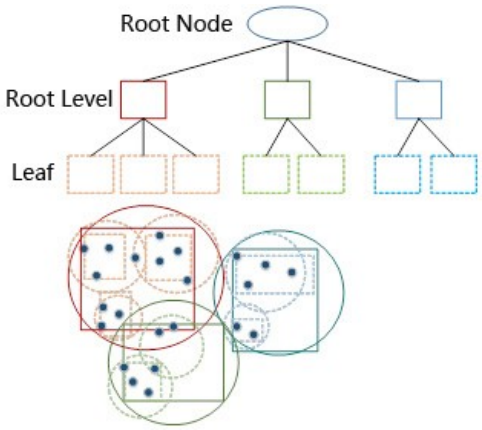


Fig. 3 The structure of RS-tree

position of interest point. The fusion of multiple features is crucial for action recognition since a single feature-base representation is not enough to capture the variants of movement. We can fuse the appearance-based and depth-base features at the same time without any effect on query speed.

Given a set of action depth maps $D = \{D_1, D_2, ..., D_M\}$ and its corresponding class label $L_c \in \{1,2,...,c\}$, we extract n spatial-temporal points and its local feature f_i^j from depth maps i, with normalized positions $N_i^j (1 \le j \le n)$. Then we can get a three-element tuple $x_i^j = [N_i^j \; f_i^j \; L_c]$. Then we collect the labeled features to construct our feature-tree.

We briefly review RS-tree data structure in Fig. 3. RS-tree successfully integrates the advantages of the R-tree and SS-tree. It's fast for large scale datasets. From Fig. 3, we can see that the region is defined by the intersection of a bounding sphere and a bounding rectangle. That helps to improve the performance of nearest neighbors search for larger region diameter and smaller region volume. In our position-tree, node contains four components: bounding sphere, bounding rectangle, numbers of normalize interest point position and pointers to all its children. The leaf stores the normalized point, its label and corresponding spatial-temporal features.

The RS-tree clusters the normalized positions which are spatially close to each other. The action within the same class would share several feature patterns. In order to incrementally learn actions, the RS-tree can grow naturally. The nearest neighbor search, insertion, splitting and deletion algorithm of RS-tree in detail can be referred in [18].

B. The Recognition Framework

The major steps of recognition phase in our method are described in Table 1. Given an unknown action video, we extract a set of spatial-temporal features from it. Suppose a video is represented by a set of normalized position of interest point $N_i^j (1 \le j \le n)$ and its related spatial-temporal features f_i^j, our recognition task is to find the nearest

normalized position point for N_i^j and compare their feature vectors. Finally we count the votes and find the maximum scores to decide the video label. For each queried normalized position N_i^j , we retrieve the k nearest points from the position tree which clusters the normalized position points spatially closed to each other. According to the k returned normalized point, we compare the spatial-temporal features and assign it a class label based on the distance of comparison between local features.

Table 1. Main steps of the action recognition framework

Objective: Given a query depth map Q, assign it a class label
(1)Extracting ST Feature. Apply Harris3D to detect the interest point and extract the LEM descriptor around the interest point.
(2)Calculating the Normalize position. According to the skeletal point, calculating the normalized position of each interest point.
(3)Query the nearest neighbour. Query the nearest normalized position point and compare their local feature vector with queried point.
(4)Recognizing action. Count the votes of K returned neighbours and find the maximum vote.

In order to recognize an action, we can assign a class label to $N_i^j (1 \le j \le n)$ using the following equation:

$$c = \arg\max_C \sum_{q=1}^{M} \sum_{r=1}^{K} \frac{\tau * I_q^r}{\| f_q - f_q^r \| + \varepsilon},$$

where I_r^q is an indicator function which is equal to 1 if the label of f_q^r is C , $M(1 \le M \le n)$ is the number of the normalized points, K is the number of nearest neighbor, and ε , τ are constant numbers (NN-Nearest Neighbor). f_q^r represents the neighbors of f_q .

When there are several local features, we can use following equation to fuse multiple features:

$$Lb = \arg\max_{Lb} \sum_{i=1}^{S} c_i * F_i ,$$

where S is the number of types of features and F_i is also an indicator function. From fusing equation, we can fuse different types of features and make simple votes to decide the final label of action.

V. Experimental Results

A. Experimental Setup

MSR Action3D dataset is a publicly available benchmarking dataset that supplies depth maps for 20 actions performed by 10 subjects performed 2 or 3 times. As illustrated in Fig. 4, actions in this dataset are complicated because of the large variation of the movements of human body. Additionally, the various styles of actions performed by different subjects also increase the difficulties of recognition. Such as Pickup & Throw, some subjects perform using only one hand whereas others using two hands, which will increase intra-class variations greatly.

Fig. 4 Examples of MSRAction3D dataset.

We follow the identical experimental setting in [13], splitting those categories into three subsets as listed in Table 2. For each subset, there are three different testing experiments. In test one ,1/3 of the subset is used as training and the rest as testing; in test two, 2/3 of the subset is used as training and the rest as testing; in cross subject test, half subjects are used as training and the rest used as testing.

Table 2. Action subsets used in the experiment

Action Set 1 (AS1)	**Action Set 2 (AS2)**	**Action Set 3 (AS3)**
Horizontal arm wave	High arm wave	High throw
Hammer	Hand catch	Forward kick
Forward Punch	Draw x	Side kick
High throw	Draw tick	Jogging
Hand clap	Draw circle	Tennis swing
Bend	Two hand wave	Tennis serve
Tennis serve	Forward kick	Golf swing
Pickup & throw	Side boxing	Pickup & throw

B. Comparisons to the State-Of-The-Art

In Table 3, we compare our method with the state of the art method [13] on the MSR Action3D dataset. As shown in the table, the average recognition rates of our method in Test One, Test Two, and Cross Subject Test are 92.3%, 97.3%, 84.9%, improving the average accuracies of [13] by 0.7%, 3.1%, 10.2%, respectively. Note that in AS3CrSub our recognition result has obvious improvement since the actions in AS3 are with significant differences. Thus, the normalized positions of interest point for different actions are also greatly different, discriminable for our method to recognize action.

TABLE 3. COMPARISON WITH BAG-OF-3D-POINTS[13]

	BAG-OF-3D-POINTS[13]	**OUR METHOD**
AS1ONE	89.5%	**90.1%**
AS2ONE	89.0%	**89.4%**
AS3ONE	96.3%	**97.4%**
AS1TWO	93.4%	**97.3%**
AS2TWO	92.9%	**97.3%**
AS3TWO	96.3%	**97.4%**
AS1CRSUB	72.9%	**81.6%**
AS2CRSUB	71.9%	**77.4%**
AS3CRSUB	79.2%	**95.7%**

We also compare our method with other very recent methods on the cross-subject test setting, where the samples of half subjects are used as training data, and the rest of the

samples are used as test data. The proposed method achieves an accuracy of 87.1%. Despite the recognition performance of our method is lower than Depth Motion Maps [14], our approach can improve the accuracy through fusing of multiple local features and has the ability to recognize an action in frame by frame model.

Table 4. Comparison with other methods on MSR Action3D dataset

Method	Accuracy
Bag-Of-3D-Points [13]	74.7%
STOP feature [19]	84.8%
Eigenjoints [14]	82.3%
Random Occupancy Patterns [16]	86.5%
Depth Motion Maps [17]	91.6%
Our Method	87.1%

C. *Incremental Action Recognition*

In realistic application, the fixed amount of training samples would not be practical. Our method can adapt to the actual requirements of adding new training samples by updating the Position-tree without reconstructing the entire tree. It's easy to insert or delete a new feature in SR-tree. In order to verify our method's incremental function, we use the MSRAtion3D data set again to show the advantages of having an extendable position-tree. In AS3, we first train our tree with five people on 7 actions (High throw, Forward kick, Side kick, Jogging, Tennis swing, Tennis serve, Golf swing). We add one person at a time from Pickup & throw action to the Position-tree and analyze its influence. The results are shown in Fig. 5. From our experiment, we can observe that the performance increases obviously as the new example is inserted into the Position-tree.

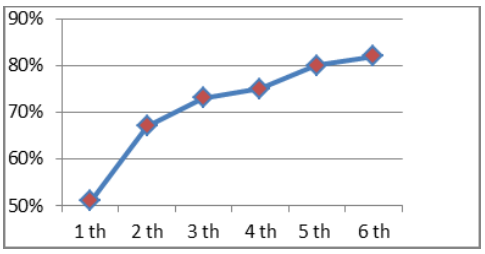


Fig. 5 shows the performance increase as the new example inserts into the Position-tree.

D. *Recognizing Multiple Actions In Video*

As mentioned above, Reddy's [2] work cannot recognize multiple actions accurately since it only simply ranks the feature queried by the class frequency and compares the frequency to a predefined threshold. However, the predefine threshold is fixed and hard to choose so the recognition performance is not very reliable. In our method, we make use of the joint information which helps us to localize the relationship of interest point and person in advance. From fig. 6, we can observe that the interest point falls into the area of human body so we can identify the interest point belongs to whomever. This property helps us easily recognize multiple people performing multiple actions simultaneously. To simulate this scenario and demonstrate our method, we record a depth map with two instances: Tennis swing and Pickup & throw happening at the same time. We extract the space-time interest point from the depth map and query the tree constructed using 8 actions and 5 people in AS3. The experiment results verify that the performance of recognizing multiple actions is totally the same as the single action.

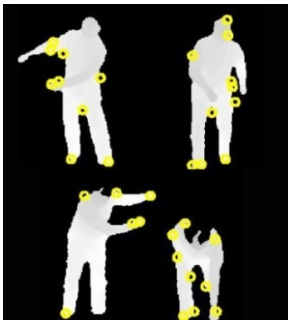


Fig. 6 recognizing two actions happening in the same video

E. *Fusing Multiple Features*

In this experiment, we show that our method can fuse multiple features to improve the overall average accuracy of recognition. HOG3D [11] is the descriptor based on gradient that is different from LEM based on accumulating the difference between consecutive frame of space-time cuboid. From table 2, we can observe that the overall average accuracy of fusing result is better that single feature of HOG3D or LEM. The performance of fusing multiple features is much more reliable. Additionally, our method can fuse any kinds of local features around interest point.

Table 5 The fusing result of multiple features. The fusing result has indeed improve the average accuracy.

	HOG3D	LEM	Fusing Result
AS1CrSub	79.6%	81.6%	82.1%
AS2CrSub	84.6%	77.4%	83.6%
AS3CrSub	85.3%	95.7%	95.7%

F. *Recognizing The Action Frame by Frame*

The application of human-computer interaction needs the requirement of recognizing an action in frame-by-frame model. In this experiment, we verify this function and show our approaches do not need to wait for all the frames of the video to recognize an action. In this experiment, we also query the tree constructed using 8 actions and 5 people in AS3. Fig. 7 shows the performance of recognizing action Pickup & throw. The average frames of the action are about 47. Note that we need only half of the overall frames to obtain an accuracy of over 90%. Our recognition performance will be stable after frame 22. In our system, each feature query tasks about only 5ms. Our Position-tree is approximately 4 times faster than Feature-tree [2].

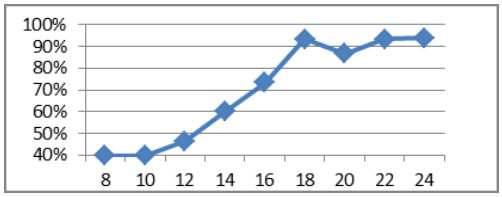


Fig. 7 Recognition the action Pickup & throw in frame-by-frame model.

G. *Effect of Number of Features and Nearest Neighbors*

In this experiment, we explore the effect of number of features and the nearest neighbors. We verify our method on the AS3CrSUB. Fig. 9 shows that more features really help to improve the performance. The best accuracy of 450 features is 96%, improving 5% than 50 features. However, the improvement of 450, 350, 250 features is not obvious. It is very impressive that using 50 features can get about average accuracy of 90%. Note that with the increasing number of nearest neighbors, the performance of 50 features will go down. The reason is that more and more irrelevant neighbors would be found with the increasing number of nearest neighbors. Larger irrelevant neighbors affect the performance of recognition.

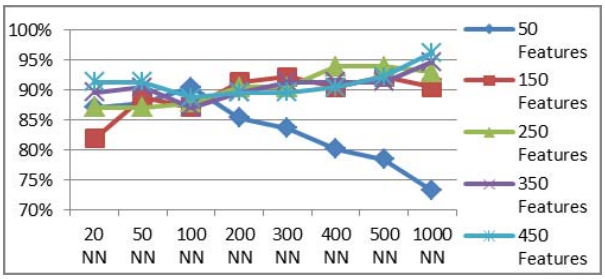


Fig. 9 The performance analysis of the position-tree in different numbers of features and neighbors.

VI. Conclusion

This paper proposes a novel method based on Position-Tree for action recognition. Our method makes full use of the position of interest point and its local features to improve the accuracy of recognition. The advantages of our method are able to fuse multiple features, recognize multiple actions happening simultaneously, incrementally learn actions and recognize action in frame by frame model. The approach is very practical and also obtains good performance of recognizing actions.

Acknowledgment

The study has been financially supported by: Shenzhen Technology Project (JSGG20130624154940238), Shenzhen Technology Project (JCYJ20130402113127502), Guangdong-CAS Strategic Cooperation Program (2012B090400044), Shenzhen Technology Project (ZD201111040087A), CAS and Locality Cooperation Projects (ZNGZ-2011-012), Guangdong Innovative Research Team Program (No. 201001D0104648280)

References

[1] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Computer Vision and Pattern Recognition (CVPR),* 2011, pp. 1297-1304.

[2] K. K. Reddy, J. Liu, and M. Shah, "Incremental action recognition using feature-tree," in *Computer Vision, IEEE International Conference on*, 2009, pp. 1010-1017.

[3] P. Hong, M. Turk, and T. S. Huang, "Gesture modeling and recognition using finite state machines," in *Automatic Face and Gesture Recognition, IEEE International Conference on*, 2000, pp. 410-415.

[4] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden Markov models for complex action recognition," in *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1997, pp. 994-999.

[5] A. D. Wilson and A. F. Bobick, "Parametric hidden markov models for gesture recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 21, pp. 884-900, 1999.

[6] S. Carlsson and J. Sullivan, "Action recognition by shape matching to key frames," in *Workshop on Models versus Exemplars in Computer Vision*, 2001.

[7] K. Cheung, S. Baker, and T. Kanade, "Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture," in *Computer Vision and Pattern Recognition*, 2003, pp. I-77-I-84 vol. 1.

[8] I. Laptev, "On space-time interest points," *International Journal of Computer Vision,* vol. 64, pp. 107-123, 2005.

[9] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of the 15th international conference on Multimedia*, 2007, pp. 357-360.

[10] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," *ECCV 2008,* pp. 650-663.

[11] A. Klaser and M. Marszalek, "A spatio-temporal descriptor based on 3D-gradients," presented at the BMVC, 2008.

[12] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Visual Surveillance and Performance Evaluation of Tracking and Surveillance. 2nd Joint IEEE International Workshop on*, 2005, pp. 65-72.

[13] W. Li, Z. Zhang, and Z. Liu, "Action recognition based on a bag of 3d points," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, 2010, pp. 9-14.

[14] X. Yang and Y. L. Tian, "EigenJoints-based Action Recognition Using Naïve-Bayes-Nearest-Neighbor," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, 2012, pp. 14-19.

[15] L. Xia, C. C. Chen, and J. Aggarwal, "View invariant human action recognition using histograms of 3D joints," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*, 2012, pp. 20-27.

[16] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu, "Robust 3d action recognition with random occupancy patterns," *ECCV,* pp. 872-885, 2012.

[17] X. Yang, C. Zhang, and Y. L. Tian, "Recognizing Actions Using Depth Motion Maps-based Histograms of Oriented Gradients," presented at the ACM Multimedia 2012.

[18] N. Katayama and S. i. Satoh, "The SR-tree: An index structure for high-dimensional nearest neighbor queries," in *ACM SIGMOD Record*, 1997, pp. 369-380.

[19] A. Vieira, E. Nascimento, G. Oliveira, Z. Liu, and M. Campos, "Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences," *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications,* pp. 252-259, 2012.