STOP: Space-Time Occupancy Patterns for 3D Action Recognition from Depth Map Sequences

Antonio W. Vieira[1,2], Erickson R. Nascimento[1], Gabriel L. Oliveira[1], Zicheng Liu[3], and Mario F.M. Campos[1,⋆]

[1] DCC - Universidade Federal de Minas Gerais, Belo Horizonte, Brazil
{awilson,erickson,gabriel,mario}@dcc.ufmg.br
[2] CCET - Unimontes, Montes Claros, Brazil
[3] Microsoft Research, Redmond, USA
zliu@microsoft.com

Abstract. This paper presents Space-Time Occupancy Patterns (STOP), a new visual representation for 3D action recognition from sequences of depth maps. In this new representation, space and time axes are divided into multiple segments to define a 4D grid for each depth map sequence. The advantage of STOP is that it preserves spatial and temporal contextual information between space-time cells while being flexible enough to accommodate intra-action variations. Our visual representation is validated with experiments on a public 3D human action dataset. For the challenging cross-subject test, we significantly improved the recognition accuracy from the previously reported 74.7% to 84.8%. Furthermore, we present an automatic segmentation and time alignment method for online recognition of depth sequences.

Keywords: Pattern recognition, Machine Learning, Human action.

1 Introduction

Human action recognition has been an active research topic for many years. It has a wide range of applications including senior home monitoring, video surveillance, video indexing and search, and human robot interaction, to name a few. So far, most of the work has been focused on using 2D video sequences as input due to the ubiquity of conventional video cameras.

State-of-the-art algorithms for action recognition use silhouettes, Space-Time Interest Point (STIP) and skeletons. Skeletons can be obtained from motion capture systems using body joint markers or directly tracked from depth maps. However, tracking of body joints from depth maps is not a completely-solved problem. For example, the joint positions returned by XBOX Kinect skeleton tracker are quite noisy [9].

A recently published work that uses depth maps for action recognition is described by Li et al. [9]. For the depth map in any given frame, their method uses the silhouettes projected onto the coordinate planes and samples a small set

⋆ This work is supported by grants from CNPq, CAPES and FAPEMIG.

L. Alvarez et al. (Eds.): CIARP 2012, LNCS 7441, pp. 252–259, 2012.

of 3D points, which are the interest points. The dissimilarity between two depth maps is computed by the Hausdorff distance between the two sets of interest points. One limitation of this approach is that the spatial context information between interest points is lost. Furthermore, due to noise and occlusions in the depth maps, the silhouettes viewed from the side and from the top may not be very reliable. This makes it very difficult to robustly sample the interest points given the geometry and motion variations between different persons. This is probably why they reported low recognition accuracy for cross-subject tests.

Our approach represents the depth sequence in a 4D space-time grid and uses a saturation scheme to enhance the roles of the sparse cells which typically consist of points on the silhouettes or moving parts of the body. These cells contain important information for action recognition. We will show that the feature vectors obtained by using this scheme perform much better than the original histogram vectors without saturation. In addition, we use an action graph based system to learn a statistical model for each action class and use a state machine to segment long depth sequences using a neutral pose classifier.

Related Work. Action recognition methods can be classified in global or local methods. The methods in the first category use global features such as silhouettes [10,8] and space-time volume information [15,4]. The methods in the second category use local features for which a set of interest points are extracted from a video and a feature descriptor is computed for each interest point. These locally extracted features are used to characterize actions for recognition [3,13]. Compared to action recognition from 2D data, the amount of work from 3D data has been quite limited due to the difficulty of 3D data acquisition.

One way to obtain 3D data is by using marker-based motion capture systems such as those made by MoCap. These systems capture the 3D positions of the markers which are attached to the joints of a performer. One dataset like this can be downloaded from [1]. Han et al. [6] developed a technique to learn a low-dimensional subspace from the high dimensional space of joint positions, and perform action recognition in the learned low-dimensional space.

The second way to obtain 3D data is to use multiple 2D video streams to reconstruct 3D information. Gu et al. [5] developed a system that generates volumetric data from multiple views as in [14]. Furthermore, they recovered the joint positions which are used as features for action and gait recognition.

The third way to obtain 3D data is to use depth sensors. One type of depth sensor is based on the time-of-flight principle [7]. Breuer et al. [2] proposed to use a 3D articulated hand model as a template to match the 3D point cloud captured by a time-of-flight camera. The other type of depth sensor uses structured light patterns. A number of systems used visible structured light patterns as in [11].

Using visible lights has the drawback that it is invasive. Recently, Microsoft released a depth camera, called Kinect, which uses invisible structured lights. Li et al. [9] developed a technique for action recognition from depth maps captured by a depth camera similar to Kinect. They captured a dataset with various people performing different actions. This dataset is used in our experiments.

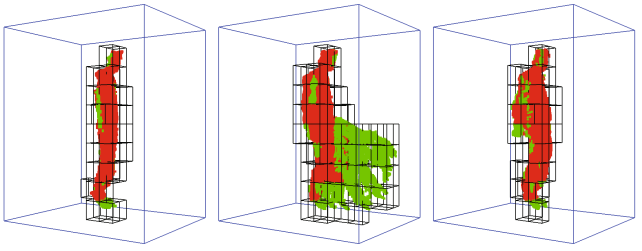

Fig. 1. Space-time cells of a depth sequence of the action *Forward Kick*. For each time segment, we place all the frames together in the same space. The red points are those in the cells with more than p points.

2 Space-Time Occupancy Patterns

To construct our visual representation, we consider a given depth sequence of someone performing an action as a set $A = \{(x_i, y_i, z_i, t_i), i = 1 \dots N\}$ into a space-time box B where the fourth dimension is the frame index t_i which specifies time. This space-time box is then partitioned into a 4-dimensional grid with m $4D$ cells. We use c_i to denote the ith cell. The set of cells is called a *partition*, denoted as $C = \{c_1, ..., c_m\}$. For each cell c_i, we denote by A_i its intersection with the set of 4-dimensional points A, that is, $A_i = A \cap c_i$. The occupancy value of c_i is defined as

$$P(c_i) = \begin{cases} 1, \text{ if } & |A_i| \geq p \\ \frac{|A_i|}{p}, & \text{otherwise} \end{cases}, \tag{1}$$

where p is a predefined saturation parameter, empirically selected to maximize recognition accuracy. Our experiments discuss the saturation parameter effect by presenting the recognition accuracies for different values of p. To construct our feature vector for a given depth sequence A and partition C, we denote $f(A, C) = (P(c_1), P(c_1), \dots, P(c_m))^T$.

$f(A, C)$ is an m-dimensional vector, which is called the Space-Time Occupancy Pattern (STOP) of A with respect to the partition C. In our offline classification, B is partitioned in 10 segments along x, y and z, and 3 segments along time axis, so the dimension of $f(A, C)$ is 3000.

Figure 1 illustrates the space-time cells from a depth sequence of the action Forward Kick. The sequence is divided into three time segments, and each segment is comprised of about 20 frames. Only the non-empty cells are drawn. The red points are those in the cells which contain more than p points.

In general, a STOP feature vector is quite sparse, that is, the majority of its elements are zero. This motivated us to perform a dimensionality reduction using a modified version of Principal Component Analysis (PCA), called Orthogonal Class Learning (OCL), as presented in [12], to obtain, for each STOP feature f_i, a low dimensional feature e_i that we call PCA-STOP. In our experiments, the dimension of a PCA-STOP feature is 300.

2.1 Offline Recognition

For offline recognition, we consider training and testing sets with segmented actions where start and end frame of each action in a depth sequence is well segmented and we use a simple classifier based on the cosine distance.

Let H denote the number of action classes and E_h denote the set with L PCA-STOP feature vectors in the training data of action class h, $h = 1, ..., H$. Given any PCA-STOP feature vector e to be classified, the distance from e to action class h is given by

$$D_h(e) = 1 - \frac{1}{L} \sum_{\hat{e} \in E_h} \frac{< e, \hat{e} >}{\|e\| \|\hat{e}\|}. \tag{2}$$

2.2 Online Recognition

To perform online recognition, we compute a short-time STOP feature vector, from every 5 frames, and obtain the short-time PCA-STOP features. These features are used to train a neutral pose classifier using a Support Vector Machine (SVM) in order to address temporal segmentation. For the back-end classifier, we use an approach called action graph [8] which was proposed by Li et al. for video action recognition.

Compared to the offline classifier, the action graph classifier has the advantage that it can perform classification (decoding) without having to wait until an action is finished. Similar to Hidden Markov Models (HMM), action graph is flexible in handling performing speed variations and takes into account the temporal dependency. Compared to HMM, action graph has the advantage that it requires less training data and allows different actions to share the states.

Figure 2 is an overview of the online system. It maintains two states: Idle state and Action state. Please note that these two states are different from the states in the action graph. For every 5 frames, the system computes the short-time PCA-STOP descriptor, and applies the SVM neutral pose classifier. If the current state is Idle while seeing a neutral pose, it stays in the Idle state. If it sees a non-neutral pose, it transitions to Action state. While it is in Action state, the action graph performs decoding whenever a new short-time PCA-STOP is

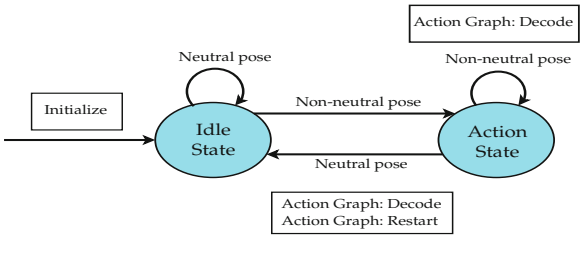

Fig. 2. Overview of the online action recognition system.

added. When the system detects a neutral pose, it transitions back to the neutral state. In the meantime, it restarts the action graph to prepare for the next action.

2.3 Action Graph

Formally, an action graph is a system composed of a set $A = \{a_1, a_2, \ldots, a_H\}$ with H trained action classes, a set with t key poses $V = \{v_1, v_2, \ldots, v_t\}$, a set $\Sigma = \{p(e|v_1), p(e|v_2), \ldots, p(e|v_t)\}$ with observation model of a feature vector e with respect to key poses $v_i \in V$ and a set $\Gamma = \{P_1, P_2, \ldots, P_H\}$ with H matrices, $t \times t$, to model the transition probability between key poses for a given action class.

Given a test depth map sequence, we obtain a sequence of short-time PCA-STOP features $T = \{e_1, e_2, \ldots, e_r\}$, and compute the probability of occurrence of T with respect to each trained action class $h \in A$. The resulting class $\bar{h}$ is given by $\bar{h} = arg\max_h\{p(T|h)\}$.

The decoding process to compute $\bar{h}$ uses a dynamic programing scheme to save computational time. More details can be found in [8].

To learn the key poses, we cluster all training feature vectors from all action types using a K-means. This is different from [9] because we have a single feature vector per frame while [9] uses a set of unorganized points as their visual description. The resulting key poses V are the nodes (also called states) of the action graph. For each cluster $v_j \in V$, we fit a Gaussian distribution and estimate the observation likelihood model $p(e|v_j) \in \Sigma$.

The transition matrix $P_h \in \Gamma$ is computed as $p(j|i) = \frac{N_{(i \to j)}}{N_i}$, where $N_{(i \to j)}$ is the number of transitions from key pose i to j in the training data that belongs to action h, and N_i is the number of times state i is observed in the training data that belongs to action h.

3 Experiments

In this section we present our experimental results. Firstly, our offline classification method is compared with another offline method using a public dataset. Then, experiments with our online classification method shows that we are able to classify unsegmented depth sequences while addressing time alignment.

3.1 Offline Experiments

We used the public MSR Action3D Dataset [9] to validate our offline classification technique. It was recorded by using a depth camera similar to the Kinect device that uses infra-red light, capturing depth maps of 640×480 at about 15 frames per second. There are 20 action types, and 10 subjects. Each subject performs each action 2-3 times. In [9] the 20 action types are divided into three subsets AS1, AS2 and AS3, each having 8 action types. In order to present a comparison, we present our recognition results on the three subsets. As in [9], all depth maps are firstly down-sampled by factor of 2.

For each subset of action types, three tests were performed. In Test I, $\frac{1}{3}$ of the instances were used in the training phase while the rest was used in actual testing. In Test II, $\frac{2}{3}$ of the instances were used for training while the rest was used for testing. Test III is a cross-subject test where instances performed by half of the subjects were used for training and the rest was used as testing. In other words, the subjects in the test data are not seen in the training data. The recognition results obtained by our method and by Li et al. [9] on the three action subjects are shown in Table 1.

Table 1. Comparison of offline recognition accuracies (%)

	Test I			**Test II**			**Test III**		
Set	**Li et al.**	**Skt**	**Our**	**Li et al.**	**Skt**	**Our**	**Li et al.**	**Skt**	**Our**
AS1	89.50	68.00	98.23	93.30	72.97	99.12	72.90	40.28	84.70
AS2	89.00	73.86	94.82	92.90	70.67	96.95	71.90	50.00	81.30
AS3	96.30	78.67	97.35	96.30	83.78	98.67	79.20	73.91	88.40
Avg	91.36	73.51	96.80	94.20	75.81	98.25	74.70	54.73	84.80

In order to show that our PCA-STOP features are more discriminative for action recognition than skeletons obtained from depth maps, we used the recorded skeletons of the same dataset for classification. There are different ways to encode the skeleton joint positions as a feature descriptor for classification. We found that the best recognition accuracy is obtained by using Fast Fourier Transform (FFT) coefficients on the curves of the joint positions over time. In Table 1, column Skt shows the classification results using the skeleton feature. We can readily see that our method outperforms the other two methods in all the test cases. In the more challenging cross-subject test (Test III), we have improved the average recognition accuracy from 74.70% to 84.80%.

To demonstrate the effect of the saturation parameter, Figure 3 shows the cross-subject recognition rates of our PCA-STOP for different saturation parameter values. Notice that the recognition accuracy is quite stable for small values of saturation and that performance decreases as the saturation parameter gets too large.

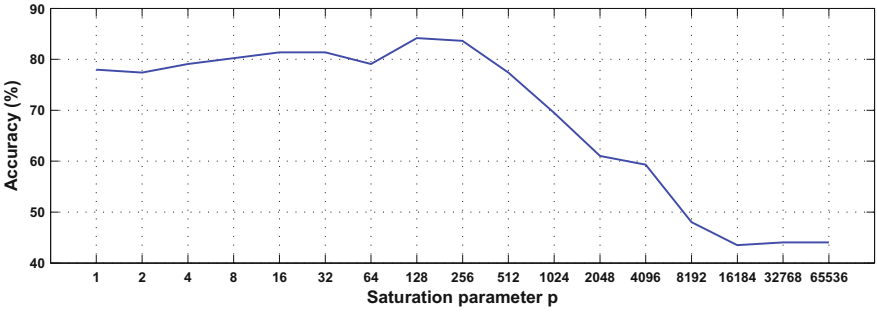

Fig. 3. Recognition accuracies for different values of saturation parameter p

3.2 Online Experiments

For our online experiments, we used long unsegmented depth sequence where several actions are performed over time. Our neutral pose and action graph is used for classification with temporal segmentation and alignment. We used unsegmented sequences with the same set of actions as used in our offline experiments. For this experiment, we used 511 depth sequences from 9 subjects as training set and 56 depth sequences from a different subject as test set. The training set is used to learn the action graph. The sequences in the test set are concatenated together to form a long sequence for testing. We do this 10 times, each with a different partition where the subject in the test set are not seen in the training set. The overall accuracy for this ten-fold cross classification was 98.41% in a test using all 20 action types, which emphasizes that, by addressing time alignment, online classification improves cross-subject recognition.

In order to illustrate segmentation and classification we show, in Figure 4, an example of classification using a long unsegmented sequence. The figure presents a matrix, where columns stands for frame number and rows stands for action classes. The actual action segmentation of each sequence along time is shown by horizontal black bars and, in gray bars, the predicted action segmentation and classification based on our online approach. Notice that a neutral pose is classified before and after the performance of each action. This is coherent with the dataset where each subject rested in a neutral pose between the performance of two different actions.

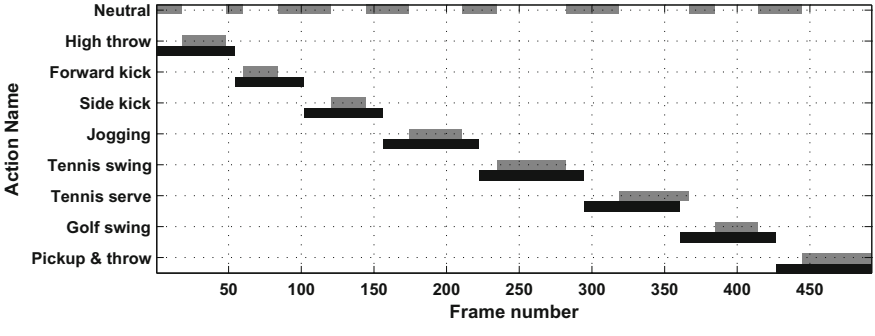

Fig. 4. Example of online action segmentation and recognition in a long depth sequence. Black horizontal bars show actual action segmentation along time and gray bars show the predicted action segmentation and classification.

4 Conclusions

We presented Space-Time Occupancy Patters (STOP), a novel feature descriptor for classifying human action from depth sequences. It leverages the spatial and temporal contextual information while allowing for intra-action variations. It is particularly suited for recognizing short and non-repetitive actions. The accuracy

of our STOP features for action classification has shown to be superior in a comparison with previous work using public dataset. Furthermore, we developed an online action recognition system based on short-time STOP features, which handles automatic segmentation and temporal alignment.

References

1. Carnegie mellon university motion capture database, `http://mocap.cs.cmu.edu`
2. Breuer, P., Eckes, C., Müller, S.: Hand Gesture Recognition with a Novel IR Time-of-Flight Range Camera–A Pilot Study. In: Gagalowicz, A., Philips, W. (eds.) MIRAGE 2007. LNCS, vol. 4418, pp. 247–260. Springer, Heidelberg (2007)
3. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: IEEE Int. Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance, San Diego, CA (2005)
4. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. IEEE Trans. PAMI 29(12) (2007)
5. Gu, J., Ding, X., Wang, S., Wu, Y.: Action and gait recognition from recovered 3D human joints. IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics 40(4) (2010)
6. Han, L., Wu, X., Liang, W., Hou, G., Jia, Y.: Discriminative human action recognition in the learned hierarchical manifold space. Image Vision Comput. 28, 836–849 (2010)
7. Iddan, G.J., Yahav, G.: 3D imaging in the studio. In: Proc. SPIE 4298 (2001)
8. Li, W., Zhang, Z., Liu, Z.: Expandable data-driven graphical modeling of human actions based on salient postures. IEEE Transactions on Circuits and Systems for Video Technology 18(11) (2008)
9. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3D points. In: CVPR Workshop for Human Communicative Behavior Analysis, San Francisco, CA (June 2010)
10. Lv, F., Nevatia, R.: Single view human action recognition using key pose matching and Viterbi path searching. In: Proc. CVPR (2007)
11. Malassiotis, S., Tsalakanidou, F., Mavridis, N., Giagourta, V., Grammalidis, N., Strintzis, M.G.: A face and gesture recognition system based on an active stereo sensor. In: Proc. ICPR, Thessaloniki, Greece, vol. 3 (October 2001)
12. Oliveira, G.L., Nascimento, E.R., Vieira, A.W., Campos, M.F.M.: Sparse spatial coding: A novel approach for efficient and accurate object recognition. In: Proc. ICRA, St. Paul, MN (May 2012)
13. Sun, J., Wu, X., Yan, S., Cheong, L., Chua, T., Li, J.: Hierarchical spatio-temporal context modeling for action recognition. In: Proc. CVPR, Miami, FL (June 2009)
14. Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes 104(2) (2006)
15. Yilmaz, A., Shah, M.: Actions sketch: a novel action representation. In: Proc. CVPR, vol. 1 (2005)