Range-Sample Depth Feature for Action Recognition*

Cewu Lu[§] Jiaya Jia[†] Chi-Keung Tang[§]
[§] The Hong Kong University of Science and Technology
[†] The Chinese University of Hong Kong

Abstract

We propose binary range-sample feature in depth. It is based on τ tests and achieves reasonable invariance with respect to possible change in scale, viewpoint, and background. It is robust to occlusion and data corruption as well. The descriptor works in a high speed thanks to its binary property. Working together with standard learning algorithms, the proposed descriptor achieves state-of-the-art results on benchmark datasets in our experiments. Impressively short running time is also yielded.

1. Introduction

Action understanding is one fundamental problem in computer vision, which enables practical machine intelligence in applications of video surveillance, sport video analysis, and video retrieval. Although there have been many research achievements over the past years, recognition accuracy still leaves much room to improve. The major bottleneck lies in the inherent difficulty in inferring 3D information in videos, where geometric relationship could be vastly valuable for accurate action recognition.

Recent emergence of low-cost depth sensors, such as Microsoft Kinect, stir research up in this field. With affordable and realtime depth capturing devices available, it is possible now to consider geometric invariance (e.g., scale, viewpoint, and background change) in activity recognition for performance enhancement in the basic feature level.

Two basic elements for activity recognition are action feature [2] and learning algorithms [20], both of which were studied broadly in color videos. In general, learning is to get statistical information from features for results in the semantic level. Given the fact that good features are instrumental to the success of learning, we propose a new configuration that taps into the full potential of depth sequences. Our range-sample feature yields reasonable invariance in a number of aspects in depth.

A few features working on depth resemble their RGB counterparts. For example, Wang *et al.* [17] proposed descriptors similar to raw RGB patch representation, i.e., occupancy pattern in 3D sub-volumes. Oreifej *et al.* [13] extended the Histogram of Gradient (HOG) to generate a histogram of normals in 4D. Unlike these methods, we exploit more 3D geometric relationship (e.g., occlusion and layers) inherent in the available depth.

Our work is inspired by how 3D computer (or human) vision effectively performs geometric reasoning in scene depth. For action understanding, our feature is built on local human parts/regions by coarsely eliminating clutter background and complex occlusion. It also robust to relatively small viewpoint change and missing depth information. We call our feature "range-sample" due to the carefully designed sampling scheme to exclude most of problematic pixel relationship and incorporate spatio-temporal human body cubes/patches to simultaneously capture shape and motion in depth sequences.

Our feature is binary with its root on the τ tests [1]. It runs quickly due to fast Boolean operations. It also works decently due to the fact that thousands of pixel pairs working in synergy can sufficiently capture local structure. By extensive comparison to other features, we conclude ours achieves hundreds of times speedup with comparable or higher recognition accuracy in benchmark datasets.

This paper provides several experimental results. We validate the invariance properties of our range-sample feature. When incorporated into standard discriminative action learning framework, this feature yields state-of-the-art results on benchmark datasets. We also report the performance on action localization. The save of computation is significant by using our feature as real-time performance for action localization (using a single CPU core) and action recognition (using multiple CPU cores) is accomplished with our unoptimized MATLAB code.

2. Related Work

We review action recognition techniques, which use depth cameras. Various attempts have been made to extend methods originally developed for color sequences to depth

*This research is supported by the Research Grant Council of the Hong Kong Special Administrative Region under grant numbers 412911 and 619313, and the Google Faculty Award.

ones. For instance, inspired by a bag of words (BoW), in [6], Li *et al.* sampled points from silhouette of a depth image to obtain a bag of 3D points, which are then clustered to enable recognition. Analogous to approaches using temporal modeling to measure human actions [11], a hidden Markov model (HMM) was adopted in [9] to represent the transition probability for pre-defined 3D joint positions. In [3], conditional random field (CRF) was used for action recognition.

Following another way, in [19], a depth sequence was processed globally where the average difference between depth frames is computed to capture the orderless motion information. A single HOG descriptor is extracted from the averaged map. In [18], Jiang *et al.* took human skeleton joints as key points to capture action context considering both the joint location and the area around the joints. In [17], possible sub-volumes were selected from input depth sequences. The most discriminative sub-volumes were selected to perform recognition.

Local descriptors using skeleton joints were explored in [14, 18]. These features do not capture local appearance. Recently, a skeleton joints geometry constrained method [8] based on dictionary learning [7] was proposed. Appearance-based approaches such as the one of [13] encode the descriptor as the histogram of normals in 4D space. Occupancy descriptor [17, 16] divides a spatio-temporal region into a 3D grid where the number of occupied pixels is counted in each cell.

Our feature is different due to its binary property with τ tests and the developed sampling scheme to reject useless and even harmful geometric relationship. It is depth range dependent and preserves several invariance properties that were not adequately addressed before in the depth descriptor level.

For example, foreground human action invariant to background scene is seldom considered in recent RGBD action descriptors. Our work shows human silhouette can be easily affected by clutter background. Handling it would be vital for high accuracy recognition. Some methods are robust against occlusion [6] in a global descriptor aggregation level (e.g., using bag of words) instead of considering it in the descriptor level. We thus regard this as another major point to take care of in our proposed feature.

3. Feature Design

We will first revisit the τ test scheme in color images [1] that forms the foundation of our feature design in depth. Then we detail our feature for local depth patches and for depth sequences. Similar to the configuration in previous work, the videos we process are captured by a fixed position camera while the foreground person can move freely.

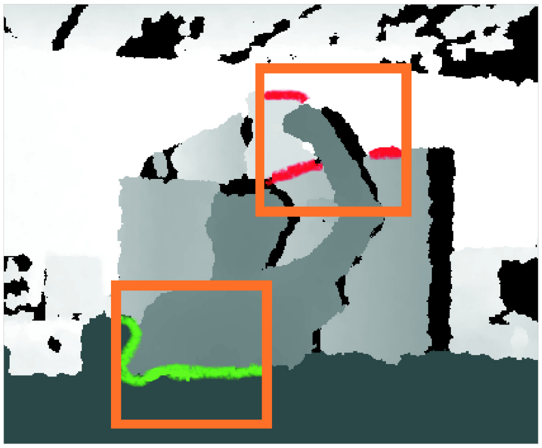

Figure 1. A depth map with many background trouble edges. Red edges belong to background and green edges are due to occlusion.

3.1. Revisiting τ Test

Given a smoothed image patch, the τ test on the given patch is defined as

$$\tau(i,j) = \begin{cases} 1 & p(i) > p(j), \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $p(j)$ is the intensity of the pixel located at j. The final binary descriptor is the concatenation of a set of τ tests of randomly sampled pixel pairs. Given two smoothed image patches, the matching distance is the Hamming distance of their descriptors with the same spatial sampling. The speed and storage efficiency is impressive.

Similar to SIFT and HOG, the success of binary descriptors in recognition lies in its expressive power for characterizing local edge structure. Specifically, a τ test is actually a weak edge indicator between pixels. A large number of τ tests work together can effectively represent complex edge structure in local regions.

3.2. Binary Descriptor in Depth

For a RGB binary descriptor, any two pixels in the input image give rise to a τ test. Thus, structures in the input image are equally likely to be presented by a set of τ tests with (semi) random sampling. There is a notable limitation in this process. That is, if we are only interested to describe foreground human structures and sample pixels in both foreground and background regions, the descriptor performance could be adversely influenced by many *trouble* structure existing on background pixels.

We illustrate this problem using Figure 1. In the orange-frame patches, the red edges are background ones and the green are caused by object conclusion. They do not help at all action recognition and could possibly damage the semantic representation when selected for human action understanding. Also, lacking depth, the original descriptor

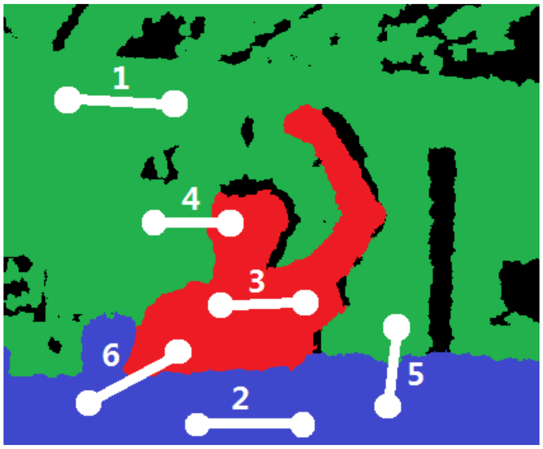

Figure 2. Three layers of a depth map. The green, red, and blue layers are respectively the background, activity, and occlusion ones. The pixel pairs 1-6 are examples of pixel pairs *Back-Both*, *Occ-Both*, *Act-Both*, *Back-Act*, *Back-Occ*, *Occ-Act*. The black pixels are with missing depth.

name	type of pixel pair
Back-Both	both pixels from background
Act-Both	both pixels from activity layer
Occ-Both	both pixels from front layer
Back-Act	pixels from background and activity layers respectively
Back-Occ	pixels from background and occlusion region respectively
Occ-Act	pixels from activity layer and occlusion region respectively

Table 1. Six types of pixel pairs.

[1] working on color videos cannot easily differentiate between human body and background “trouble” edges.

Our depth feature is constructed by locating useful edges. This method has a simple preprocess (in Appendix) to estimate human-activity depth range (activity range for short). We then partition the depth map into three layers, namely background layer for pixels with depth smaller than the activity range, activity layer for pixels within the range, and the occlusion layer for pixels beyond the range as shown in Figure 2.

Depth-Aware Sampling With the three layers we eliminate most “trouble” edges. It is achieved by categorizing pixel pairs into six types as listed in Table 1. Obviously, pixel pairs *Back-Both*, *Occ-Both*, and *Back-Occ* are irrelevant to human action. Pixel pairs *Occ-Act* normally represent occlusion boundary that is also not that relevant. Our τ test only samples pixel pairs *Act-Both* and *Back-Act* where *Back-Act* mostly describes the human body outline, useful in recognition tasks.

3.3. Feature Extraction and Matching in Depth

Each of our feature instance is generated in a depth spatio-temporal local 3D cube centered at (x, y, t), where (x, y) and t are the spatial and temporal coordinates.

Defining Depth Cubes The cube dimension is $d_x \times d_y \times d_t$. d_t refers to the number of frames taking the center of point (x, y, t) in the 3D cube. We denote this value as T. We normally set d_x and d_y as those values corresponding to a real-world 0.6m × 0.6m square to roughly capture part of human body when using Kinect where camera configuration is fixed. Smaller or larger d_x and d_y can be used; they do not affect results too much.

In our configuration, the actual image-plane patch size, i.e., d_x or d_y, with regard to center point (x, y, t) is $\alpha/H(x, y, t)$, where $H(x, y, t)$ is the depth at (x, y, t) and α is the number of pixels along the vertical (or horizontal) line of length 0.6m at 1 meter depth.

Modified τ Tests Within each cube (or each patch inside the involved frames), we perform modified τ tests. As tests require pixel-pair sampling, we first follow the original procedure to define the sampling order so that the same number of pixel pairs can be extracted from each cube as descriptors to match.

Also our pixel pair sampling is after slight Gaussian smoothing. Only pixel pairs in cases of *Act-Both* and *Back-Act* are used. Our test is different from that of [1] also by using two bits for information recording, expressed as

$$\tau_1(i, j) = \begin{cases} 1 & p(i) > p(j), \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

$$\tau_2(i, j) = \begin{cases} 1 & |p(i) - p(j)| < c, \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$p(j)$ here is the depth of pixel j and $c = 20$cm in real world. The two bits τ_1 and τ_2 respectively encode the sign and amplitude of depth difference between pixels i and j that are one pair sampled from the corresponding cube/frame as mentioned above.

In each frame, we sample 512 pixel pairs in each cube, half of which follow random sampling. The other half is with pixel spatial distance following a Gaussian distribution. The uniform and Gaussian distributions are to roughly capture loose and tight local structure in each patch. Our method excludes pixels without depth from sampling. Our final *range-sample* feature is the concatenation of binary features, taking $T \times 512 \times 2$ bits in total in each cube.

Range-Sample Feature Matching Given two *range-sample* descriptors for different cubes in depth, we compute the Hamming distance between a few rotated versions of them to measure similarity, This process makes matching more robust to possible rotation. Each feature has five

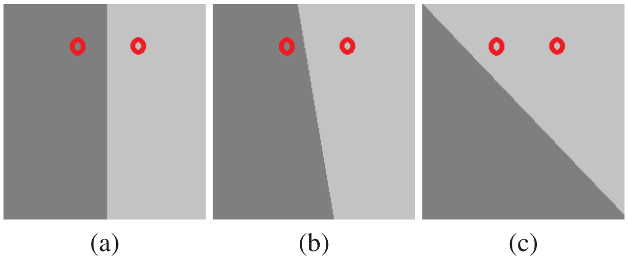

Figure 3. A depth patch and its rotated versions. Red points are used in a τ test.

rotated versions in the $x - y$ plane with rotation angles $-30°$, $-15°$, 0, $15°$,and $30°$.

During *range-sample* feature matching, we report the smallest Hamming distance in the 5×5 combinations.

4. Analysis

Despite its simplicity, our binary descriptor exhibits reasonable robustness due to the tests and adaptive pixel-pair sampling.

Scale Invariance (SI) Thanks to the available depth from Kinect and other depth capturing devices, the spatial scale of all descriptors is kept at $0.6\text{m} \times 0.6\text{m}$ as described above.

Background Invariance (BI) Pairs with both pixels in the background layer are out of the activity range and thus are not sampled. This scheme does not guarantee to remove 100% background pixels. But in our experiments, most problematic depth pixels are excluded in descriptor construction, sufficient to resist background variation.

Occlusion Invariance (OI) Occlusion invariance is partially achieved by our descriptor. When we sample pixel pairs, objects very close to the camera and out of the activity range are rejected. Also, we do not sample pixels straddling the activity and occlusion layers, since the resulting occlusion boundary is not usable for recognition. Although our feature does not handle occlusion occurring within the activity range, it would not largely affect the performance. It is because, in most cases, such occluders are objects interacting with human body. They can be elements that help characterize human activity. For example, a guitar is part of the action "playing guitar". As an important recognition cue, it should not be removed.

Small Rotation and Viewpoint Change Invariance (RI) We require our descriptor to differentiate among actions with significant difference due to rotation while exhibiting good invariance over a reasonably small range. These requirements can be met. Referring to Figures 3, a small rotation in (b) does not affect our modified τ test (in red circles) much in depth. When the rotation is large, as shown in (c), the τ test is able to notice the change accordingly. Another example is shown in Figure 4.

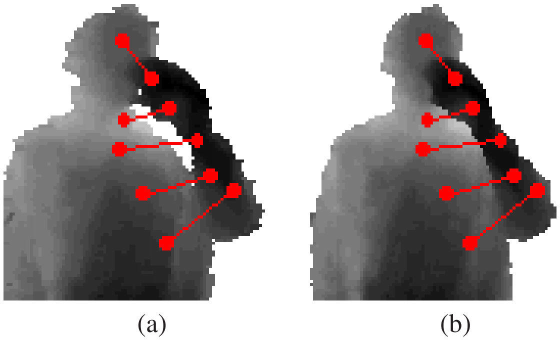

Figure 4. Two viewpoints with 7° difference. The red pixel pairs are for τ tests. The relationship of these pairs remains unchanged.

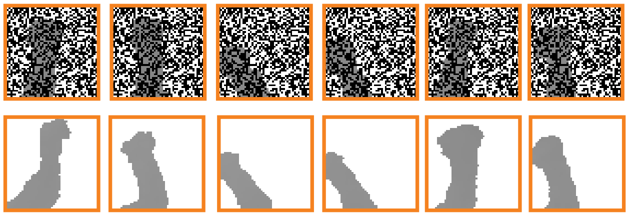

Figure 5. Matching with highly corrupted depth. Top: key frames of a cube with 46% pixel corruption of the action type "cheer up". Bottom: corresponding matching cube in another video.

Corruption Invariance (CI) No missing-depth pixel is sampled for τ tests. In contrast to the gradient- and normal-based descriptors that could be vulnerable to corrupted pixels (missing information), our descriptor is robust with adaptive range sampling. It still works when up to 45% of the pixels are corrupted. Figure 5 shows an example where two cubes representing the same action can be successfully matched even with 46.7% of the pixels corrupted.

5. Experiments

We evaluate our feature invariance property. Combined with standard learning procedure, our descriptor is applied to action recognition and localization. Quantitative comparison with state-of-the-art methods is presented and discussed.

5.1. Performance on Invariance Properties

We first test how possible identical cubes, after variation, can be matched using our features purely based on the low-level information given a large pool of cubes. We name this test *feature retrieval* analogous to high-level image retrieval. We conduct two experiments for evaluation.

Based on Kinect output, we collect human activity depth sequences with corresponding skeletons available. Then, we manually find 200 pairs of cubes that can be matched

Properties	SI	BI	OI	CI	RI
OD	31	23	24	19	18
HON4D	17	29	32	31	15
ours	**1.7**	**2.7**	**3.3**	**1.4**	**1.3**

Table 2. Average matching ranks on individual invariance properties. "OD" denotes the occupancy descriptor of [18]. "HON4D" is the descriptor using 4D histogram of normals [13].

s	2	3	4	5
OD [18]	42.6	97.2	273	429
HON4D [13]	51.2	104	201	351
our descriptor	**5.9**	**19.2**	**29.7**	**59.9**

Table 3. Average matching ranks on s-invariance.

for each invariance property using the following scheme. In total, there are 1,000 matchable cube pairs.

For SI, the matchable cube pairs are obtained at different depth locations; for BI the matchable pairs are from different backgrounds ($5-30\%$ background coverage); for OI the pairs are behind different occluders ($5-30\%$ occlusion coverage); for RI the matchable pairs within a rotation range are sampled; for CI, we generate matchable cubes with different level of corruption by random data removal ($5-30\%$ missing data). The same sampling procedure is applied to constructing features on all these cubes.

To make descriptor retrieval challenging, we also set up a cube pool with 10,000 cubes randomly selected from depth sequences.

In the first experiment, we pick one pair of matchable cubes from the 1,000 set each time, and put one cube into the pool and the other for feature retrieval. Our evaluation is to sort the matching scores between each cube for retrieval and all cubes in the pool and report the rank of ideally matchable cubes. If the rank is 1, obviously the result is perfect. For each invariance property, we use the average rank to measure the performance. The results are tabulated in Table 2, manifesting the quality and usefulness of our low-level range-sample feature.

The second experiment is more challenging. It is to test descriptor's retrieval performance on s-invariance where $s \in [2,5]$. That is, the matchable two cubes in the aforedefined set undergo more than one type of variation in scale, rotation, background, occlusion, and corruption. The first experiment can thus be regarded as 1-invariance matching. We use the average matching rank again to measure feature performance. The results are listed in Table 3.

5.2. Action Recognition

We apply our method to action recognition on 3D action datasets, namely, MSR Actions 3D [6] and MSR Daily Activity 3D [18].

Method	Accuracy $\%$
HON4D [13]	80.00
Dynamic Temporal Warping [10]	54.00
Jiang *et al.* [18]	85.75
Luo *et al.* [8]	95.00
Ours	**95.63**

Table 4. Recognition accuracy on the *Daily Activity 3D* dataset.

Learning Framework Our action recognition framework follows the standard one [4]. It includes a few common steps, such as codebook learning and max pooling. We first select a set of discriminative spatio-temporal 3D cubes, i.e, range-sample features, to build a codebook. According to [4], codebook learning adopts the routine clustering paradigm: we sample a set of 10K+ cubes in the training data, perform hierarchical clustering on range-sample features to find representative clusters, and rank these clusters based on their entropy on the distribution of membership in different action classes. Here, we use the Hamming distance of two features as the clustering distance in hierarchical clustering.

We select 800 features in the top 100 entropy clusters as the atoms of our codebook. Given a testing video, we sample 5000 cubes with range-sample features. For each feature, we report the best matching distance to all the codebook atoms as a 800D vector. Max pooling is then performed on these 5000 vectors to form a video feather vector. Finally, SVM classification is conducted. The distance metric for matching is again the Hamming distance of two descriptors. This scheme is known as "max pooling + SVM" in most classification work.

Daily Activity 3D Dataset This dataset captures 16 types of daily activity using the Kinect sensor. The total number of the activity samples is 320. They cover most human daily activities in the living room: drink, eat, read book, use cellphone, write on paper, use laptop, vacuum floor, cheer up, sit still, toss crumbled paper, play game, lie on sofa, walk, play guitar, stand up, and sit down. For most action types the subjects were asked to perform an activity in two different poses: sitting on sofa and while standing. When a subject is standing close to the sofa or sitting on the sofa, the 3D joint positions are very noisy. Many activities involve human-object interaction, and are thus challenging to recognize. We use our descriptor in learning on this dataset and compare the accuracy with other state-of-the-art methods in Table 4.

MSR Action 3D Dataset The MSR Action 3D dataset [6] contains 20 action types: high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw cross, draw tick, draw circle, hand clap, two hand

Method	Accuracy %
HON4D + D_{dist} [13]	88.89
HON4D [13]	85.85
Jiang *et al.* [18]	88.20
Jiang *et al.* [17]	86.50
Yang *et al.* [19]	85.52
Dollar [2] + BoW	72.40
STIP [16] + BoW	69.57
Vieira *et al.* [16]	78.20
Klaser *et al.* [5]	81.43
OhnBar *et al.* [12]	94.84
Luo *et al.* [8]	**96.70**
Ours	95.62

Table 5. Recognition accuracy on the *MSR Action 3D* dataset.

wave, sideboxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up and throw. Each action was performed by ten different subjects and was repeated for three times. This dataset contains many activities with similar appearance. Performance of our method and others is listed in Table 5.

5.3. Action Localization

We evaluate the performance of our descriptor on action localization. The goal is to locate when a given action occurs in the video . We create a few *Activity-Location-3D* data on top of the well-designed *Daily Activity 3D* dataset, by manually picking frames in *Daily Activity 3D* that contain given action types. We do *not* spatially label action regions for this task. With the selected frames, we take 20% of them as the training data and the remaining 80% as testing data.

For each action type, we manually determine two evidence action parts (see an example in Figure 6). For each action part, we manually select 5 cubes to represent it from the training data. Here we define the distance between any cube C to an evidence action part $\mathcal{P}$ as the Hamming distance between C and its nearest neighbor among the 5 representation cubes of $\mathcal{P}$.

In the localization phase, we search for in each frame two patches with their spanned cubes having the smallest distances to the two evidence action parts, and sum the two distances as the localization score. If the localization score of a frame is smaller than a threshold (3000 in our experiments), we label it as an action active frame. We compare our detected action active frames with the human selected ones and obtain precision, recall and F-measure (the harmonic mean of precision and recall) values. Table 6 compares the performance of our feature with others. It shows our feature is effective to encode important action information.

Descriptor	precision	recall	F-measure
occupancy [18]	73.1	77.5	75.2
HON4D [13]	71.4	78.3	74.7
Ours	**79.4**	**85.2**	**82.2**

Table 6. Precision, recall and F-measure (in %) comparison on the *Activity 3D Location* dataset for action localization.

Problem	time performance
action localization (single core)	42 FPS
action recognition (16 cores)	19 FPS

Table 7. Average FPS. Action localization is tested on a 3.4GHz CPU computer. Action recognition is tested on computers with 2.66GHz CPU.

5.4. Efficiency Discussion

Thanks to the binary form, our descriptor is 100+ times faster in descriptor construction and matching than occupancy descriptor [18], 40+ times faster than HON4D [13]. To analyze running time, we perform the following operations. Given the range-sample feature of a cube, we search for its nearest neighbor cubes in the whole video. Our feature for one cube-to-video matching achieves $100-200$ frames per second (FPS).

Both action localization and recognition involve multiple cube-to-video matching. In our method, triangle inequality property of the Hamming distance is made use of to reduce the amount of computation.

Denote by $\mathcal{C}$ a cube in the testing video, which has been compared with features $\mathcal{T}_1, \dots, \mathcal{T}_{u-1}$ of the learned cubes. During comparing $\mathcal{C}$ with $\mathcal{T}_u$, the Hamming distance of $\mathcal{C}$ and $\mathcal{T}_u$ is

$$d(\mathcal{T}_u, \mathcal{C}) \geq \max_{i=1,\dots,u-1} \{d(\mathcal{T}_u, \mathcal{T}_i) - d(\mathcal{T}_i, \mathcal{C})\} \qquad (4)$$

where $d(\mathcal{T}_u, \mathcal{T}_i)$ is pre-computed and $d(\mathcal{T}_i, \mathcal{C})$ is the distance we calculate in previous rounds. Therefore, we can quickly estimate a lower bound of $d(\mathcal{T}_u, \mathcal{C})$ and reject cubes whose lower bounds are large.

With this strategy, we reach real-time performance for action localization using only one CPU core. Because our descriptor can be used in parallel, real-time performance for action recognition is yielded on a 16-core computer. We report the average FPS in Table 7.

Note that our system is an unoptimized MATLAB implementation. There is still much room for improvement when re-implementing it in C language, using GPU or advanced data structures.

6. Conclusion

We have presented a robust binary action descriptor for depth sequences. Unlike existing methods, we utilize depth

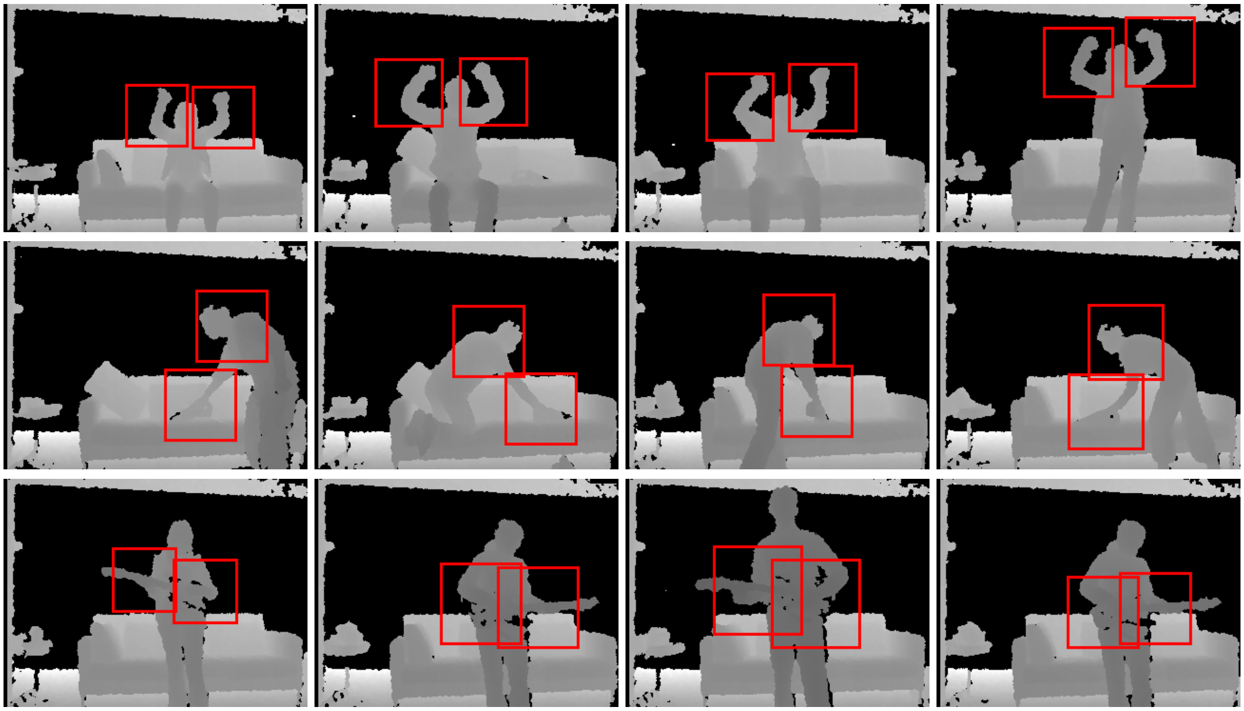
(a) (b) (c) (d)

Figure 6. Two evidence action parts selected for “cheer up” (first row), “vacuum floor” (second row), and “play guitar” (third row).

to delineate occluder, action, and background layers at the early sampling stage, thus achieving good geometric invariance considering various action durations. Scale, background, occlusion, viewpoint and rotation change is taken into consideration and is properly handled. The new descriptor also runs very fast, thanks to the binary nature and τ tests for action characterization.

When combined with standard learning, our feature yields decent recognition results with a large speedup factor. Our MATLAB code is not well optimized, while still reporting impressive frame rates. It is potential to be parallelized in C/C++ on multi-core computers. Part of our future work is to improve the activity range estimation. Using pose information may be useful.

Appendix: Activity Range Estimation

This appendix describes activity range estimation. We use the extracted skeleton points from Kinect to estimate the activity range. It is known that skeleton points may not be that reliable when individual joint locations are problematic. We thus only use their statistical information.

The pipeline is illustrated in Figure 7. We fist extract human body skeleton output from Kinect. For each skeleton in training, we normalize it by setting the mean depth of the joint points to zero. This procedure offsets human location. Consequently, there are two sets of joint points: one in front and the other behind the mean depth point.

Scale Normalization Given the variety of actions types, the human body regions can have different scales, which need to be normalized. The region we consider is a tight rectangular bounding box of the skeleton points with 15% extra padding, as shown in Figure 7(a). We normalize all human regions to 300×100. The joint points are also proportionally mapped into their corresponding positions, as shown in Figure 7(b).

Point Cloud Boundary Our goal is to generate front, activity, and back layers. Seeds for generating the two bounding planes to separate them are required. Joint points with depth less than zero can be naturally regarded as the front seed points (see red points in Figure 7(b)). For the back seeds, we take into account rough thickness of human body and produce them by moving backward positive-depth joint points 30cm further (see blue points in Figure 7(b)). This is a very coarse operation, but is already sufficient in our feature construction.

As the training data contain many frames, we collect all front and back seeds from them and put them to two dense point clouds in one frame. These clouds form the coarse front and back boundary surfaces (see Figure 7(c)). We denote them respectively as $\mathcal{C}_{front}$ and $\mathcal{C}_{back}$.

Max- and Min-Pooling To capture the space reachable by the human body, the boundary surface of pixel p are respectively $\mathcal{S}_{front}(p) = \text{mins}_{q\in\Omega(p)}\{\mathcal{C}_{front}(q)\}$ and

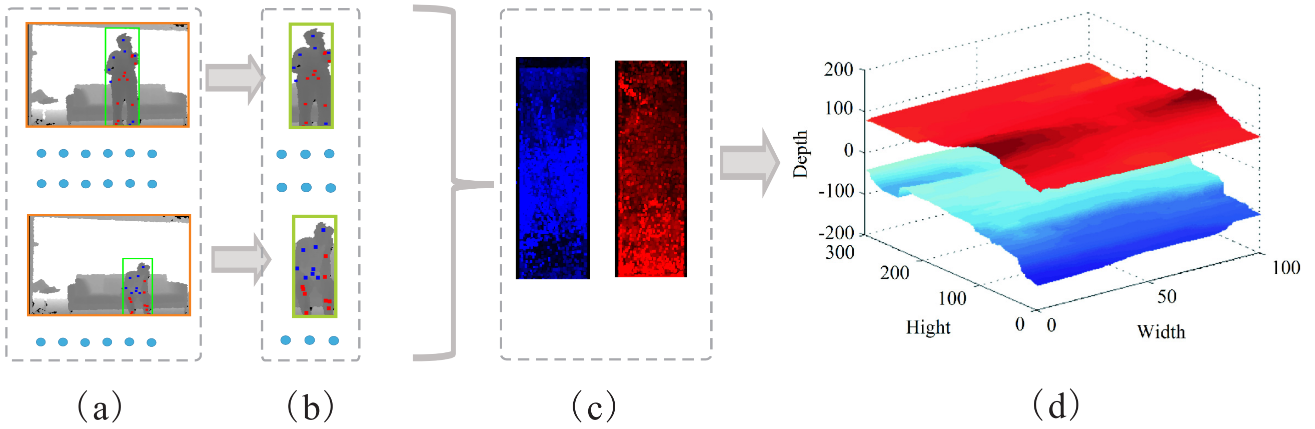

Figure 7. Pipeline of our human activity range estimation: (a) shows the depth frames, (b) is the human region after scale normalization, (c) shows point clouds of the front and black surfaces, (d) our final front and back surfaces.

$\mathcal{S}_{back}(p) = \text{maxs}_{q\in\Omega(p)}\{\mathcal{C}_{back}(q)\}$. where $\Omega(p)$ is a patch centered at pixel p. The patch size we use is 21. Here, mins and maxs are respectively the 5^{th} and 95^{th} percentile, which are the “soft min” and “soft max” to make our processing robust against outliers.

Boundary Surfaces in Testing In the testing phase, we resize the template boundary surface from scale 300×100 to the scale in the testing frame. Suppose the testing skeleton is with mean depth h. We set human activity range as $\mathcal{S}^o_{front} = \psi[\mathcal{S}_{front}(p)] + h$ and $\mathcal{S}_{front}(p)^o = \psi[\mathcal{S}_{front}(p)] + h$, where $\psi[\cdot]$ is a resizing operator to change the scale.

We also consider depth error $\xi(z) = 2.73z^2 + 0.74z - 0.58$ [mm] (z is depth with unit meter according to [15]). The final front and back boundary surfaces are respectively expressed as $\mathcal{S}^{'}_{front}(p) = \mathcal{S}^o_{front} - \xi(\mathcal{S}^o_{front})$ and $\mathcal{S}^{'}_{back}(p) = \mathcal{S}^o_{back} + \xi(\mathcal{S}^o_{back})$.

References

[1] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *ECCV*, 2010.

[2] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VSPETS*, 2005.

[3] L. Han, X. Wu, W. Liang, G. Hou, and Y. Jia. Discriminative human action recognition in the learned hierarchical manifold space. *IVC*, 2010.

[4] A. Jain, A. Gupta, M. Rodriguez, and L. Davis. Representing videos using mid-level discriminative patches. In *CVPR*, 2013.

[5] A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.

[6] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *CVPRW*, 2010.

[7] C. Lu, J. Shi, and J. Jia. Scale adaptive dictionary learning. *IEEE Transactions on Image Processing (TIP)*, 2013.

[8] J. Luo, W. Wang, and H. Qi. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In *ICCV*, 2013.

[9] F. Lv and R. Nevatia. Recognition and segmentation of 3-d human action using hmm and multi-class adaboost. In *ECCV*, 2006.

[10] M. Muller and T. Roder. Motion templates for automatic classification and retrieval of motion capture data. In *ACM SIGGRAPH*, 2006.

[11] H. Ning, W. Xu, Y. Gong, and T. Huang. Latent pose estimator for continuous action recognition. In *ECCV*, 2008.

[12] E. Ohn-Bar and M. Trivedi. Joint angles similiarities and hog2 for action recognition. In *CVPRW*, 2013.

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

[14] 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 *CVPR*, 2011.

[15] J. Smisek, M. Jancosek, and T. Pajdla. 3d with kinect. In *Consumer Depth Cameras for Computer Vision*, 2013.

[16] A. Vieira, E. Nascimento, G. Oliveira, Z. Liu, and M. Campos. Stop: Space-time occupancy patterns for 3d action recognition from depth map sequences. In *Progress in PRIACV&A*, 2012.

[17] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3d action recognition with random occupancy patterns. In *ECCV*, 2012.

[18] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, 2012.

[19] X. Yang, C. Zhang, and Y. Tian. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *ACM international conference on Multimedia*, 2012.

[20] F. Zhou and F. De la Torre. Deformable graph matching. In *CVPR*, 2013.