Open access • Proceedings Article • DOI:10.1109/CVPR.2013.365

Spatio-temporal Depth Cuboid Similarity Feature for Activity Recognition Using Depth Camera — Source link

Lu Xia, Jake K. Aggarwal

Institutions: University of Texas at Austin

Published on: 23 Jun 2013 - Computer Vision and Pattern Recognition

Topics: Feature (machine learning), Feature extraction, Cuboid and Activity recognition

Spatio-Temporal Depth Cuboid Similarity Feature for Activity Recognition Using Depth Camera

Lu Xia and J.K. Aggarwal
Computer & Vision Research Center/Department of ECE
The University of Texas at Austin
xialu@utexas.edu, aggarwaljk@mail.utexas.edu

Abstract

Local spatio-temporal interest points (STIPs) and the resulting features from RGB videos have been proven successful at activity recognition that can handle cluttered backgrounds and partial occlusions. In this paper, we propose its counterpart in depth video and show its efficacy on activity recognition. We present a filtering method to extract STIPs from depth videos (called DSTIP) that effectively suppress the noisy measurements. Further, we build a novel depth cuboid similarity feature (DCSF) to describe the local 3D depth cuboid around the DSTIPs with an adaptable supporting size. We test this feature on activity recognition application using the public MSRAction3D, MSRDailyActivity3D datasets and our own dataset. Experimental evaluation shows that the proposed approach outperforms state-of-the-art activity recognition algorithms on depth videos, and the framework is more widely applicable than existing approaches. We also give detailed comparisons with other features and analysis of choice of parameters as a guidance for applications.

1. Introduction

Activity recognition has been an active field of research in the past several decades [1, 12, 2]. It has connections to many fields of study such as medicine, human-computer interaction, and sociology. The activities that have been studied include single person activities, interactions of several people, and group activities. In the past, research has mainly focused on learning and recognizing actions from image sequences taken by visible light cameras [15, 11]. There are inherent limitations of this type of data source, e.g. it is sensitive to color and illumination changes, occlusions, and background clutters. Despite significant effort, recognizing actions accurately still remains a challenging task.

With the recent advent of the cost-effective Kinect, depth cameras have received a great deal of attention from researchers. It excited interest within the vision and robotics community for its broad applications [6]. The depth sensor has several advantages over visible light camera. First, the range sensor provides 3D structural information of the scene, which offers more discerning information to recover postures and recognize actions. The common low-level difficulties in RGB imagery are significantly alleviated. Second, the depth camera can work in total darkness. This is a benefit for applications such as patient/animal monitoring systems which run 24/7. These advantages lead to interesting research such as estimating human skeletons from a single depth image [21]. The skeletons estimated from depth images are quite accurate under experimental settings and bring benefits to many applications including activity recognition, but the algorithm is limited at the same time. It can hardly work when the human body is partly in view, and the estimation is not reliable or can fail when the person touches the background or when the person is not in an upright position (e.g. patient lying on bed). In surveillance, the camera is usually mounted on an elevated location and the subjects is not facing the camera; these issues will cause difficulties for skeletal estimation.

In this paper, we present algorithm for extracting STIPs from depth videos (DSTIPs) and describing local 3D depth cuboid using the Depth Cuboid Similarity Feature (DCSF). The DSTIP and DCSF can be effectively used to recognize activities without the dependence on skeleton tracking, thus they offer greater flexibility. Our contribution can be summarized as follows: first, we present an algorithm to extract DSTIPs which deals with the noise in depth videos. Second, we present DCSF as a descriptor for the 3D local cuboid in depth videos. Third, we show that this DSTIP+DCSF pipeline may be applied to recognize activity from depth videos, with no dependence on the skeletal joints information, motion segmentation, tracking, or denoising procedures. Moreover, its flexibility and recognition accuracy outperforms other state-of-the-art methods.

2. Related Work

2.1. Interest points extraction and description

Interest points provide a compact representation of image content by describing local parts of the scene thus offer robustness to clutter, occlusions, and intra-class variations. Interest points from 2D images can be used for image matching and retrieval, instance recognition, scene classification, and so on. Its extension into 3D is STIP which is usually used for activity or event recognition. The widely used STIP detectors include the Harris3D detector [11], cuboid detector [5], and Hessian detector [28]. The popular STIP descriptors include Cuboid descriptor [5], HOGHOF [12], HOG3D [10], and extended SURF [28].

Although depth data has existed for several decades, the interest point detection and description has stayed at the level of describing static scenes or objects for a long time [24, 16, 23]. Existing descriptors that describe the local geometry around given points are aimed for object recognition, pose estimation, or 3D registration, such as NARF [22], VFH [19], FPFH [18], and C^3-HLAC [9].

Until recently, a few spatial-temporal cuboid descriptors for depth videos were proposed. Cheng et al. [4] build a Comparative Coding Descriptor(CCD) to describe the $3 \times 3 \times 3$ depth cuboid by comparing the depth value of the center point with the nearby 26 points. Zhao et al. [32] build Local Depth Pattern(LDP) by computing the difference of the average depth values between the cells. In this paper, we propose DCSF as the descriptor for the spatio-temporal depth cuboid that describes the local "appearances" in the depth video based on self-similarity concept. Local self-similarity captures internal geometric layouts of local patches of videos without putting strict restrictions of sharing the same visual patterns. The self-similarity based feature has been proven to work well at object detection/retrieval and action detection on RGB data [20] and object recognition tasks on depth data [8, 9].

2.2. Activity recognition from depth videos

Human activity or gesture recognition using depth images may be divided into two categories: algorithms based on low-level features, and algorithms based on high-level features. In the first category, Li et al. [13] sample a bag of 3D points from the depth maps to characterize a set of salient postures and employ an action graph to model the dynamics of the actions. The algorithm was tested on a clean dataset where humans have been segmented out. Several researchers tried the STIP framework on depth videos. Ni et al. [14] use depth information to partition the space into layers, extract STIPs from RGB channels of each layer using a Harris3D detector, and use HOGHOF to describe the neighborhood of STIPs also in the RGB channel. [32, 7, 4] choose Harris3D detector [11] to extract STIPs from RGB or depth channels.Hernández-Vela et al. [7] uses VFHCRH to describe the 2D depth image patch around the STIPs, Zhao et al. [32] tried HOGHOF and LDP for representation. Zhang et al. [31] extract STIPs by calculating a response function from both depth and RGB channels and use the gradients along x, y, t directions as the descriptor. Notice that most existing methods still depend on the detectors and descriptors designed for RGB images.

In the second category, Wang et al. [27] combine joint location features and local occupancy features and employ a Fourier temporal pyramid to represent the temporal dynamics of the actions. Xia et al. [29] take the skeletal joint locations and vote them into 3D spatial bins and build posture words for action recognition. The second category usually gives better recognition rates, because the high-level skeletal information is well trained and greatly alleviates the difficulties. But the application of such algorithms is also limited, because the skeleton information is not always available for real applications.

3. DSTIP Detection

Like much of the work on interest point detection, a response function is computed at each pixel in the 3D spatio-temporal volume. Our response function is calculated by application of separable filters.

3.1. Spatio-Temporal Filtering

First, a 2D Gaussian smoothing filter is applied on to the spatial dimensions:

$$\boldsymbol{D_s}(x, y, t) = \boldsymbol{D}(x, y, t) * g(x, y \mid \sigma) \tag{1}$$

where $*$ denotes convolution, $\boldsymbol{D}$ and $\boldsymbol{D_s}$ denote the original depth volume and that after spatial filtering respectively. $g(x, y; \sigma)$ is a 2D Gaussian kernel:

$$g(x, y \mid \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/(2\sigma^2)} \tag{2}$$

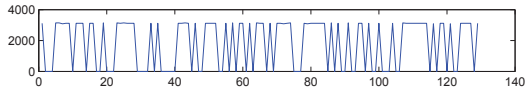
σ controls the spatial scale along x and y. Then we apply a temporal filter along the t dimension:

$$\boldsymbol{D_{st}}(x, y, t) = \boldsymbol{D_s}(x, y, t) * h(t \mid \tau, \omega) \circ \bar{s}(x, y, t \mid \tau) \tag{3}$$

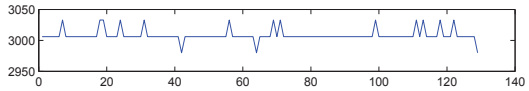
where $\boldsymbol{D_{st}}$ denotes the depth volume after spatio-temporal filtering. $\circ$ denotes element wise matrix multiplication and $h(t \mid \tau, \omega)$ is a 1D complex Gabor filter:

$$h(t \mid \tau, \omega) = e^{-t^2/2\tau^2} \cdot e^{2\pi i \omega t} \tag{4}$$

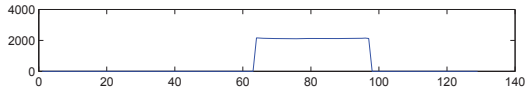
where τ controls the temporal scale of the filter. We use $\omega = 0.6/\tau$. $\bar{s}(x, y, t \mid \tau)$ is a correction function for the noise of the depth sequence at location (x, y, t). τ is the same control parameter as in the Gabor filter. The next section introduces the correction function in detail.



(a) Signal from pixel on object boundary: the value flips from 0 to about 3000 (mm) at a high frequency.



(b) Signal from pixel in the middle of a static object, the value fluctuates around $3006 \pm 27(mm)$.



(c) Signal from pixel where movement happens.

Figure 1: Temporal evolution of pixel values at different locations in the scene.

3.2. Noise Suppression

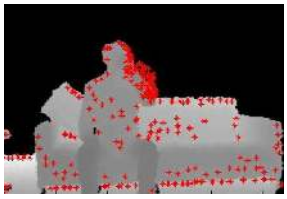
In RGB videos, smoothing functions usually serve to suppress noise. The reason we choose a correction function instead of using filters is based on the different nature of the noise in depth videos. One may divide the noise in depth videos into three categories: The first category of noise comes from the variation of the sensing device, which is evenly distributed throughout the entire image, the magnitude of which is comparatively small. The second category of noise occurs around the boundary of objects, the values jump from the depth of the background to the depth of the foreground, back and forth frequently. The magnitude of the jump can be a few thousand (mm). The third category of noise is the "holes" that appear in the depth images, caused by special reflectance materials, fast movements, porous surfaces, and other random effects. The magnitude of the noise can be a few thousand (mm) as well. Figure 1 gives the temporal evolution of pixel values at different locations in the scene.

The first category is similar to the noise in RGB images, it is usually less distinguishable than real movements. This noise may be reasonably removed using smoothing filters, but in the second and third categories, the magnitude of the noise is usually many times larger than real movements. We can hardly smooth out the noise while leaving the real movement signals unaffected.

The flip of the signal caused by sensor noise usually happens much faster than human movements, and it can happen from once to dozens of times during the whole video. In view of this, we calculate the average duration of the flip of the signal, and use it as a correction function:

$$s(x, y, t_0 \mid \tau) = \frac{\sum_{i=1}^{n_{fp}} \delta t_i(x, y)}{n_{fp}(x, y)} \tag{5}$$

where $n_{fp}(x, y)$ is the total number of flips during the time interval $[t_0 - \tau, t_0 + \tau]$ at location (x, y), and $\delta t_i(x, y)$ is the duration of the i-th flip. We define the number of flips as the number of zero-crossing of the normalized signal $\tilde{d}(t) = d(t) - (d(t)_{max} + d(t)_{min})/2$.



(a) without correction function

(b) with correction function

Figure 2: DSTIPs projected onto x-y dimensions on top of one frame of the video *drink*

This correction function is an indicator of the noise-signal ratio of the pixel at location (x, y, t) during interval $[t_0 - \tau, t_0 + \tau]$. It has a higher value at the pixels where real movement happens thus highlight those movements. We take a threshold so that it only affects the noises and does not discriminate between different movements:

$$\bar{s} = \begin{cases} s_0, & if \;\; s > s_0 \\ s, & else \end{cases} \tag{6}$$

where s_0 is selected to best separate the value $s(x, y, t)$ at the location of noises and location of real motions (e.g.$s_0 = 2$). Figure 2 shows the DSTIPs before and after the correction function. We can see the correction function effectively removes interest points resulting from noise.

3.3. Interest point extraction

Finally, we take the response as:

$$\boldsymbol{R}(x, y, t) = \|\boldsymbol{D}_{st}(x, y, t)\|_2^2 \tag{7}$$

The overall response can be written in a closed form:

$$\boldsymbol{R}(x, y, t) = (\boldsymbol{D} * g * h_{ev} \circ \bar{s})^2 + (\boldsymbol{D} * g * h_{od} \circ \bar{s})^2 \tag{8}$$

$$\begin{aligned} h_{ev}(t \mid \tau, \omega) &= cos(2\pi\omega t) e^{-t^2/2\tau^2} \\ h_{od}(t \mid \tau, \omega) &= sin(2\pi\omega t) e^{-t^2/2\tau^2} \end{aligned} \tag{9}$$

DSTIP is selected at the local maximum of $\boldsymbol{R}$ in spatio-temporal domains and also in scale domain. We take the local maximum with top N_p largest response value as the DSTIPs for each video.

4. Interest Point Description

Here we propose a descriptor for the local 3D cuboid centered at DSTIP. Note it is 3D instead of 4D because the depth image is a function of x and y, not all 3D points $\{x, y, z\}$, but it still provides useful information along the z dimension.

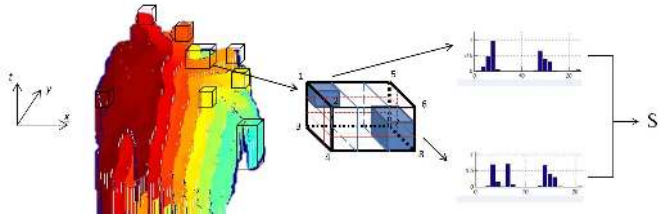


Figure 3: Illustration of extracting DCSF from depth video

4.1. Adaptable supporting size

We extract a 3D cuboid which contains the spatio-temporally windowed pixel values around the DSTIP. Considering objects appear smaller in the image at a farther distance, we design the cuboid size to be adaptable to the depth. We define the spatial size of the cuboid to be proportional to the scale at which it was detected and inversely proportional to the depth at which it locates:

$$\Delta_x^{(i)} = \Delta_y^{(i)} = \sigma \frac{L}{d^{(i)}} \tag{10}$$

where σ is the scale at which the i-th cuboid was detected. And $d^{(i)}$ denotes the depth of the i-th cuboid. Notice that we do not take the depth pixel value at the interest point $\boldsymbol{D}(x_i, y_i, t_i)$ as $d^{(i)}$, because the DSTIP sometimes lands at the edge of body parts. Instead, we compute the minimum non-zero depth value in the 2τ time interval round the location (x_i, y_i, t_i), i.e. $\{\boldsymbol{D}(x_i, y_i, t_i - \tau), \ldots, \boldsymbol{D}(x_i, y_i, t_i + \tau)\}$. This usually gives the depth we want for the cuboid locations. In this way, the size of the cuboid is adjusted according to the real-world size of the object, which corresponds to smaller pixel-size at farther distances and vice-versa. This renders noticeable improvement as compared to a fixed pixel size in our experiments.

The side length of the temporal dimension of a cuboid is simply defined as:

$$\Delta_t^{(i)} = 2\tau \tag{11}$$

4.2. Depth cuboid similarity feature

Different from RGB data, depth data lacks texture, and is inherently noisy. We define a DCSF feature based on the self-similarity to encode the spatio-temporal shape of the 3D cuboid, and we show in Section 6 that this feature is better than other commonly used features.

As shown in Fig. 3, we divide the cuboid into $n_{xy} \times n_{xy} \times n_t$ voxels. (We cut the borders when needed to make sure each voxel contains an integer number of pixels). We define the block as containing $1 \times 1 \times 1$ to $n_{xy} \times n_{xy} \times n_t$ voxels.

We compute a histogram of the depth pixels contained in each block, normalize them to make the total value of every histogram to be 1. Let the histogram calculated from block p and q be h_p and h_q respectively, we use the Bhattacharyya distance to define the similarity:

$$S(p, q) = \Sigma_{n=1}^{M} \sqrt{h_p^{(n)} h_q^{(n)}} \tag{12}$$

which describes the depth relationship of the two blocks. M denotes the number of histogram bins. Note in this definition, the length of the feature depends on n_{xy} and n_t only, it does not relate to the actual size of the cuboid which offers greater freedom for the interest point detection and the cuboid extraction process.

We generate a feature vector by concatenating the similarity scores for all combinations of blocks. Varying spatial-size from 1×1 to $n_{xy} \times n_{xy}$ gives $n_{xy}(n_{xy} - 1)(2n_{xy} - 1)/6$ possibilities, varying temporal-size from 1 to n_t gives $n_t(n_t + 1)/2$ possibilities. In total, the number of blocks N_b generated by varying the number of voxels it contains is at the order of $n_t^2 n_{xy}^3/6$, and the total length of the DCSF feature is $C_{N_b}^2$.

To reduce computational cost, we use integral histograms [17] to compute the depth histograms rapidly. We quantize the depth pixels into M bins, $M = (d_{max} - d_{min})/\Delta d$, where Δd is chosen according to the spatial level of movements to recognize, e.g. $\Delta d = 100mm$. Then we generate M quantized video volumes $\boldsymbol{Q}^{(n)}, n = 1, \ldots, M$, corresponding to the M bins:

$$\boldsymbol{Q}^{(n)}(x, y, t) = \begin{cases} 1, & if(n-1)\Delta d + 1 \leq \boldsymbol{D}(x, y, t) \leq n\Delta d \\ 0, & else \end{cases} \tag{13}$$

We compute an integrated video volume $I^{(n)}, n = 1, \ldots, M$ for each of the quantized video volume $Q^{(n)}$:

$$\begin{aligned} r^{(n)}(x, y, t) &= r^{(n)}(x, y-1, t) + \boldsymbol{Q}^{(n)}(x, y, t) \\ c^{(n)}(x, y, t) &= c^{(n)}(x-1, y, t) + r^{(n)}(x, y, t) \\ \boldsymbol{I}^{(n)}(x, y, t) &= \boldsymbol{I}^{(n)}(x, y, t-1) + c^{(n)}(x, y, t) \end{aligned} \tag{14}$$

where $r^{(n)}(x, y, t)$ denotes the sum of pixels in the rows of $\boldsymbol{Q}^{(n)}(x, y, t)$, $c^{(n)}(x, y, t)$ denotes the sum of pixels in the columns of $r^{(n)}(x, y, t)$, and $\boldsymbol{I}^{(n)}(x, y, t)$ denotes the sum through the temporal dimension of $c^{(n)}(x, y, t)$. The calculation of the histogram of a block at bin n can be obtained using only 7 add operations:

$$\begin{aligned} \boldsymbol{B}^{(n)} = &\{\boldsymbol{I}^{(n)}(p_8) - \boldsymbol{I}^{(n)}(p_7) - \boldsymbol{I}^{(n)}(p_6) + \boldsymbol{I}^{(n)}(p_5)\} \\ &-\{\boldsymbol{I}^{(n)}(p_4) - \boldsymbol{I}^{(n)}(p_3) - \boldsymbol{I}^{(n)}(p_2) + \boldsymbol{I}^{(n)}(p_1)\} \end{aligned} \tag{15}$$

the label of the locations $p_1, \ldots, p_8$ is given in Figure 3. The integral video volume is computed once for each video, and the histogram of each block is computed with $7M$ add operations.

Note the histogram technique renders invariants to small translation and rotations. We intentionally do not rotate the cuboid itself to retain the direction of the movements so that

we can distinguish between actions such as *stand up* and *sit down*. The local feature captures characteristic shapes and motion, thus it provides robust representation of events that is invariant to spatial and temporal shifts, scales, background clutter, partial occlusions, and multiple motions in the scene.

5. Action Description

5.1. Cuboid codebook

Inspired by the successful bag-of-words approach at RGB image classification and retrieval, we build a cuboid codebook by clustering the DCSF using K-means algorithm with Euclidean distance. The spatio-temporal codewords are defined by the center of the clusters and each feature vector can be assigned to a codeword using Euclidean distance or rejected as an outlier. Thus, each depth sequence can be represented as a bag-of-codewords from the codebook. These bag-of-codewords describe what's happening in the depth sequences in a simple yet powerful way. To incorporate the positional information of the cuboid, we concatenate the spatio-temporal information x, y, z, t with the DCSF feature before clustering. This gives small improvements under our experimental settings. Dimension reduction methods such as PCA can be incorporated before clustering without sacrificing the performance when choosing a suitable number of dimensions while making the clustering process much faster. We use a histogram of the cuboid prototypes as the action descriptor and SVM [3] for classification with histogram intersection kernel:

$$\boldsymbol{K}(a,b) = \Sigma_{i=1}^{n} min(a_i, b_i), a_i \geq 0, b_i \geq 0 \qquad (16)$$

5.2. Mining discriminative feature pool

Not all the cuboid prototypes give the same level of discrimination among different actions, some cuboids may be related with movements that do not offer good discrimination among different actions, e.g. the sway of the body. To select the discriminative feature set from the pool, we use F-score. In a binary class case, given training vectors $x_k, k = 1, \ldots, m$, if the number of positive and negative instances are n_+ and n_- respectively, the F-score of the i-th feature $F(i)$ is defined as:

$$\frac{(\bar{x}_i^{(+)} - \bar{x}_i)^2 + (\bar{x}_i^{(-)} - \bar{x}_i)^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n_+}(x_{k,i}^{(+)} - \bar{x}_i^{(+)})^2 + \frac{1}{n_- - 1}\sum_{k=1}^{n_-}(x_{k,i}^{(-)} - \bar{x}_i^{(-)})^2} \qquad (17)$$

where $\bar{x}_i, \bar{x}_i^{(+)}, \bar{x}_i^{(-)}$ are the average of the i-th feature of the whole, positive, and negative data. $x_{k,i}^{(+)}$ is the i-th feature of the k-th positive instance, and $x_{k,i}^{(-)}$ is the i-th feature of the k-th negative instance. The F-score indicates the discrimination between the positive and negative sets. We rank the cuboid prototypes by their F-scores and select features with high F-scores. The threshold is manually selected to cut between low and high F-scores. The number of features to keep generally depends on how good the STIPs are. In our experiments, small improvement is observed by deleting 1-2% cuboid prototypes. We also tested the well-known TF-IDF weighting or stop-words, it turns out it does not give noticeable improvement in our experiments.

Method	Accuracy
Li [13]	74.7%
STOP [25]	84.8%
Eigenjoints [30]	82.3%
Random Occupancy Pattern [26]	86.50%
Wang [27]	88.2%
Ours	**89.3%**

Table 1: Comparison of accuracy on MSRAction3D dataset.

6. Experimental Results

We test our algorithm on two public datasets: MSRAction3D dataset [13] and MSRDailyActivity3D dataset [27], and our own dataset. We compare our algorithm with state-of-the-art methods on activity recognition algorithms from depth videos [13, 27, 30, 25, 26]. Experimental results show that our algorithm gives significantly better recognition accuracy than algorithm based on low-level features and gives even better results than algorithm using high-level joint features. We also give detailed comparisons on other choices of detectors or features and evaluation of parameters on our model. We take support region size $L = 6$ in all experiments.

6.1. MSRAction3D dataset

The MSRAction3D dataset [13] mainly collects gaming actions. The depth image is clean, there are no background objects, and the subjects appear at the same depth to the camera. On this dataset, we take $\sigma = 5, \tau = T/27, T/17$ (T denotes the duration of the action sequence) and $N_p = 160$ for DSTIP extraction, and take the number of voxels for each cuboid to be $n_{xy} = 4, n_t = 2$. We fix the cuboid spatial size $\Delta_x = \Delta_y = 6\sigma$ because all actions take place at the same depth.

Table 1 shows the comparison of our algorithm with state-of -the-art algorithms on the MSRAction3D dataset. All algorithms are tested on the 20 actions, and we select half of the subjects as training and the rest as testing. Our algorithm outperforms the algorithms based on 3D silhouette features [13], skeletal joint features [30, 27] and local occupancy patterns[25, 26].

Method	Accuracy
LOP feature [27]	42.5%
Joint position feature [27]	68.0%
Cuboid descriptor [5]	73.6%
HOG [12]	79.1%
DCSF(Ours)	**83.6%**
LOP+Joint [27]	85.75%
DCSF+Joint(Ours)	**88.2%**

Table 2: Comparison of recognition accuracy on MSRDailyActivity3D dataset.

6.2. MSRDailyActivity3D dataset

The MSRDailyActivity3D dataset collects daily activities in a more realistic setting, there are background objects and persons appear at different distances to the camera. Most action types involve human-object interaction. In our testing, we removed the sequences in which the subject is almost still (This may happen in action type: sit still, read books, write on paper, use laptop and play guitar). Note that Li et al.'s algorithm [13] cannot work without segmenting out the human subjects from the depth image, which is not a trivial work considering the human appears at different depths and interacts with objects. Such dependence on important preprocessing largely limits the application of this algorithm. Here, we compare to Wang et al. [27] and other choices of STIP detectors and features, and we show the evaluation of parameters on this dataset.

Table 2 shows the accuracy of different features and methods. We take $\sigma = 5, 10, \tau = T/17, N_p = 500$ for DSTIP extraction and take the number of voxels for each cuboid to be $n_{xy} = 4, n_t = 3$. Wang et al.'s low-level feature LOP only achieves 42.5% while our DCSF feature achieves 83.6%, which is also better than Wang's high-level joint position feature. When concatenate our DCSF feature with joint position feature, it presents an accuracy of 88.2% which is higher than LOP combined with Joint position feature reported in [27] 85.75%.

We also compared our DCSF descriptor with widely used descriptors in RGB images: Cuboid descriptor and HOG descriptor. To control the variables, we use the same set of DSTIP locations detected by our DSTIP detector at $\sigma = 5, \tau = T/17$ for all the descriptors and perform no feature selection. For the Cuboid descriptor, we use a fixed cuboid size $\Delta_x = \Delta_y = 6\sigma$, because it does not handle different sizes. For the HOG descriptor, we incorporate the adaptable cuboid size and take $n_{xy} = 6, n_t = 4$ and use 4-bin histograms of gradient orientations, which is the best parameter for HOG on this dataset. Our DCSF descriptor performs significantly better than the Cuboid descriptor or gradient based descriptor even with adaptable cuboid size.

Figure 4 shows some examples of extracted DSTIPs on the MSRDailyActivity dataset using our detector. We also compared our DSTIP detector with widely used detectors in RGB images, including the Harris3D detector [11] and Cuboid detector [5]. We implemented the Cuboid detector and keep the same setting of spatial and temporal scale with our DSTIP detector. Figure 5 shows the STIPs extracted by the Cuboid detector and our DSTIP detector when take the STIPs at local maximum with the top 50,100,200,300,500,800 response values. As we can see, the Cuboid detector first captures the noise in the background, then gradually begins to capture a few points around the moving arm at $N_p = 200$, but those informative points are overwhelmed by the large number of noisy points. This also suggests that the noise is at a larger magnitude than the real movements. Our DSTIP detector effectively captures the movement of the arm, and noisy points begin to appear as late as $N_p = 800$, but the majority of the STIPs still gather around the person.

(a) drink-sit (b) drink-stand

Figure 6: STIPs extracted using Harris3D detector [11]

For the Harris3D detector, we use the code on-line[1] and use the standard parameters: number of spatial pyramid equals 3 combined with $\sigma^2 = 4, 8$, $\tau^2 = 2, 4$, $k = 0.0005$. For the tool to work, we smooth and scale the depth pixels to 0-255. Figure 6 shows the STIPs extracted. Only a small fraction of STIPs locates around the moving body parts, most of them appear near edges or static objects. We tried varying the parameters but it gives similar results.

Figure 7 shows the influence of parameters on the average accuracy of our algorithm. The parameter tested are No. of STIPs per video N_p, No. of bins for the depth histogram M, No. of voxels for a cuboid n_{xy}, n_t, support region L, and codebook size k.

6.3. The University of Texas dataset

Our dataset contains 10 actions: *hello, push, pull, boxing, step, forward-kick, side-kick, wave hands, bend, and clap hands*. These actions cover the movements of hands, arms, legs, and upper torso. Each action was collected from 10 different persons each performing the actions 3 times. The resolution of the depth map is 320×240. Each action sample spans about $8 - 46$ frames. We take $\sigma = 5, 10$, $\tau = T/8, T/5, T/3$ when filtering and take the number of voxels for each cuboid to be $n_{xy} = 4, n_t = 2$.

[1] http://www.di.ens.fr/ laptev/interestpoints.html

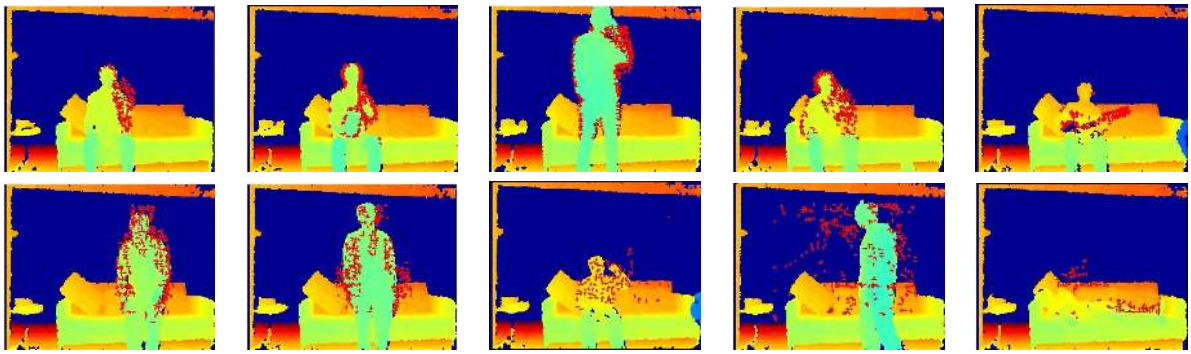
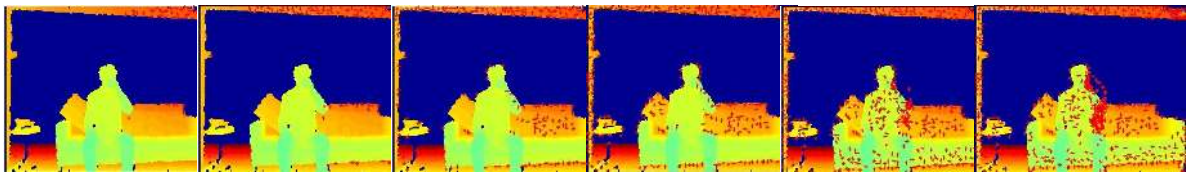
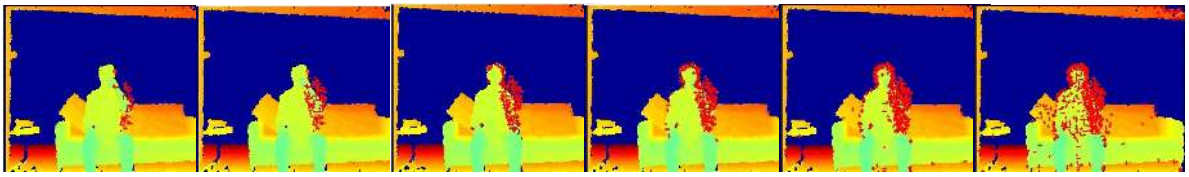


Figure 4: Example of STIPs extracted using our algorithm. They are projected on to x-y dimensions with one depth frame from the video for display. Action type from left to right, up to down: *drink-sit, eat, drink-stand,call cellphone, play guitar, sit down, stand up, toss, walk and lay-down*



(a) Cuboid detector



(b) DSTIP detector

Figure 5: Comparison of our DSTIP detector with Cuboid detector. Example video is action drink. Column from left to right is taken $N_p = 50, 100, 200, 300, 500, 800$ respectively.

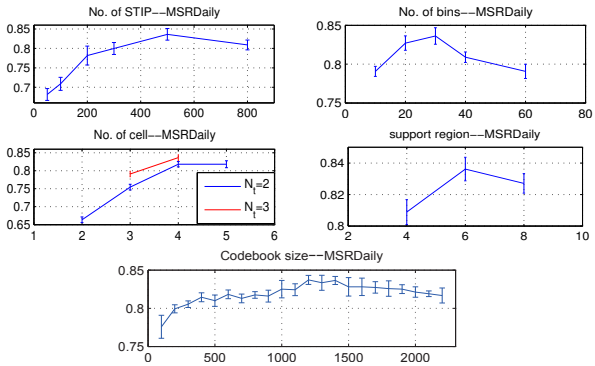


Figure 7: Parameter evaluation around optimum value on the MSRDailyActivity3D dataset. The average accuracy with the standard deviation denoted by error bar is plotted.

There is no skeleton information recorded so skeleton feature based algorithms [27, 30] cannot be applied onto it. On this dataset, we tried another method in which we take the 3D point clouds of the whole body in each frame and map it to a posture word. Then each action is represented by a sequence of posture words and we classify upon that (we refer to it as the "posture word method"). Table 3 gives the results of the two algorithms on three testing cases. The proposed DSTIP+DCSF pipeline performs significantly better than posture words method in that it focuses on the location of movement instead of trying to model the whole body, and the DSTIP pipeline automatically finds the movements without requiring segmentation of the human body as the posture word method does.

	Test One	**Test Two**	**Cross Subject**
DSTIP+DCSF	**93.5%**	**96.7%**	**85.8%**
Posture Word	83.89%	75.65%	79.57%

Table 3: Comparison of recognition rate on our own dataset. In **test one**, 1/3 of the samples were used as training samples and the rest as testing samples; in **test two**, 2/3 samples were used as training samples; In **cross subject test**, half of the subjects were used as training and the rest as testing.

Notice from the experiments that our algorithm does not depend on the availability of skeleton information or preprocessing as other methods do. By this means, our algorithm is a more general approach to processing depth videos and recognizing activities, which may also be used for a

wider variety of settings, e.g. group activities, local body parts activities, or non-human behavior studies.

7. Discussion and Conclusions

This paper presents algorithm to extract DSTIPs from depth videos and to calculate descriptors for the local 3D depth cuboid around the DSTIPs. The descriptor can be used to recognize actions of either humans or animals with no dependence on skeleton information or preprocessing like human detection, motion segmentation, tracking, or even image denoising or hole-filling. Thus it is more flexible than existing algorithms. It has been applied on three different datasets and presents better recognition accuracy than other state-of-the art algorithms based on either low-level features or high-level features.

Also, there is rich possibility for extensions. As shown in the experiment, when skeletal joint information is available, the DCSF can be concatenated with the joint features to bring more accurate recognition results. Or, joint locations can be regarded as a type of interest points and cuboids can be extracted from those locations. On the other hand, when the corresponding RGB video is available, the DCSF features can be easily combined with STIP features from RGB videos to combine the information from two sources. Additionally, the STIP locations extracted from the depth videos and RGB videos can be combined or filtered to provide more discriminate interest point locations and thus render better recognition performance.

8. Acknowledgement

Special thanks to Dr. Changbo Hu, the CVPR area chair, and the reviewers for their comments and corrections.

References

[1] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 2011.

[2] S. Belongie, K. Branson, P. Dollar, and V. Rabaud. Monitoring animal behavior in the smart vivarium. *Measuring Behavior*, 2005.

[3] C. Chang and C. Lin. Libsvm: a library for support vector machines. *TIST*, 2(3):27, 2011.

[4] Z. Cheng, L. Qin, Y. Ye, Q. Huang, and Q. Tian. Human daily action analysis with multi-view and color-depth data. In *Computer Vision–ECCV. Workshops and Demonstrations*, pages 52–61, 2012.

[5] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *PETS*, pages 65–72, 2005.

[6] J. Goles. Inside the race to hack the kinect. *New Scientist*, 208(2789):22, 2010.

[7] A. Hernández-Vela, M. Bautista, X. Perez-Sala, V. Ponce, X. Baró, O. Pujol, C. Angulo, and S. Escalera. Bovdw: Bag-of-visual-and-depth-words for gesture recognition. In *ICPR*, 2012.

[8] S. Ikemura and H. Fujiyoshi. Real-time human detection using relational depth similarity features. In *ACCV 2010*, pages 25–38.

[9] A. Kanezaki, H. Nakayama, T. Harada, and Y. Kuniyoshi. High-speed 3d object recognition using additive features in a linear subspace. In *ICRA*, pages 3128–3134, 2010.

[10] A. Klaser and M. Marszalek. A spatio-temporal descriptor based on 3d-gradients. In *BMVC*, 2008.

[11] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2):107–123, 2005.

[12] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8, 2008.

[13] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. *CVPR Workshop*, 1(5):9–14, 2010.

[14] B. Ni, G. Wang, and P. Moulin. Rgbd-hudaact: A color-depth video database for human daily activity recognition. In *ICCV Workshop*, pages 1147–1153, 2011.

[15] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79(3):299–318, 2008.

[16] M. Pauly, R. Keiser, and M. Gross. Multi-scale feature extraction on point-sampled surfaces. In *Computer graphics forum*, volume 22, pages 281–289, 2003.

[17] F. Porikli. Integral histogram: A fast way to extract histograms in cartesian spaces. In *CVPR*, volume 1, pages 829–836, 2005.

[18] R. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *ICRA*, pages 3212–3217, 2009.

[19] R. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *IROS*, pages 2155–2162, 2010.

[20] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8, 2007.

[21] 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 a single depth image. *CVPR*, 2011.

[22] B. Steder, R. Rusu, K. Konolige, and W. Burgard. Narf: 3d range image features for object recognition. In *IROS Workshop*, volume 44, 2010.

[23] J. Stuckler and S. Behnke. Interest point detection in depth images through scale-space surface analysis. In *ICRA*, pages 3568–3574, 2011.

[24] B. C. Vemuri, A. Mitiche, and J. K. Aggarwal. Curvature-based representation of objects from range data. *Image and vision computing*, 4(2):107–114, 1986.

[25] 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*, pages 252–259, 2012.

[26] J. Wang, Z. Liu, J. Chorowski, Z. Chen, and Y. Wu. Robust 3d action recognition with random occupancy patterns. In *ECCV*, pages 872–885. 2012.

[27] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, pages 1290–1297, 2012.

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

[29] L. Xia, C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *CVPR Workshop*, pages 20–27, 2012.

[30] X. Yang and Y. Tian. Eigenjoints-based action recognition using naïve-bayes-nearest-neighbor. In *CVPRW*, pages 14–19, 2012.

[31] H. Zhang and L. Parker. 4-dimensional local spatio-temporal features for human activity recognition. In *IROS*, pages 2044–2049, 2011.

[32] Y. Zhao, Z. Liu, L. Yang, and H. Cheng. Combing rgb and depth map features for human activity recognition. In *APSIPA ASC*, pages 1–4, 2012.