Real Time Action Recognition Using Histograms of Depth Gradients and Random Decision Forests

Hossein Rahmani Arif Mahmood Du Q. Huynh Ajmal Mian
The School of Computer Science and Software Engineering
The University of Western Australia, Crawley, WA 6009 Australia
hossein@csse.uwa.edu.au, {arif.mahmood, du.huynh, ajmal.mian}@uwa.edu.au

Abstract

We propose an algorithm which combines the discriminative information from depth images as well as from 3D joint positions to achieve high action recognition accuracy. To avoid the suppression of subtle discriminative information and also to handle local occlusions, we compute a vector of many independent local features. Each feature encodes spatiotemporal variations of depth and depth gradients at a specific space-time location in the action volume. Moreover, we encode the dominant skeleton movements by computing a local 3D joint position difference histogram. For each joint, we compute a 3D space-time motion volume which we use as an importance indicator and incorporate in the feature vector for improved action discrimination. To retain only the discriminant features, we train a random decision forest (RDF). The proposed algorithm is evaluated on three standard datasets and compared with nine state-of-the-art algorithms. Experimental results show that, on the average, the proposed algorithm outperform all other algorithms in accuracy and have a processing speed of over 112 frames/second.

1. Introduction

Automatic human action recognition in videos is a significant research problem with wide applications in various fields such as smart surveillance and monitoring, health and medicine, sports and recreation. Human action recognition is a challenging problem mainly because significant intra-action variations exist due to the differences in viewpoints, visual appearance such as color, texture and shape of clothing, scale variations including performer's body size and distance from the camera, and variations in the speed of action performance. Some of these challenges have recently been simplified by the availability of real time depth cameras, such as the Microsoft Kinect depth sensor, which offer many advantages over the conventional RGB cameras. Depth sensors are color and texture invariant and have eased the task of object segmentation and background subtraction. Moreover, depth images are not sensitive to illumination and can be captured in complete darkness using active sensors like Kinect.

In RGB images, pixels indicate the three color reflectance intensity of a scene, whereas in depth images, pixels represent the calibrated depth values. Although depth images help overcome the problems induced by clothing color and texture variation, many challenges still remain such as variations in scale due to body size and distance from the sensor, non-rigid shape of the clothing and variations in the style and speed of actions. To handle scale and orientation variations, different types of normalizations have been proposed in the literature [6, 5, 3, 12, 15, 7]. Intra-action variations are catered by information integration at coarser levels, for example, by computing 2D silhouettes [6, 3, 5] or depth motion maps [15]. However, coarse information integration may result in the loss of classification accuracy due to suppression of discriminative information. Moreover, these algorithms must handle large amounts of data and, therefore, have high computational complexity. Some other techniques have tried to overcome these challenges by mainly depending on the 3D joint positions [14, 13]. However, joint position estimation may be inaccurate in the presence of occlusions [13]. Furthermore, in some applications, such as hand gesture recognition, where joint positions are not available, these approaches cannot be adopted.

In this paper, we propose a new action classification algorithm by integrating information from both the depth images and the 3D joint positions. We normalize the inter-subject temporal and spatial variations by dividing the depth sequence into a fixed number of small subvolumes. In each subvolume we encode the depth and the depth gradient variations by making histograms. To preserve the spatial and temporal locality of each bin, we map it to a fixed position on the real line by using an invertible mapping function. Thus we obtain local information integration in a spatiotemporal feature descriptor.

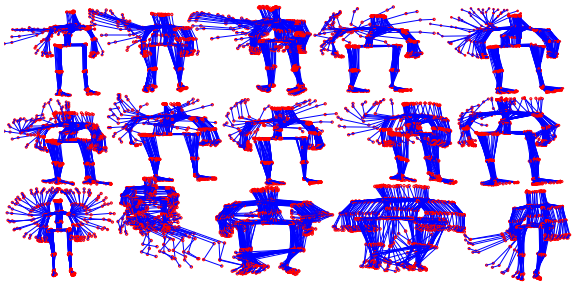

Figure 1. Skeleton plots for the first 15 actions in the MSR Action 3D dataset [6, 13]. For some actions, skeletons are discriminative while some skeletons are subsets of others.

Temporal variations of 3D joint positions (Fig. 1) also contain discriminative information of actions; however, it may be noisy and sometimes incorrect or unavailable. We handle these problems by capturing the dominant skeleton movements by computing local joint position difference histograms. As the joint position estimation errors scatter over many histogram bins, their effect is suppressed. Also those joints that are more relevant to a certain action exhibit larger motions and sweep larger volumes compared to non-relevant joints (Fig. 1). For each joint, we compute a 3D space-time motion volume (Fig. 2(c)) which is used as an importance indicator and incorporated in the feature vector for improved action discrimination. To give more importance to the joint motion volume, we divide it into a fixed number of cells and encode the local depth variations for each cell. Note that these features are not strictly based on the joint trajectories and only require the joint extreme positions which are robustly estimated. The joint position inaccuracies due to occlusions within these extreme positions do not affect the quality of the proposed feature.

We evaluate the proposed algorithm on three standard datasets [12, 5, 6, 13] and compare it with nine state-of-the-art methods [12, 13, 7, 6, 14, 11, 2, 4, 5]. The proposed algorithm outperforms all existing methods while achieving a processing speed of over 112 frames/sec.

1.1. Contributions

1. The proposed algorithm combines the discriminative information from depth images as well as from 3D joint positions to achieve high action recognition accuracy. However, unlike [13, 14], the proposed algorithm is also applicable when joint positions are not available.
2. To avoid the suppression of subtle discriminative information, we perform local information integration and normalization which is in contrast to previous techniques which perform global integration and normalization. Our feature is a collection of many independent local features. Consequently, it has more chances of success in the presence of occlusions.
3. Encoding joint importance by using joint motion volume is an innovative idea and to the best of our knowledge, has not been done before.
4. To retain only the discriminant features, we train a random decision forest (RDF). Smaller feature dimensionality leads to fast feature extraction and subsequent classification. As a result, we obtain very fast speedup.

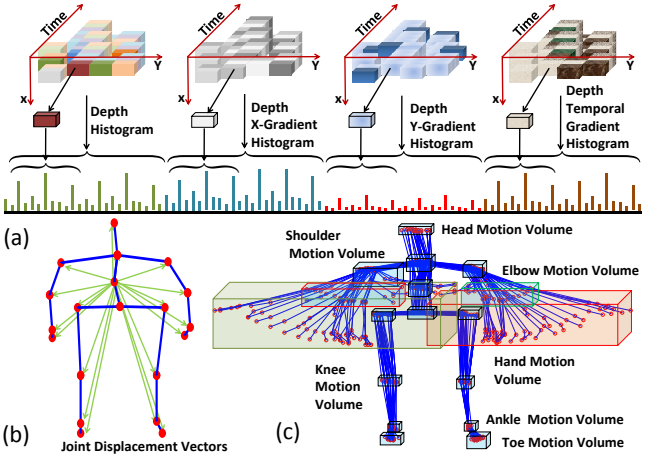

Figure 2. (a) *3D spatiotemporal depth and depth-Gradient histograms* for each space-time subvolume are mapped to the line of real numbers using one-to-one mapping function. (b) *3D Skeleton Displacement Vectors* are extracted from each frame. (c) Features are extracted from *Joint Movement Volume* for each joint.

2. Related Work

Action recognition using depth images has become an active research area after the release of Microsoft Kinect depth sensor. Real time computation of 3D joint positions [9] from the depth images has facilitated the application of skeleton based action recognition techniques. In this context, the depth based action recognition research can be broadly divided into two categories, namely, algorithms mainly using depth images [6, 5, 3, 12, 15, 7] and others mainly using joint positions [14, 13].

Some algorithms in the first category have exploited silhouette and edge pixels as discriminative features. For example, Li et al. [6] sampled boundary pixels from 2D silhouettes as a bag of features. Yang et al. [15] added temporal derivative of 2D projections to get Depth Motion Maps (DMM). Vieira et al. [11] computed silhouettes in 3D by using the space-time occupancy patterns. The spatiotemporal depth volume was divided into cells. A filled cell was represented by 1, an empty cell by 0 and a partially-filled cell by a fraction. An ad hoc parameter was used to differentiate between fully and partially filled cells. No automatic mechanism was proposed for the optimal selection of this parameter. Instead of these very simple occupancy features, Wang et al. [12] computed a vector of 8 Haar features on a uniform grid in the 4D volume. LDA was used to detect the discriminative feature positions and an SVM classifier was used for

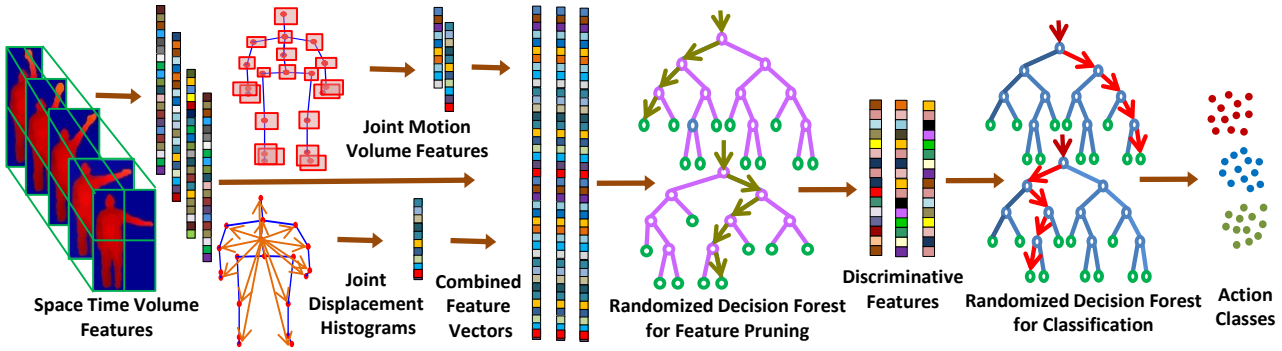

Figure 3. The proposed algorithm: If 3D joint positions are not available, then only depth and depth gradient features are used. Two different types of decision forests are trained. The first type is used only for feature pruning and is discarded after the training phase. The second type of forest is used for classification.

action classification. Both of the techniques [11, 12] have high computational complexity.

Tang et al. [10] proposed histograms of the normal vectors for object recognition in depth images. Given a depth image, spatial derivatives were computed and transformed to the polar coordinates (θ, ϕ, r). 2D histograms of (θ, ϕ) were used as object descriptors. Oreifej and Liu [7] extended it to the temporal dimension by adding time derivative. The gradient vector was normalized to unit magnitude and projected on fixed basis to make histograms. The last component of the normalized gradient vector was inverse of the gradient magnitude. As a result, information from very strong derivative locations, such as edges and silhouettes, may get suppressed.

In the second category, Yang and Tian [14] proposed pairwise 3D joint position differences in each frame and temporal differences across frames to represent an action. Since 3D joints cannot capture all discriminative information, the action recognition accuracy reduced. Wang et al. [13] extended previous approach by adding the depth histogram based features computed from a fixed region around each joint in each frame. In the temporal dimension, low frequency Fourier components were used as features. A discriminative set of joints were found using an SVM. Although their algorithm achieved high performance, it required 3D joint positions to be known beforehand.

3. Proposed Algorithm

We consider an action as a function operating on a four dimensional space with (x, y, t) being independent variables and the depth d being the dependent variable: $d = h(x, y, t)$. The discrimination of a particular action can be characterized by using the variations of the depth values with the variations of spatial and temporal dimensions. We capture these variations by 4D histograms of depth (Fig. 2(a)). The complete algorithm is shown in Fig. 3.

3.1. Histograms of Spatiotemporal Depth Function

In the depth frame at time (or frame number) t, we segment the foreground and the background using depth values and compute a 2D bounding rectangle containing the foreground only. For all the depth frames in one action, we compute a 3D action volume $V = P \times Q \times T$ containing all the 2D rectangles. The spatial dimensions $P \times Q$ mainly depend upon the size of the subject and his relative distance from the sensor, while the temporal dimension T depends on the speed of execution of the action.

The size of V often varies during the performance of the same action, whether it is inter-subject or intra-subject execution. Therefore, some spatial and temporal normalization on V needs to be enforced. Instead of scale normalization, we divide V into n_v smaller volumes Δ_V, such that $n_v = p \times q \times t$, where $p = P/\Delta_P$ and $q = Q/\Delta_Q$ are the spatial divisions and $t = T/\Delta_T$ are the temporal divisions and (Δ_P, Δ_Q) are horizontal and vertical numbers of pixels and Δ_T is the number of frames included in the subvolume Δ_V, such that $\Delta_V = \Delta_P \Delta_Q \Delta_T$. For each Δ_V, we assign a unique identifier using and invertible mapping function to preserve the spatial and temporal position of Δ_V. As the size of V changes, the number of subvolumes n_v remains fixed while the size of each subvolume Δ_V changes.

First, the minimum non-zero depth $d_{\min} > 0$ and maximum depth $d_{\max}$ in the action volume V are found. The depth range $d_{\max} - d_{\min}$ is then divided into $n_d = (d_{\max} - d_{min})/\delta d$ equal steps, where δd is the step size. In each subvolume Δ_V, the frequency of pixels with respect to the depth range is calculated and a depth histogram vector $\mathbf{h}_d \in \mathbb{R}^{n_d}$ is computed using the quantized depth values:

$$\hat{d}_{x,y,t} = \left\lceil \frac{d(x, y, t)}{\delta d} \right\rceil, \quad (1)$$

$$\mathbf{h}_d\{\hat{d}_{x,y,t}\} = \mathbf{h}_d\{\hat{d}_{x,y,t}\} + 1, \forall (x, y, t) \in \Delta_V. \quad (2)$$

Since the subvolume size may vary across different action

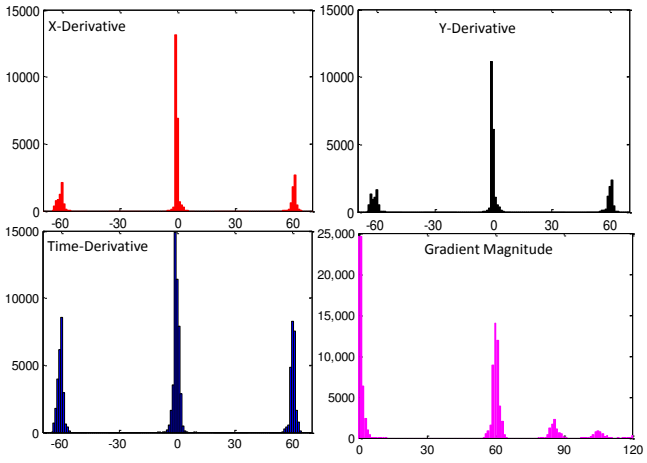

Figure 4. Histograms of depth gradients and the gradient magnitude for MSRGesture3D dataset. Strong depth gradients are obtained at the foreground edges, while gradients inside the object are weak and noisy.

videos, we normalize the histogram by the subvolume size:

$$\hat{\mathbf{h}}_d\{i\} = \frac{\mathbf{h}_d\{i\}}{\Delta_P \Delta_Q \Delta_T}, \text{ for } 1 \leq i \leq n_d. \tag{3}$$

Note that we represent the background by a depth value of zero and ignore the count of 0 values in Δ_V. A feature vector $\mathbf{f}_d \in \mathbb{R}^{n_d n_v}$ for action volume V is obtained by concatenating the depth histograms vectors for each Δ_V, i.e.,

$$\mathbf{f}_d = [\hat{\mathbf{h}}_{d1}^\top \quad \hat{\mathbf{h}}_{d2}^\top \quad \cdots \quad \hat{\mathbf{h}}_{dn_v}^\top]^\top. \tag{4}$$

The position of each histogram in this global feature vector is determined from the unique identifier of the corresponding Δ_V.

3.2. Histograms of Depth Derivatives

The depth function variations has also been characterized with the depth gradient:

$$\nabla d(x,y,t) = \frac{\partial d}{\partial x}\hat{\mathbf{i}} + \frac{\partial d}{\partial y}\hat{\mathbf{j}} + \frac{\partial d}{\partial t}\hat{\mathbf{k}}, \tag{5}$$

where the derivatives are given by: $\partial d/\partial x = (d(x,y,t) - d(x+\delta x, y, t))/\delta x$, $\partial d/\partial y = (d(x,y,t) - d(x, y+\delta y, t))/\delta y$, and $\partial d/\partial t = (d(x,y,t) - d(x,y,t+\delta t)/\delta t$. Fig. 4 shows the histograms of derivatives and the gradient magnitude $||\nabla d(x,y,t)||_2$ over the first video of the hand gesture dataset [5]. Each of the derivative histograms has three peaks. The peak centered at 0 is for the depth surface variations, while the peaks at $\Lambda \triangleq \pm|d_{\text{fg}} - d_{\text{bg}}|$ correspond to silhouette pixels, where d_{fg} is the foreground depth and d_{bg} is the background depth. In contrast to [7], we do not normalize the gradient magnitude to 1.0, because normalization destroys the structure by mixing the surface gradients and silhouette gradients. The un-normalized gradient magnitude histogram has four distinct peaks. The peak at the small gradient magnitudes corresponds to the surface variations while the other three peaks correspond to the silhouettes and have mean values given by Λ, $\sqrt{2}\Lambda$, and $\sqrt{3}\Lambda$. By making gradient histograms at the subvolume level, we encode the spatiotemporal position of a specific type of local depth variations and the position and shape of the silhouette into the feature vector.

For each depth derivative image, we find the derivative range as $\omega_{dx} = \max(\frac{\partial d}{\partial x}) - \min(\frac{\partial d}{\partial x})$ and divide it into n_{dx} uniform steps $\Delta\omega_{dx} = \omega_{dx}/n_{dx}$. The derivative values are quantized using $\Delta\omega_{dx}$ as follows

$$\hat{d}_x = \left\lceil \frac{\frac{\partial d}{\partial x}}{\Delta\omega_{dx}} \right\rceil. \tag{6}$$

From these quantized derivative values, the x-derivative histogram $\mathbf{h}_{\frac{\partial d}{\partial x}}$ is computed for each subvolume Δ_V. All the x-derivative histograms are concatenated according to the unique identifier for the Δ_V to make a global x-derivative feature vector $\mathbf{f}_{\frac{\partial d}{\partial x}}$. Three feature vectors $\mathbf{f}_{\frac{\partial d}{\partial y}}$, $\mathbf{f}_{\frac{\partial d}{\partial t}}$ and $\mathbf{f}_{\nabla d(x,y,t)}$ are also computed in a similar manner for $\frac{\partial d}{\partial y}$, $\frac{\partial d}{\partial t}$ and gradient magnitude $||\nabla d(x,y,t)||_2$ using n_{dy}, n_{dt}, and n_∇ uniform steps.

The five global feature vectors are concatenated to get the global feature vector $\mathbf{f}_g \in \mathbb{R}^{(n_d+n_{dx}+n_{dy}+n_{dt}+n_\nabla)n_v}$ for the full action volume:

$$\mathbf{f}_g = [\mathbf{f}_d^\top \quad \mathbf{f}_{\frac{\partial d}{\partial x}}^\top \quad \mathbf{f}_{\frac{\partial d}{\partial y}}^\top \quad \mathbf{f}_{\frac{\partial d}{\partial t}}^\top \quad \mathbf{f}_{\nabla d(x,y,t)}^\top]^\top. \tag{7}$$

The feature vector $\mathbf{f}_g$ is computed only from the depth images. In the following subsections we discuss the computation of the proposed 3D joint position features.

3.3. Histograms of Joint Position Differences

The Kinect sensor comes with a human skeleton tracking framework (OpenNI) [9] which efficiently estimates the 3D positions of 20 joints of the human skeleton (Fig. 1). Although the information of joint position is already present in the depth images in raw form, we observe that the estimated joint position information can be more efficiently used for action recognition. We propose two different types of features to be extracted using the 3D joint positions.

The proposed action recognition algorithm is applicable even if the 3D joint positions are not available. However, we observe that the 3D joints movement patterns are discriminative across different actions. Therefore, the classification performance should improve if joint movement patterns are incorporated within the feature vector. Let $[x_i, y_i, d_i, t]^\top$ be the 3D position of a joint, where (x_i, y_i) are spatial coordinates, d_i is the depth and t is the time for the i^{th} joint. In the current setting, in each frame, the 20 joint positions are determined. In each frame, we find the distance of each joint

i from a reference joint c:

$$(\Delta X_i, \Delta Y_i, \Delta d_i) = (x_i, y_i, d_i) - (x_c, y_c, d_c). \quad (8)$$

The reference joint may be a fixed joint position or centroid of the skeleton. We use the torso joint as the reference due to its stability. We capture the motion information of the skeleton over each action sequence by making one histogram for each component ΔX_i, ΔY_i, and Δd_i. These histograms are concatenated to make a global skeleton feature vector, which is included in the depth feature vector given by Eq. (7).

3.4. Joint Movement Volume Features

We observe that the joints relevant to an action exhibit larger movements and there are different sets of active and passive joints for each action. We incorporate this discriminative information in our algorithm by computing the 3D volume occupied by each joint (Fig. 2(c)). For the j^{th} joint, we find extreme positions over the full action sequence by computing maximum position differences along x- and y-axes. The lengths along the x-axis, y-axis and depth range is given by: $L_x = \big[\max(x_j) - \min(x_j)\big]$, $L_y = \big[\max(y_j) - \min(y_j)\big]$, $L_d = \big[\max(d_j) - \min(d_j)\big]$ and joint volume is given by

$$V_j = L_x L_y L_d. \quad (9)$$

For each joint, we incorporate the joint volume V_j and L_x, L_y, L_d in the feature vector. We also incorporate the centroid of each joint movement volume and the distance of each centroid from the torso joint into the feature. To capture the local spatiotemporal structure of the joint volume V_j, we divide L_d into n_{L_d} bins and compute the depth step size $\Delta_{L_d} = L_d / n_{L_d}$.

We divide each V_j into a fixed number of cells which have a different size compared to the subvolume Δ_V. For each cell, we compute the histogram of local depth variations using Δ_{L_d} as the bin size, starting from $\min(d_j)$. The histograms for the 20 joint volumes are concatenated to form a spatiotemporal local joint movement feature vector.

3.5. Random Decision Forests

A random decision forest [8, 1] is an ensemble of weak learners which are decision trees. Each tree consists of split nodes and leaf nodes. Each split node performs binary classification based on the value of a particular feature $\mathbf{f}_g[i]$. If $\mathbf{f}_g[i] \leq \tau_i$ (a threshold), then the action is assigned to the left partition, else to the right partition. If the classes are linearly separable, after $\log_2(c)$ decisions each action class will get separated from the remaining $c-1$ classes and reach a leaf node. For a given feature vector, each tree independently predicts its label and a majority voting scheme is used to predict the final label of the feature vector. It has been shown that random decision forests are fast and effective multi-class classifiers [9, 3]. The human action recognition problem tackled in this paper fits well in the random decision forest framework as the number of actions to be classified is quite large.

3.5.1 Training

In the random decision forest, each decision tree is trained on a randomly selected 2/3 part of the training data. The remaining 1/3 of the training data are out-of-bag (OOB) samples and used for validation. For each split node, from the set of total features having cardinality M, we randomly select a subset ξ of cardinality $m = \lceil \sqrt{M} \rceil$ and search for the best feature $\mathbf{f}_g[i] \in \xi$ and an associated threshold τ_i such that the number of class labels present across the partition is minimized. In other words, we want the boundary between left and right partitions to maximally match with the actual class boundaries and not to pass through (divide) any action class. This criterion is ensured by maximizing the reduction in entropy of the training data after partitioning, also known as *information gain*. Let $H(Q)$ be the original entropy of the training data and $H(Q|\{\mathbf{f}_g[i], \tau_i\})$ the entropy of Q after partitioning it into left and right partitions, Q_l and Q_r. Entropy reduction or information gain, G, is given by

$$G(Q|\{\mathbf{f}_g[i], \tau_i\}) = H(Q) - H(Q|\{\mathbf{f}_g[i], \tau_i\}), \quad (10)$$

where

$$H(Q|\{\mathbf{f}_g[i], \tau_i\}) = \frac{|Q_l|}{|Q|} H(Q_l) + \frac{|Q_r|}{|Q|} H(Q_r), \quad (11)$$

and $|Q_l|$ and $|Q_r|$ denote the number of data samples in the left and right partitions. The entropy of Q_l is given by

$$H(Q_l) = -\sum_{i \in Q_l} p_i \log_2 p_i, \quad (12)$$

where p_i is the number of samples of class i in Q_l divided by $|Q_l|$. The feature and the associated threshold that maximize the gain are selected as the splitting test for that node

$$\{\mathbf{f}_g[i], \tau_i\}^* = \arg \max_{\{\mathbf{f}_g[i], \tau_i\}} G(Q|\{\mathbf{f}_g[i], \tau_i\}) \quad (13)$$

If a partition contains only a single class, it is considered as a leaf node and no further partitioning is required. Partitions that contain multiple classes are further partitioned until they all end up as leaf nodes (contain single classes) or the maximum height of the tree is reached. If a tree reaches its maximum height and some of its partitions contain labels from multiple classes, the majority label is used as its label.

3.5.2 Feature Pruning and Classification

After a set of trees have been trained, the *significance score* of each variable, $\mathbf{f}_g[i], \forall i$, in the global feature vectors, $\mathbf{f}_g$, can be assessed as follows [8, 1]. For the k^{th} tree in the forest,

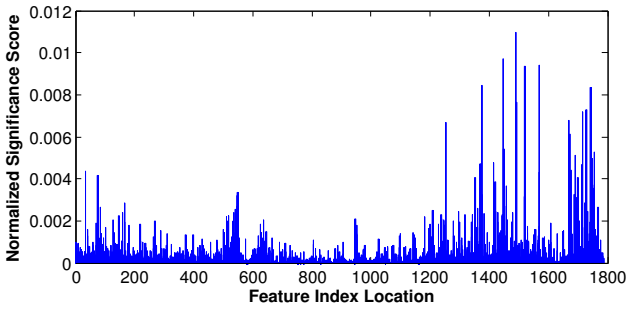

Figure 5. Normalized importance score $\hat{\Omega}[i]$ of 1800 features retained after feature pruning in MSRGesture3D Dataset.

1. Pass down all the OOB samples and count the number of votes cast for the correct classes, Ω_c^k.
2. To find the significance of $\mathbf{f}_g[i]$ in classification, randomly permute the values of $\mathbf{f}_g[i]$ in the OOB samples and pass these new samples down the tree, counting the correct number of votes, $\Omega_p^k[i]$.
3. Compute the importance as $\Delta\Omega^k[i] = \Omega_c^k - \Omega_p^k[i]$.

An average over K number of trees in the forest is considered as the *significance score* for feature $\mathbf{f}_g[i]$: $\Omega[i] = \frac{1}{K}\sum_{k=1}^{K}\Omega^k[i]$. The sum of significance scores of all features is normalized to unit magnitude: $\hat{\Omega}[i] = \frac{\Omega[i]}{\sum_{j=1}^{M}\Omega[j]}$. A random decision forest is first trained using the set of all proposed features. All the features whose normalized importance scores $\hat{\Omega}[i]$ are below a specified threshold are then deleted.

The normalized importance score for the selected features for gesture dataset is shown in Fig. 5. The original index location for each selected feature is also preserved. The subvolumes Δ_V not contributing any discriminative features are identified and skipped during the feature extraction step from the test video sequences. For the remaining subvolumes, only the required histogram bins are computed to reduce the feature extraction time. The lower dimensional feature vectors are fed to the trained random decision forest for classification.

4. Experiments and Results

The proposed algorithm is evaluated on three standard databases including MSR Action 3D [6, 13], MSR Gesture 3D [5, 12], and MSR Daily Activity 3D [13]. The performance of the proposed algorithm is compared with nine state-of-the-art algorithms including Random Occupancy Pattern (ROP) [12], Actionlet Ensemble [13], HON4D [7], Bag of 3D Points [6], Eigen Joints [14], Space Time Occupancy Patterns(STOP) [11], Convolutional Networks [2], 3D Gradient based descriptor [4], and Cell and Silhouette Occupancy Features [5]. For the Random Decision Forest (RDF), we use the implementation of [1]. Except for HON4D, all accuracies are reported from original papers or from the implementations of [13] and [7]. For HON4D, the code and feature set were obtained from the original author.

4.1. MSR Gesture 3D Dataset

The proposed algorithm with only depth histograms and depth-gradient histograms is evaluated on MSR Gesture 3D dataset [12, 5] because it does not contain the 3D joint positions (Fig. 6). This dataset contains 12 American Sign Language (ASL) gestures for *bathroom, blue, finish, green, hungry, milk, past, pig, store, where, letter J,* and *letter Z.* Each gesture is performed 2 or 3 times by each of the 10 subjects. In total, the dataset contains 333 depth sequences. Two actions "store" and "finish" are performed with two hands while the rest with one hand.

We divide each video into $n_v = 700$ subvolumes with $(p, q, t) = (14, 10, 5)$. From each bin, we make a depth histogram with $n_d = 10$, and one similar size histogram each for the x, y and time derivatives. The average feature extraction time is 0.0045 sec/frame. The first RDF is trained with 300 trees in 14.1 sec and the second RDF is trained in 4.75 sec with 200 trees on the pruned feature vectors. The pruned feature vector length is 1800 which is 19 times smaller than the length 34176 of HON4D. The test time is 10^{-6} sec/frame. The overall test speed including all overheads is 222 frames/sec which is 13.7 times faster than HON4D.

For comparison with previous techniques, we use the leave-one-subject-out cross-validation scheme proposed by [12]. Experiments are repeated 10 times, each time with a different test subject. The performance of the proposed method is evaluated in three different settings. In the first setting, one RDF is trained to classify all gesture classes. The accuracy of our algorithm is 92.76% in this setting. In the second setting, we first find the number of connected components in each video and two different RDFs are trained for two connected components and for the one component cases. The accuracy of our algorithm in this setting is 93.61%. Note that all subjects performed the actions "store" and "finish" by two hands while subject one performed these actions by one hand causing six videos to be invalid. In the third setting, we exclude these six invalid videos from the experiment and the average accuracy of our algorithm on the remaining dataset becomes 95.29%.

In all the three settings, our algorithm outperformed existing state-of-the-art algorithms (Table 1). Note that [13] cannot be applied to this dataset because of the absence of 3D joint positions. Detailed speedup comparison of HON4D with the proposed algorithm is shown in Table 2. To show the recognition accuracy of each gesture, in setting 2, the confusion matrix is shown in Fig. 7.

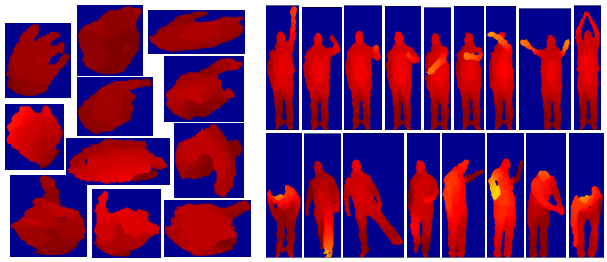

Figure 6. Sample depth images from MSR Gesture 3D dataset (left) and MSR Action 3D dataset (right).

Method	Accuracy (%)
Convolutional Network [2]	69.00
3D-Gradients Descriptor [4]	85.23
Action Graph on Occupancy Features [5]	80.5
Action Graph on Silhouette Features [5]	87.7
Random Occupancy Pattern [12]	88.5
HON4D [7]	92.45
Proposed Method (Setting 1)	**92.76**
Proposed Method (Setting 2)	**93.61**
Proposed Method (Setting 3)	**95.29**

Table 1. Comparison of action recognition rate on MSRGesture3D.

Algo	Data	Mean±STD	Max	Min	5/5
HON4D	Gesture	92.4±8.0	100	75	-
Proposed	Gesture	93.6± 8.3	100	77.8	-
HON4D	Action	82.1±4.2	90.6	69.5	88.9
Actionlet	Action	-	-	-	88.2
Proposed	Action	82.7±3.3	90.3	70.9	88.8
HON4D	DailyAct	73.20±3.92	80.0	66.25	80.0
Actionlet	DailyAct	-	-	-	85.75
Proposed	DailyAct	74.45±3.97	81.25	67.08	81.25

Table 2. Comparison with HON4D and Actionlet. Mean±STD are computed over 10 folds for the Gesture dataset, 252 folds for the Action dataset and 14 folds for the Daily Activity Dataset. 5/5 means subjects {1,2,3,4,5} used in Action and {1,3,5,7,9} used in Daily Activity for training.

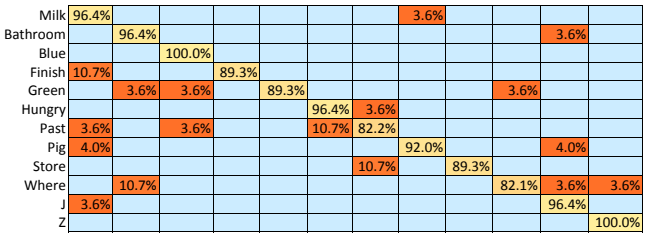

Figure 7. Confusion matrix of the proposed algorithm on MSR Gesture 3D dataset.

4.2. MSR Action 3D dataset

The proposed algorithm with full feature set is evaluated on MSR Action 3D dataset which consists of both depth sequences (Fig. 6) and skeleton data (Fig. 1) of twenty human actions: *high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, sideboxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up and throw*. Each action is performed by 10 subjects two or three times. The frame rate is 15 frames per second and the resolution is 320×240. The dataset consists of 567 actions samples. These actions were chosen in the context of interactions with game consoles and cover a variety of movements related to torso, legs, arms and their combinations.

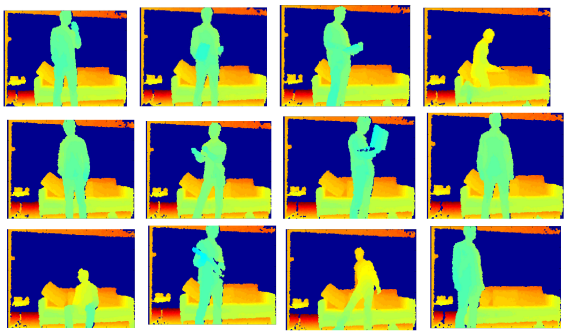

Figure 8. Sample depth images from MSR Daily Activity 3D dataset. This dataset is more challenging because the images contain background objects very close to the subject.

We divide each action video into $n_v = 500$ subvolumes with $(p, q, t) = (10, 10, 5)$. For each bin, we make one depth histogram with $n_d = 5$, and three gradient histograms of the same size. For joint motion features, we divide each volume into $n_c = 20$ cells with $(p, q, t) = (2, 2, 5)$ and for each cell a local depth variation histogram is made with $n_d = 5$. The average feature extraction time per frame is 0.0071 sec. An RDF for feature pruning is trained in 2.93 sec and RDF for classification is trained in 1.07 sec. The pruned feature vector length is 1819 which is 9.83 times smaller than the length 17880 of HON4D. Overall, the test speed including all overheads is 140 frames/sec which is 16 times faster than HON4D.

Experiments are performed using five subjects as training and five subjects as test. The experiments are repeated 256 folds exhaustively as proposed by [7]. The average accuracy obtained is 82.7±3.3% which is higher than the average accuracy 82.1±4.2% of HON4D (Table 2). Also, the standard deviation of our algorithm is significantly less than the HON4D. Our paired t-test shows that our results are statistically very significant: $p < 0.003$. For the Actionlet method [13] apart from the accuracy of one fold, no parameters are available. Comparison with other techniques for subjects {1,2,3,4,5} as training and others for testing, as used by [13, 7] is given in Table 3.

4.3. MSR Daily Activity 3D Dataset

The MSR Daily Activity 3D dataset [13] contains 16 daily activities including *drink, eat, read book, call cell*

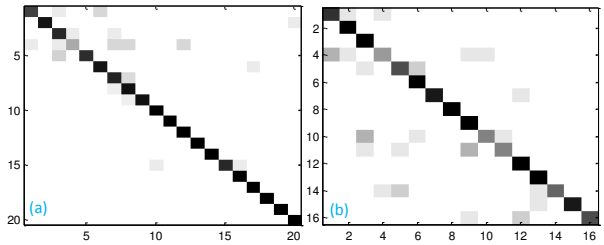

Figure 9. The confusion matrix of the proposed method for (a) MSR Action 3D dataset (b) MSR Daily Activity 3D dataset.

Method	Accuracy (%)
Depth Motion Maps(DMM) [15]	85.52
3D-Gradients Descriptor [4]	81.43
Action Graph on Bag of 3D Points [6]	74.70
Eigenjoints [14]	82.30
STOP Feature [11]	84.80
Random Accupancy Pattern [12]	86.50
Actionlet Ensemble [13]	88.20
HON4D [7]	88.90
Proposed Method(FAV Features)	81.50
Proposed Method(JMV Features)	81.10
Proposed Method(JDH Features)	78.20
Proposed Method(All Features)	88.82

Table 3. Comparison of the proposed algorithm with the current state-of-the-art techniques on MSR-Action3D dataset. Accuracy of the proposed algorithm is reported for different features types: Full Action Volume Features (FAVF), Joint Displacement Histograms (JDH),Joint Movement Volume Feature (JMVF). For a detailed accuracy comparison with HON4D, see Table 2.

phone, write on a paper, use laptop, use vacuum cleaner, cheer up, sit still, toss paper, play game, lay down on sofa, walk, play guitar, stand up, and *sit down*. Each activity is performed two times by each of the 10 subjects in two different poses: *sitting on a sofa* and *standing* (Fig. 8). In total, the dataset contains 640 RGB-D sequences at 30 frames per second and 320×240 resolution.

We use half of the subjects for training and the rest for testing in 14-fold experiments, each time randomly selecting the training subjects. Average feature extraction time is 0.0089 sec/frame. Feature pruning RDF is trained in 6.96 sec and classification RDF required only 0.43 sec. The feature vector length is 1662, which is 90.97 times smaller than the length 151200 of HON4D. The proposed algorithm has achieved average accuracy $74.45 \pm 3.97\%$ which is higher than the accuracy of HON4D (Table 2). Overall, the testing speed including all overheads is 112 frames/sec which is 10.4 times faster than HON4D.

5. Conclusion and Future Work

In this paper we propose a fast and accurate action recognition algorithm using depth videos and the 3D joint position estimates. The proposed algorithm is based on 4D depth and depth gradient histograms, local joint displacement histograms and joint movement occupancy volume features. Random Decision Forest (RDF) is used for feature pruning and classification. The proposed algorithm is tested on three standard datasets and compared with nine state-of-the-art algorithms. On average, the proposed algorithm is found to be more accurate than all other algorithms while achieving a testing speed of more than 112 frames/sec. Execution time can be further improved by parallel implementation of the proposed algorithm.

References

[1] L. Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.

[2] S. Ji, W. Xu, M. Yang, and K. Yu. 3D Convolutional Neural Networks for Human Action Recognition. *PAMI*, 35(1):221–231, 2013.

[3] C. Keskin, F. Kirac, Y. Kara, and L. Akarun. Real Time Hand Pose Estimation using Depth Sensors. In *ICCVW*, 2011.

[4] A. Klaeser, M. Marszalek, and C. Schmid. A Spatio-Temporal Descriptor Based on 3D-Gradients. In *BMVC*, 2008.

[5] A. Kurakin, Z. Zhang, and Z. Liu. A Real Time System for Dynamic Hand Gesture Recognition with a Depth Sensor. In *EUSIPCO*, pages 1975–1979, 2012.

[6] W. Li, Z. Zhang, and Z. Liu. Action Recognition based on a Bag of 3D Points. In *CVPRW*, 2010.

[7] O. Oreifej and Z. Liu. HON4D: Histogram of Oriented 4D Normals for Activity Recognition from Depth Sequences. In *to appear in CVPR*, 2013.

[8] J. R. Quinlan. Induction of Decision Trees. *Machine Learning*, pages 81–106, 1986.

[9] 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*, pages 1297–1304, 2011.

[10] S. Tang, X. Wang, X. Lv, T. Han, J. Keller, Z. He, M. Skubic, and S. Lao. Histogram of Oriented Normal Vectors for Object Recognition with a Depth Sensor. In *ACCV*. 2012.

[11] A. W. Vieira, E. Nascimento, G. Oliveira, Z. Liu, and M. Campos. STOP: Space-Time Occupancy Patterns for 3D Action Recognition from Depth Map Sequences. In *CIARP*, pages 252–259, 2012.

[12] 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.

[13] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining Actionlet Ensemble for Action Recognition with Depth Cameras. In *CVPR*, pages 1290 –1297, 2012.

[14] X. Yang and Y. Tian. EigenJoints-based Action Recognition using Naive Bayes Nearest Neighbor. In *CVPRW*, 2012.

[15] X. Yang, C. Zhang, and Y. Tian. Recognizing Actions using Depth Motion Maps-based Histograms of Oriented Gradients. In *ACM ICM*, 2012.