Kinematic-Layout-aware Random Forests for Depth-based Action Recognition

Seungryul Baek[1]
s.baek15@imperial.ac.uk
Zhiyuan Shi[1]
z.shi@imperial.ac.uk
Masato Kawade[2]
kawade@ari.ncl.omron.co.jp
Tae-Kyun Kim[1]
tk.kim@imperial.ac.uk

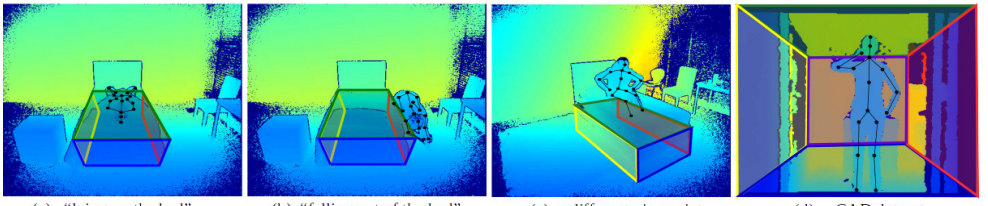
[1] Imperial College London.
London, UK
[2] Omron Corporation.
Japan

Abstract

This paper tackles the problem of 24 hours monitoring patient actions in a ward such as "lying on the bed", "stretching an arm out of the bed" and "falling out of the bed". In the concerned scenario, 3D geometric information (*e.g.* relations between scene layouts and body kinematics) is important to reveal the actions; however securing them at testing itself is a challenging problem. Especially in our data, securing human skeletal joints at testing time is not easy due to unique and diverse human posture. To address the problem, we propose a kinematic-layout-aware random forest considering the geometry between scene layouts and skeletons (*i.e. kinematic-layout*), secured in the offline manner, in the training of forests to maximize the discriminant power of depth appearance. We integrate the kinematic-layout in the split criteria of random forests to guide the learning process by 1) measuring the usefulness of kinematic-layout information and switching the use of kinematic-layout, and 2) implicitly closing the gap between two distributions obtained by the kinematic-layout and the appearance, if the kinematic-layout appears useful. Experimental evaluations on our new dataset (PATIENT) demonstrate that our method outperforms various state-of-the-arts for this problem. We have also demonstrated accuracy improvements by applying our method to conventional single-view and cross-view action recognition datasets (*e.g.* CAD-60, UWA3D Multiview Activity II).

1 Introduction

The recent emergence of cost-effective and easy-operation depth sensors have opened the door to a new family of methods [19, 23, 27, 28, 33, 60, 64] for action recognition from depth sequences. Compared to conventional color images, depth maps offer several advantages: 1) Depth maps encode rich 3D structural information, including informative shape, boundary, geometric cues of a human body and an entire scene. 2) Depth maps are insensitive to changes in lighting and illumination conditions that make it possible to monitor patient/animal 24/7. 3) It is invariant to texture and color variations, which benefits various recognition tasks.



(a) "lying on the bed" (b) "falling out of the bed" (c) a different viewpoint (d) CAD dataset

Figure 1: Depth maps visualized with kinematic-layout. Note that kinematic-layout has a potential to improve the ambiguity of depth appearance. (a)-(c) are depth maps from PATIENT dataset while (d) is the depth map from the conventional (CAD-60) dataset.

These advantages have promoted the fast pace development of depth-based techniques for action recognition. A number of spatio-temporal representations [24, 33, 37, 38, 56, 60] have been proposed to well represent the depth appearance, which is different from color maps. Recent approaches resorted to selecting the informative points around skeleton joints and modelling their temporal dynamics [10, 49, 55, 57, 68, 70], when human skeleton can be estimated from depth sequences. However, it is important to note that human pose estimation is known to be not always reliable and can fail when the human is not in an upright and frontal view position (e.g. lying) [58] or observed from unseen camera viewpoints [18]. Our scenario lies in these cases as in Fig 1 (a)-(c) and 2. To utilize the information which is not reliably obtainable at testing, we seek to formulate human poses and their 3D relations to layouts only during training by using their offline-secured ground-truths. Our aim is therefore to learn more robust classification models with more information at training and to obtain improved testing accuracy without explicit use of them at testing.

In order to investigate these issues, in this paper we make following contributions:

New action recognition dataset (PATIENT) has been collected containing patient behaviors (15 actions) in a ward by a depth camera. Actions in our dataset have close ties with scene layouts (*e.g*. *bed*, *floor*) and human body joints as in Fig. 1, 2; thus, utilizing kinematic-layout (*i.e.* 3D geometric relations between layouts and human body joints) is important to discriminate targeted actions. On the contrary, due to unique viewpoints and human poses in our dataset, skeleton information cannot be reliably tracked [18, 58] in a real-time manner, using a conventional depth sensor (*e.g*. kinect).

Kinematic-layout-aware random forest (KLRF) is introduced to improve the discriminant power of depth appearance by encoding the kinematic-layout. Considering that obtaining kinematic-layouts at testing itself is a challenging problem, we formulate KLRFs to use their offline-secured ground-truth implicitly at training and do not use them at testing (see Fig. 3 (a), (b)). We also make KLRFs encode the kinematic-layout adaptively: first cluster data samples into two groups where a group whose kinematic-layout is useful and a group whose kinematic-layout is less useful, then adaptively use it depending on the usefulness.

Both cross and single-view settings are tested on our own scenario (*i.e.* PATIENT dataset) and conventional (*i.e.* CAD60, UWA3D Multiview Activity II datasets) action recognition. Cross-view experiment is demonstrated to show the generalization ability of our method.

2 Related works

In this section, we review various cues available for the depth-based action recognition and random forest variants for considering the additional information other than the original

feature space:

Spatio-temporal depth cue. Spatial cue captures the static appearance information of single frames. Temporal cue conveys the movement of the observee or objects in the form of motion across frames. These two cues are usually encoded together as a spatio-temporal representation. The interest point detection and description has been widely studied [35, 39, 50] to provide reliable features for describing humans, objects or scenes. The spatio-temporal interest points (STIPs) are often adopted [9, 22, 24, 25, 58] for compact representations of activities and events. These conventional RGB-based methods do not perform well on depth maps [9, 21, 24, 46, 53, 59]. Recent efforts [26, 33, 51, 54, 60, 64, 69], therefore, have been devoted to developing reliable interest points and tracks for depth sequences. The interest points are extracted from low-level pixels [3, 26, 31] or mid-level parts [27, 40, 71]. In contrast to using local points, a holistic representation [26, 51, 54, 65] is recently popular as it is shown generally effective and computationally efficient. Yang *et al*. [65] extracted Histograms of Oriented Gradients (HOG) descriptors from Depth Motion Maps (DMM), where the DMM are generated by stacking motion energy of depth maps projected onto three orthogonal Cartesian planes. Wang *et al*. [56] defined Hierarchical Dynamic Motion Maps (HDMM) by using different offsets between frames and extracting Convolutional Neural Network (CNN) features from them. More recently, Rahmani *et al*. proposed a view-invariant descriptor HOPC [38] to deal with the 3D action recognition from unknown and unseen views. View-invariant representation, proposed in [37] has shown the state-of-the-art accuracy on both single-view and multi-viewed depth-based action recognition benchmarks.

Skeleton/pose cue. Pose estimation is beneficial for understanding human actions [13, 30, 66], while action recognition can also facilitate 3D human pose estimation [67]. The joint modeling of action and pose has been studied on RGB data [9, 11, 29, 32, 48, 63]. They perform pose estimation at testing stages, which either helps further action recognition or is helped by prior action recognition. In either case, accurate pose estimation at testing is aimed. A well trained skeleton tracker can provide a high-level cue for depth sequences. The use of skeleton joints has been suggested by [55, 61] for alleviating ambiguities in action recognition. Jiang *et al*. [55] represent the interaction between human body parts and environmental objects with an ensemble of human joint-based features. Skeleton joints have also been used to constrain the dictionary learning for feature representation [28]. There have been many later works [1, 10, 15, 16, 19, 20, 68, 72] that use skeleton/pose cues both at testing and training stages. In those works, estimated poses are relatively stable and provide good discrimination among actions, since most human poses are captured in the upright position and camera is located in front of humans. However, human pose estimation is not always stable due to the noisy depth maps, self-occlusions by camera views and diverse human poses [18, 38, 52]. To relieve the issue, Wang *et al*. [52] consider the best-K joint configurations to reduce the joint estimation errors. In our work, estimating human body joints is even more challenging, due to ambiguous and unique human poses (*e.g.* lying) in hospital environment. To avoid the unreliable dependency, we use the ground-truth of human poses and their 3D relation to layouts to aid model decision at training while bypassing their explicit estimation at testing. (see Fig. 4 (a), (b)).

Random forest variants. Standard random forests make the assumption that the output variables are independent over the parameter space. Conditional regression forest was presented by Sun *et al*. [44] and Dantone *et al*. [6], which demonstrates that the incorporation of prior information (such as human height, head pose) can enhance the dependency between output variables and latent variables, resulting in more accurate predictions. Similarly, Dapogny *et al*. [7] and Pham *et al*. [36] utilize expression prior and crowdedness prior respectively to

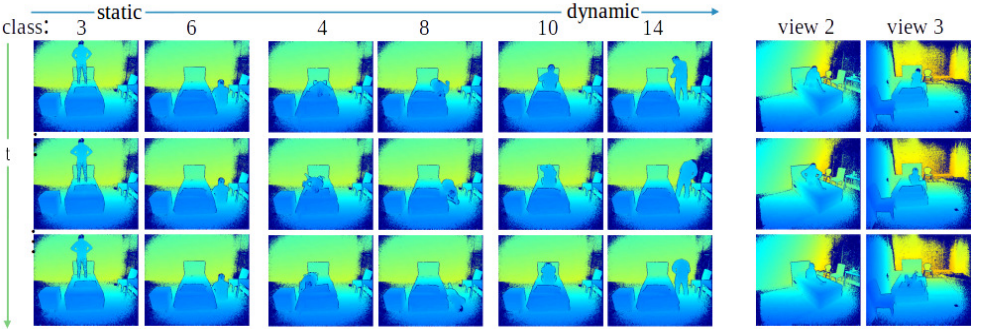


Figure 2: Examples of our PATIENT dataset. Our dataset contains both static (left side) and dynamic actions (right side). Action labels are given in Sec. 3. Examples for different views are also shown in last two columns.

reduce the variability within classes. Our method differs from existing conditional forests in that most of them exploit prior information to model the probability functions over the leaf nodes while we utilize the prior information at the split nodes during the tree growing.

While growing trees, some introduce the additional information that provides better explanations of the data. Tang *et al*. [47] exploited the pairwise relationship between synthetic and real data for the transfer learning of forests. Yang *et al*. [62] exploited the discrete additional prior explicitly to improve the quality of decision trees. Baek *et al*. [1] exploited pairwise and high-order associations of data samples as contexts. Differently to previous works, we incorporate continuous prior information to guide the model decision process both explicitly and implicitly. Furthermore, we adaptively use the information by empirically measuring its usefulness.

3 PATIENT dataset

We collect our own dataset (*i.e.* PATIENT) in a hospital scenario which contains 15 actions, performed by 10 subjects in 3 different viewpoints having close ties with *bed* and *floor* layouts. The dataset contains both static and dynamic actions and all 15 actions are: (1) lying, (2) sitting and (3) standing on the bed; (4-5) stretching body parts out of the bed when the patient is lying and sitting; (6-7) sitting and standing on the floor; (8) falling out of the bed; (9-15) suffering status of actions (1-8) except (3). In Table 1, we compare the PATIENT dataset with recently proposed action recognition datasets.

In most action DBs in Table 1, human joints are well captured by kinect sensors at testing, since humans are in upright positions (*e.g*. standing, sitting) and the camera is located in front of humans. In our scenario, humans' depth appearance is ambiguous due to their unique poses (*e.g*. lying, sitting back) and camera views (*i.e.* not human's frontal). Thus, capturing human joints is not also easy [18, 38]. Another characteristic of our dataset is that actions that we aim to recognize are closely related to 3D geometric relations between layouts (*i.e. bed*, *floor*) and human joints. Thus, we provide ground-truths for both human body joints and layout planes (*i.e. bed*, *floor*) to help reveal the actions. Also, we generate ground-truths for 5 layout planes (*i.e. floor*, *left wall*, *mid wall*, *right wall*, *ceiling*) for testing conventional (CAD60, UWA3D) datasets, as in Fig. 1 (d). Fig. 2 shows example frames of our dataset spanning static and dynamic actions.

Dataset	Geometric info.	Samples	Classes	Subjects	Views	Human poses
CAD-60 [45]	3D joints	60	12	4	1	Frontal/Upright
3D Action Pairs [33]	3D joints	360	12	10	1	Frontal/Upright
UTD-MHAD [3]	3D joints	861	27	8	1	Frontal/Upright
UWA3D [38]	3D joints	1075	30	10	5	Frontal/Upright
NTU [41]	3D joints	56880	60	40	80	Frontal/Upright
Ours	3D joints+**Layout**	450	15	10	3	**Various**

Table 1: Dataset comparison to recent benchmarks.

4 Kinematic-layout-aware random forest

In this section, we first introduce our appearance $\mathcal{A}$ and kinematic-layout $\mathcal{K}$ (Sec. 4.1) and then present how our approach exploits both information at training (Sec. 4.2). Testing stage of KLRFs and cross-view setting are explained in Sec. 4.3 and Sec. 4.4, respectively.

4.1 Appearance and kinematic-layout information

We construct the appearance $\mathcal{A}$ using the depth sequence V and the kinematic-layout $\mathcal{K}$ using layouts and skeleton joints for V, respectively. 1) We first extract depth cue $\mathbf{C}_t^D$, layout cue $\mathbf{C}_t^L$ and skeleton cue $\mathbf{C}_t^J$ for each frame t. 2) Then, we generate the spatio-temporal representation, $\mathcal{A}(V)$ and $\mathcal{K}(V)$ for a depth sequence V, by applying the Fourier transform on per-frame cues as in [37, 55]. The per-frame cues are defined as follows:

Depth cue $\mathbf{C}_t^D$: For each frame t, we extract the $4,096$ dimensional feature $\mathbf{C}_t^D$ from the $fc7$ layer of the CNN architecture proposed in [37]. This architecture is pre-trained on synthetic multi-view depth maps and shown to produce the state-of-the-art accuracy on both single and multi-viewed 3D action recognition benchmarks [37].

Skeleton cue $\mathbf{C}_t^J$: Skeleton cue $\mathbf{C}_t^J$ is encoded similar to [49, 68, 72] as $\mathbf{C}_t^J = [\mathbf{d}_t^P; \mathbf{d}_t^M; \mathbf{d}_t^O]$. (1) Skeleton *Pairwise* distance vector, $\mathbf{d}_t^P = [\mathbf{p}_1(t) - \mathbf{p}_2(t), ..., \mathbf{p}_p(t) - \mathbf{p}_q(t), ..., \mathbf{p}_{P-1}(t) - \mathbf{p}_P(t)]$ is defined for $\forall p, \forall q, p \neq q \in [1, P]$ to encode current frame's human poses. (2) Skeleton *Motion* vector, $\mathbf{d}_t^M = [\mathbf{p}_1(t) - \mathbf{p}_1(t-1), ..., \mathbf{p}_p(t) - \mathbf{p}_p(t-1), ..., \mathbf{p}_P(t) - \mathbf{p}_P(t-1)]$ is defined for $\forall p \in [1, P]$ to encode its temporal motion information. (3) Skeleton *Offset* vector, $\mathbf{d}_t^O = [\mathbf{p}_1(t) - \mathbf{p}_1(1), ..., \mathbf{p}_p(t) - \mathbf{p}_p(1), ..., \mathbf{p}_P(t) - \mathbf{p}_P(1)]$ is defined for $\forall p \in [1, P]$ to encode human offset information to their initial values *i.e.* $t = 1$. Skeleton cue can consider the spatial location of human body parts.

Layout cue $\mathbf{C}_t^L$: For each frame t, we propose to extract $\mathbf{C}_t^L$ by 3D displacements between layout planes $\mathbb{L} = \{\mathbf{L}_1, ...\mathbf{L}_l, ..., \mathbf{L}_L\}$ and skeleton joints $\mathbb{P}(t) = \{\mathbf{p}_1(t), ...\mathbf{p}_p(t), ..., \mathbf{p}_P(t)\}$ as:

$$\mathbf{C}_t^L = [\mathbf{d}_{t11}; ...; \mathbf{d}_{t1L}; \mathbf{d}_{t21}; ...; \mathbf{d}_{t2L}; ...; \mathbf{d}_{tP1}; ...; \mathbf{d}_{tPL}] \tag{1}$$

where $\mathbf{d}_{tpl} = \mathbf{p}_p(t) - \bar{\mathbf{p}}_{\mathbf{L}_l}$, $\mathbf{p}_p(t)$ is a 3-dimensional vector whose entry corresponds to its x, y and depth value and $\bar{\mathbf{p}}_{\mathbf{L}_l}$ is a projection of $\mathbf{p}_p(t)$ to the plane $\mathbf{L}_l$, respectively. This layout cue provides information on how humans interact with their environments. There exists a strong physical and functional coupling between human actions/poses and the 3D geometry of a scene [8, 12, 17]. We try to consider physical constraints to support actions such as "sitting" and "lying" by layout planes.

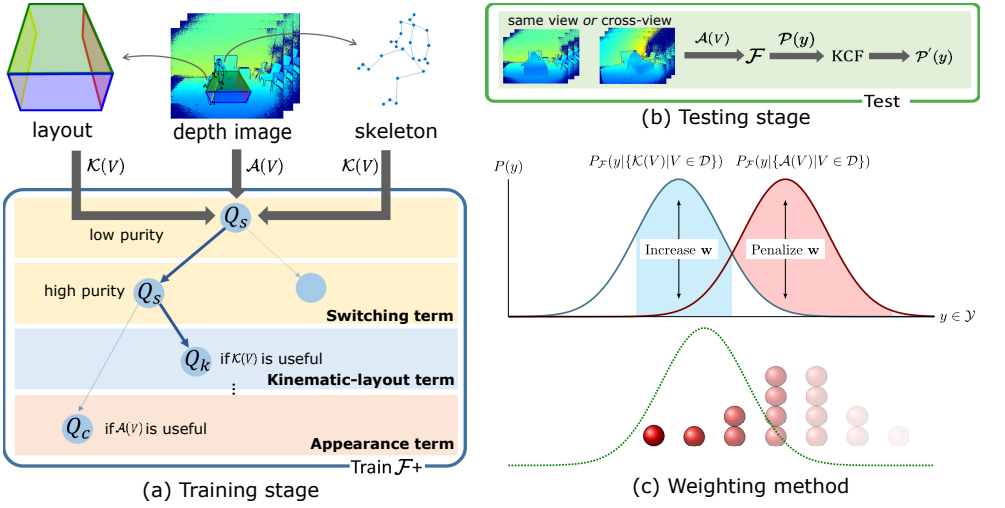


Figure 3: Flowchart of our method. (a) Training stage of KLRFs, (b) Testing stage of KLRFs, (c) Weighting method to reduce the gap between $P_{\mathcal{F}}(y|\{\mathcal{A}(V)|V\in\mathcal{D}\})$ and $P_{\mathcal{F}}(y|\{\mathcal{K}(V)|V\in\mathcal{D}\})$. Red balls denote samples constituting the appearance-based distribution $P_{\mathcal{F}}(y|\{\mathcal{A}(V)|V\in\mathcal{D}\})$ with their weights in fade-out. Green line denotes the gap-reduced class distribution.

4.2 Learning kinematic-layout-aware forests

Random forests (RFs) $\mathcal{F}$ aim to learn a mapping from the appearance $\mathcal{A}$ to the label set $\mathcal{Y}$:

$$\mathcal{F}:\mathcal{A}\mapsto\mathcal{Y}. \tag{2}$$

We propose kinematic-layout-aware random forests (KLRFs) $\mathcal{F}^+$ to optimize the mapping in Eq. 2 with the help of kinematic-layout $\mathcal{K}$ at training, if it appears useful :

$$\mathcal{F}^+:\begin{cases}\mathcal{A}\overset{\mathcal{K}}{\mapsto}\mathcal{Y}, & \text{, if } \mathcal{K} \text{ is useful}\\ \mathcal{A}\mapsto\mathcal{Y} & \text{, otherwise}\end{cases} \tag{3}$$

Same as RFs, KLRFs $\mathcal{F}^+$ are ensembles of binary trees, containing two types of nodes: *split* and *leaf*. At training, trees are grown by deciding the split function $\Psi(\mathcal{A}(\cdot)^\gamma,\tau)$ recursively from the root node, where $\mathcal{A}(\cdot)^\gamma$ denotes the γ-th value in the appearance feature and τ is a threshold. At each *split* node, arrived samples $V\in\mathcal{D}$ are divided into two subsets $\mathcal{D}_l$ and $\mathcal{D}_r$ ($\mathcal{D}_l\cap\mathcal{D}_r=\emptyset$) by a set of split function candidates $\{\Psi^c\}$ that is generated randomly. Samples whose $\mathcal{A}(V)^\gamma$ are less than τ go to the left child node ($\mathcal{D}_l$) while others go to the right child node ($\mathcal{D}_r$). Among candidates, the one that maximizes the quality function $\mathcal{Q}$ is selected as a split function Ψ^*:

$$\Psi^*=\arg\max_{\Psi\in\{\Psi^c\}}\mathcal{Q}(\Psi). \tag{4}$$

Trees are grown while sample number is above the minimum threshold (*i.e.* 5 in our experiments) or information gain is positive, where the information gain is defined as $\mathcal{Q}(\Psi^*)-\mathcal{Q}(\Psi^0)$ and Ψ^0 is the reference split that have all samples in $\mathcal{D}_l$ and no samples in $\mathcal{D}_r$. The terminating node becomes a *leaf* node and saves class distribution of arrived samples to use

it at testing. Note that $\mathcal{Q}$ in Eq. 4 can depend on both appearance $\mathcal{A}$ and kinematic-layout $\mathcal{K}$ since it is used at offline training, while $\Psi(\mathcal{A}(\cdot)^{\gamma},\tau)$ depends only on $\mathcal{A}$ to prevent the dependency of $\mathcal{K}$ at testing.

To train KLRFs $\mathcal{F}^{+}$ hierarchically as in Eq. 3 and Fig. 3 (a), we propose three types of quality functions: Q_s, Q_c and Q_k , which are called as switching, appearance and kinematic-layout term, respectively. Q_s first measures the usefulness of $\mathcal{K}$ as the *if-statement* of Eq. 3. Then, Q_c, Q_k selectively performs either $\mathcal{A}\mapsto\mathcal{Y}$ or $\mathcal{A}\overset{\mathcal{K}}{\mapsto}\mathcal{Y}$ depending on the node characteristics. The three quality functions are combined into a $\mathcal{Q}$ by variables α,β as:

$$\mathcal{Q}(\Psi)=\alpha Q_s+(1-\alpha)\{\beta Q_c+(1-\beta)Q_k\} \tag{5}$$

where variables α and β controls KLRFs to first select Q_s until certain number of data samples remain in a node and then select either Q_c or Q_k to perform further classification according to the node characteristics:

$$\alpha=\begin{cases}1 & ,\text{if } |\mathcal{D}|>\eta\\ 0 & ,\text{otherwise}\end{cases},\beta=\begin{cases}1 & ,\text{if } \zeta>\Delta\\ 0 & ,\text{otherwise}\end{cases}$$

where $|\mathcal{D}|$ is the number of samples in a current node, η is empirically set to 0.1 times total number of training samples, Δ is the ratio of samples having positive usefulness score $U(V)$ (Eq. 6) and $\zeta\in[0,1]$ is a randomly sampled value at each node. If Δ is high, it implies that $\mathcal{K}$ is useful in a current node. At the same time, the probability for $\zeta>\Delta$ becomes low and nodes tend to select Q_k more frequently than Q_c. If Δ is low, the opposite happens. Each tree's diversity obtained by this random configuration makes the KLRF ensemble become robust [14]. As in Fig. 3 (a), thanks to the hierarchical nature of trees, we are able to utilize different quality functions within a tree: nodes near the root select Q_s while nodes in the bottom gradually select either Q_c or Q_k. In the remainder of this section, we explain more about training with individual quality functions:

Pre-trained forests $\mathcal{F}_{\mathcal{K}}$, $\mathcal{F}_{\mathcal{A}}$: Before training each tree, we pre-train two forests $\mathcal{F}_{\mathcal{K}}$ and $\mathcal{F}_{\mathcal{A}}$ using out-of-bag (OOB) samples[1] and their kinematic-layout and appearance, respectively. Forests are pre-trained to obtain two class distributions for a sample V (*i.e.* $P(y|\mathcal{A}(V))=\mathcal{F}_{\mathcal{A}}(V)$, $P(y|\mathcal{K}(V))=\mathcal{F}_{\mathcal{K}}(V)$) and two class distributions for each node (*i.e.* $P_{\mathcal{F}}(y|\{\mathcal{K}(V)|V\in\mathcal{D}\})=\frac{1}{|\mathcal{D}|}\sum_{V\in\mathcal{D}}\mathcal{F}_{\mathcal{K}}(V)$, $P_{\mathcal{F}}(y|\{\mathcal{A}(V)|V\in\mathcal{D}\})=\frac{1}{|\mathcal{D}|}\sum_{V\in\mathcal{D}}\mathcal{F}_{\mathcal{A}}(V)$) at each tree training. Pre-trained forests are used whenever either Q_s or Q_k is selected for each node split.

Switching term Q_s: This term measures the usefulness of kinematic-layout $\mathcal{K}$ and selects Ψ that clusters samples into two groups: a group whose $\mathcal{K}$ is useful and another group whose $\mathcal{K}$ is less useful. The underline rationale of this term is our observation that kinematic-layout $\mathcal{K}$ does not always help improve the classification accuracy. For some samples, appearance $\mathcal{A}$ is enough or better than kinematic-layout $\mathcal{K}$ (see Fig. 4 (a), (b) and (d)). We define the score $U(V)\in[-1,1]$ to measure the usefulness of kinematic-layout for sample V:

$$U(V)=\mathcal{F}_{\mathcal{K},y^*}(V)-\mathcal{F}_{\mathcal{A},y^*}(V) \tag{6}$$

where y^*, $\mathcal{F}_{\mathcal{K},y^*}(V)$ and $\mathcal{F}_{\mathcal{A},y^*}(V)$ are the the ground-truth class label, y^*-th dimensional value of $\mathcal{F}_{\mathcal{K}}(V)$ and $\mathcal{F}_{\mathcal{A}}(V)$, respectively. The positive $U(V)$ implies that kinematic-layout $\mathcal{K}$ is empirically more useful than the appearance $\mathcal{A}$, while negative $U(V)$ means the opposite. The $Q_s=\left[1+\sum_{m\in\{l,r\}}\frac{|\mathcal{D}_m|}{|\mathcal{D}|}\text{var}\left(\{U(V)|V\in\mathcal{D}_m\}\right)\right]^{-1}$ prefers Ψ that clusters samples into

[1] Samples, not used in current tree training for **b**ootstrap **agg**regat**ing** (bagging).

left or right child nodes by the value of $U(V)$, where var$(\cdot)$ is the variance operator.

Appearance term Q_c: This term is same as Shannon entropy measure employed in standard classification RFs [43]. It measures the *uncertainty* of class distributions in $\mathcal{D}_l$ and $\mathcal{D}_r$ based on the appearance $\mathcal{A}$. It prefers Ψ that makes the class posterior distribution, empirically the class histograms, in $\mathcal{D}_l$ and $\mathcal{D}_r$ are dominated by a certain class.

Kinematic-layout term Q_k: To prevent the explicit usage of kinematic-layout $\mathcal{K}$ at testing, this term implicitly exploits the $\mathcal{K}$ at training, by minimizing the gap between two class distributions: $P_{\mathcal{F}}(y|\{\mathcal{K}(V)|V\in\mathcal{D}\})$, $P_{\mathcal{F}}(y|\{\mathcal{A}(V)|V\in\mathcal{D}\})$ at each node training. The gap is minimized by controlling each sample's weight and Q_k is defined based on the weighted distribution as in Eq. 8. The weight $\mathbf{w}^* = [w_1,...,w_{|\mathcal{D}|}]^\top \in \mathbb{R}^{|\mathcal{D}|\times 1}$ is optimized by:

$$\mathbf{w}^* = \min_{\mathbf{w}} ||\mathbf{A}\cdot\mathbf{w}-\mathbf{b}||_2^2 \text{ s.t. } \forall w_i \geq 0, \tag{7}$$

where w_i denotes each sample's weight, the i-th column of $\mathbf{A}\in\mathbb{R}^{|\mathcal{Y}|\times|\mathcal{D}|}$, $A_i \in \mathbb{R}^{|\mathcal{Y}|\times 1}$ corresponds to each sample's $P(y|\mathcal{A}(V))$ and $\mathbf{b}\in\mathbb{R}^{|\mathcal{Y}|\times 1}$ corresponds to $P(y|\{\mathcal{K}(V)|V\in\mathcal{D}\})$. The Eq. 7 can be optimized by the least-square solver with non-negativity constraints (*e.g.* `lsqnonneg` function in MATLAB). Meanwhile, as in Fig. 3 (c), a sample V', whose $P(y=l_1|\mathcal{A}(V'))=\mathcal{F}_{\mathcal{A},l_1}(V)$ is high, is emphasized if $P(y=l_1|\{\mathcal{K}(V)|V\in\mathcal{D}\}) > P(y=l_1|\{\mathcal{A}(V)|V\in\mathcal{D}\})$ while surpressed, otherwise (for $1\leq l_1\leq|\mathcal{Y}|$). Samples with high discrepancy can be benefitted by kinematic-layouts $\mathcal{K}$; thus they are emphasized and carefully considered for deciding Ψ while others are surpressed regarded as a noise. The Q_k is defined as the Shannon entropy measure on the weighted class histograms $n_{\mathbf{w}}(y,\mathcal{D}_m)$ as follows:

$$Q_k = \sum_{m\in\{l,r\}}\sum_{y\in\mathcal{Y}} n_{\mathbf{w}}(y,\mathcal{D}_m)\log\frac{n_{\mathbf{w}}(y,\mathcal{D}_m)}{\sum_{i=1}^{|\mathcal{D}|} w_i} \tag{8}$$

where $n_{\mathbf{w}}(y,\mathcal{D}) = \sum_{V\in\mathcal{D}} w_i\cdot\mathbb{I}(y=y^*)$ and $\mathbb{I}(\cdot)$ is an impusle function.

4.3 Inference by kinematic-layout-aware forests

At testing, as in Fig. 3 (b), $\mathcal{A}(V)$ is passed down the KLRFs $\mathcal{F}^+$ by learned split functions $\{\Psi(\mathcal{A}^\gamma(\cdot),\tau)\}$ until it reaches the leaf nodes, which store both the class distribution $P(y|V)$ and the kinematic vectors $\mathcal{K}(V)$. Split nodes decide its input V goes either to the left child (if $\mathcal{A}(V)^\gamma<\tau$) or to the right child (otherwise) according to learned split functions $\{\Psi(\mathcal{A}(\cdot)^\gamma,\tau)\}$. The responses are averaged to output the final $P(y|V)$ and $\hat{\mathcal{K}}(V)$ for each V.

4.4 Cross-view setting

Cross-view setting is challenging: the model is testified for unseen camera views, which have much impact on the depth appearance [37, 38]. Depth appearance $\mathcal{A}$ by [37] is view-invariant to a certain degree. To further help, we augment depth maps by synthetic rotations and translations as in [56], and consider their coherency using Q_v at training and kinematic consistency filter (KCF) at testing. Though both Q_v and KCF are designed for cross-view, we also apply them for single-view experiment and report their results (see Fig. 4 (e)).

View clustering term Q_v: This term enforces Ψ to cluster data samples according to the value of $\mathcal{K}(V)$ at training: $Q_v = \left[1+\sum_{m\in\{l,r\}}\frac{|\mathcal{D}_m|}{|\mathcal{D}|}\Lambda\left(\{\mathcal{K}(V)|V\in\mathcal{D}_m\}\right)\right]^{-1}$, where $\Lambda=$ trace(var$(\cdot)$) is defined as trace of a variance operator. Augmented data (*i.e.* translated, rotated) share the same kinematic-layout $\mathcal{K}$; thus they are clustered together by Q_v. As a

result, it enhances the view-invariance. Either Q_v or $\mathcal{Q}$ in Eq. 5 is randomly selected at each node, where random selection is known effective to mix up quality functions in a forest [14].

Kinematic consistency filter: At testing, after obtaining both $P(y|V)$ and $\hat{\mathcal{K}}(V)$ from the leaf nodes, we reduce noise by applying the KCF to $P(y|V)$. KCF exploits pairwise similarities of inferred kinematic-layout $\hat{\mathcal{K}}(V)$ to smooth the result by: $P^*(y|V) = \frac{1}{W_p}\sum_{J'\in\mathcal{S}(V)} P(y|J)g(||\hat{\mathcal{K}}(J) - \hat{\mathcal{K}}(J')||)$ where $W_p = \sum_{J'\in\mathcal{S}(V)} g(||\hat{\mathcal{K}}(J) - \hat{\mathcal{K}}(J')||)$ is a normalizing factor, $g(\cdot)$ is a Gaussian kernel and $S(V)$ is the augmented dataset of V. $P^*(y|V)$ is the final class distribution.

5 Experiments

We perform both single-view (on PATIENT, CAD-60 [45] datasets) and cross-view (on PATIENT, UWA3D Multiview Activity II [38] datasets) experiments to validate our methods. The "Baseline (RFs)" is the combination of depth appearance from [37] and standard RFs using additional translational, rotational data augmentation as in [56]. "Ours (KLRFs)" replaces RFs of "Baseline (RFs)" to KLRFs and consider the kinematic-layout $\mathcal{K}$ at training.

Same-view. We first evaluate our method for single-view action recognition using PATIENT and CAD60 datasets and each result is shown in Table 2 "View 1" column and Table 3, respectively. The classification accuracy is averaged over all classes, which corresponds to the mean of the confusion matrix diagonal. For PATIENT, we use the first 5 subjects as training and others as testing samples. We evaluate the recent state-of-the-art depth-based methods [33, 37, 38, 56, 60] using their publicly available codes. Our method produces a significant accuracy gain $(6-10\%)$ over these methods. For CAD60, we follow the cross-person experimental settings in [19, 55]. We also use two more measures (*i.e.* Precision/Recall) to compare with various state-of-the-arts for this dataset. KLRFs show good accuracy compared to depth-based approaches [33, 72]. Since this conventional dataset contains mostly upright humans with frontal views, most state-of-the-arts use real-time obtained skeleton joints at testing to obtain their results. Thus, for fair comparison, we combine skeleton cues to our method at testing: We train half of trees as RFs using pure skeletons and half of trees as KLRFs (denoted as "Ours (KLRFs+Skeleton)"). Also, "Baseline (RFs+Skeleton)" consists with half skeleton-based RFs and half depth-based RFs. Showing 5% accuracy gain to "Baseline (RFs+Skeleton)", "Ours (KLRFs+Skeleton)" shows the best result in Table 3.

Cross-view. We also applied our method to cross-view experiments using PATIENT and UWA3D datasets. For UWA3D, we follow the same experimental setting as in [37]. Averaged accuracy for all cross-views is reported in the "UWA3D Cross View" column of Table 2. For PATIENT dataset, we applied the same model trained in the single-view setting to View 2 and View 3 for cross-view tesing. The results are summarized in "View 2", "View 3" columns of PATIENT in Table 2, respectively. "Baseline (RFs)" often performs worse than [37] in cross-view experiments, while "Ours (KLRFs)" shows consistent accuracy improvement.

Usefulness score U vs. classes. In Fig. 4 (a), (b) and (d), We plot the averaged usefulness score $U(V) \in [-1,1]$ for samples in each action classes. Positive $U(V)$ means that $\mathcal{K}$ is more useful than $\mathcal{A}$, while negative $U(V)$ means the opposite. The results imply that in PATIENT dataset, static actions (*i.e.* (1)-(7)) are well explained by $\mathcal{K}$ rather than $\mathcal{A}$ while dynamic actions (*i.e.* (8)-(15)) are well classified by using only $\mathcal{A}$ without $\mathcal{K}$. In CAD-60 and UWA3D datasets, we also report the usefulness scores per each class, showing variations. Class index is given in Sec. 3 for PATIENT and in the supplementary page for other datasets.

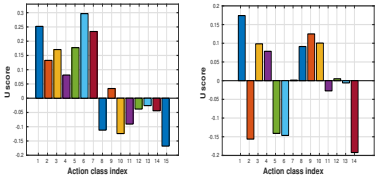
Utilizing $\mathcal{K}$ at testing. To test the strength of kinematic-layout $\mathcal{K}$, we report the classification accuracy explicitly using ground-truths of $\mathcal{K}$ as input features in Fig. 4 (c). Note that

Method	PATIENT			UWA3D
	View 1	View 2	View 3	Cross View
DCSF [51]	18.7	6.7	16.0	–
HON4D [43]	21.1	6.3	13.8	28.9
HOPC [33]	28.2	15.4	23.1	52.2
DMM [37]	29.3	19.3	24.0	–
Novel View [34]	43.8	23.8	32.5	76.9
Baseline (RFs)	47.8	21.5	27.2	77.1
Ours (KLRFs)	**53.2**	**27.5**	**36.2**	**80.4**

Table 2: Results for PATIENT (single-view (View 1), cross-view (View 2, 3)) and UWA3D (cross-view) datasets.

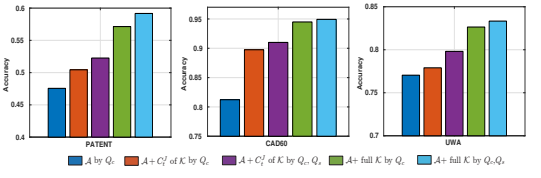
Method	Accuracy	Precision	Recall
Testing Input: Depth			
HON4D [43]	72.7	–	–
Zhu *et al.* [48]	75.0	–	–
Baseline (RFs)	81.6	93.2	78.6
Ours (KLRFs)	87.1	92.3	85.7
Testing Input: Skeleton			
GI *et al.* [30]	–	91.9	90.2
Shan *et al.* [40]	91.9	93.8	**94.5**
Cippitelli *et al.* [8]	–	93.9	93.5
Testing Input: Depth+Skeleton			
Actionlet Ensemble [36]	74.7	–	–
Zhu *et al.* [48]	87.5	93.2	84.6
Baseline (RFs+Skeleton)	89.7	92.9	89.3
Ours (KLRFs+Skeleton)	**94.1**	**97.5**	92.7

Table 3: Results for CAD-60 dataset.

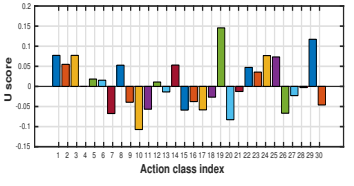


(a) U score vs. class in PATIENT.

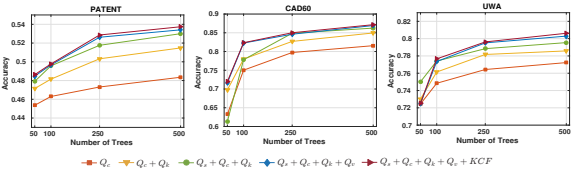
(b) U score vs. class in CAD60.



(c) Utilizing $\mathcal{K}$ at testing for 3 datasets.



(d) U score vs. classes in UWA3D.



(e) Parameter sensitivity: PATIENT(left), CAD(mid), UWA3D(right).

Figure 4: Further analysis result

utilizing $\mathcal{K}$ at testing is not realistic in our scenario; we conducted the experiments only for evaluation purpose using ground-truths of $\mathcal{K}$. We configure 3 different features $\{\mathcal{A}, \mathcal{A}+\mathbf{C}_t^J$ of $\mathcal{K}, \mathcal{A}+\mathcal{K}\}$ and two classifiers by standard RFs (*i.e.* Q_c) and KLRFs using Q_c+Q_s terms. The graph shows that $\mathcal{K}$ offers $5-10\%$ accuracy gain, when combined with $\mathcal{A}$.

Sensitivity to parameters. We evaluate the sensitivity of our model depending on tree numbers in Fig. 4 (e). The performance increases as tree numbers increase and saturates around 500 trees. Component analysis is further reported in the same figure.

6 Conclusion

In this paper, we study the problem of depth-based action recognition in a 24 hours-monitoring patient actions in a ward, with the goal of effectively recognizing human actions by exploiting the scene layout and skeleton information in the learning process. We propose the kinematic-layout-aware random forest to encode this prior information, thereby capturing more geometry that provides greater discriminant power in action classification.

Acknowledgement

The authors acknowledge the support of the Omron Corporation.

References

[1] S. Baek, K. I. Kim, and T-K. Kim. Real-time online action detection forests using spatio-temporal contexts. In *WACV*, 2017.

[2] C. Chen, R. Jafari, and N. Kehtarnavaz. UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth. In *ICIP*, 2015.

[3] Z. Cheng, L. Qin, Y. Ye, Q. Huang, and Q. Tian. Human daily action analysis with multi-view and color-depth data. In *ECCV Workshop*, 2012.

[4] G. Chéron, I. Laptev, and C. Schmid. P-CNN: pose-based CNN features for action recognition. In *ICCV*, 2015.

[5] E. Cippitelli, S. Gasparrini, E. Gambi, and S. Spinsante. A human activity recognition system using skeleton data from RGBD sensors. In *Computational Intelligence and Neuroscience*, 2016.

[6] M. Dantone, J. Gall, G. Fanelli, and L. V. Gool. Real-time facial feature detection using conditional regression forests . In *CVPR*, 2012.

[7] A. Dapogny, K. Bailly, and S. Dubuisson. Pairwise conditional random forests for facial expression recognition. In *ICCV*, 2015.

[8] V. Delaitre, D. F. Fouhey, I. Laptev, J. Sivic, A. Gupta, and A. A. Efros. Scene semantics from long-term observation of people. In *ECCV*, 2012.

[9] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, 2005.

[10] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015.

[11] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Progressive search space reduction for human pose estimation. In *CVPR*, 2008.

[12] D. F. Fouhey, V. Delaitre, A. Gupta, A. A. Efros, I. Laptev, and J. Sivic. People watching: human actions as a cue for single view geometry. *IJCV*, 2014.

[13] J. Gall, A. Yao, and L. V. Gool. 2D action recognition serves 3D human pose estimation. In *ECCV*, 2010.

[14] J. Gall, A. Yao, N. Razavi, L. V. Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *TPAMI*, 2011.

[15] G. Garcia-Hernando and T-K. Kim. Transition forests: learning discriminative temporal transitions for action recognition and detection. In *CVPR*, 2017.

[16] G. Garcia-Hernando, H. J. Chang, I. Serrano, O. Déniz, and T-K. Kim. Transition hough forest for trajectory-based action recognition. In *WACV*, 2016.

[17] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From 3d scene geometry to human workspace. In *CVPR*, 2011.

[18] A. Haquea, B. Peng, Z. Luo, A. Alahi, S. Yeung, and L. Fei-Fei. Towards viewpoint invariant 3D human pose estimation. In *ECCV*, 2016.

[19] J.-F. Hu, W.-S. Zheng, J. Lai, and J. Zhang. Jointly learning heterogeneous features for RGB-D activity recognition. In *CVPR*, 2015.

[20] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *ICCV*, 2013.

[21] T-K. Kim and R. Cipolla. Canonical correlation analysis of video volume tensors for action categorization and detection. *TPAMI*, 2009.

[22] A. Klaser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *BMVC*, 2008.

[23] Y. Kong and Y. Fu. Bilinear heterogeneous information machine for RGB-D action recognition. In *CVPR*, 2015.

[24] I. Laptev. On space-time interest points. *IJCV*, 2005.

[25] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[26] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3D points. In *CVPR Workshop*, 2010.

[27] C. Lu, J. Jia, and C.-K. Tang. Range-sample depth feature for action recognition. In *CVPR*, 2014.

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

[29] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *CVPR*, 2007.

[30] B. Mahasseni and S. Todorovic. Regularizing long short term memory with 3D human-skeleton sequences for action recognition. In *CVPR*, 2016.

[31] B. Ni, G. Wang, and P. Moulin. RGBD-HuDaAct: a color-depth video database for human daily activity recognition. In *ICCV Workshop*, 2011.

[32] B. X. Nie, C. Xiong, and S.-C. Zhu. Joint action recognition and pose estimation from video. In *CVPR*, 2015.

[33] O. Oreifej and Z. Liu. HON4D: histogram of oriented 4D normals for activity recognition from depth sequences. In *CVPR*, 2013.

[34] G. I. Parisi, C. Weber, and S. Wermter. Self-organizing neural integration of pose-motion features for human action recognition. In *Frontier in Neurobotics*, 2015.

[35] M. Pauly, R. Keiser, and M. Gross. Multi-scale feature extraction on point-sampled surfaces. *Computer Graphics Forum*, 2003.

[36] V.-Q. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada. COUNT forest: co-voting uncertain number of targets using random forest for crowd density estimation. In *ICCV*, 2015.

[37] H. Rahmani and A. Mian. 3D action recognition from novel viewpoints. In *CVPR*, 2016.

[38] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian. Histogram of oriented principal components for cross-view action recognition. *TPAMI*, 2016.

[39] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3D recognition and pose using the viewpoint feature histogram. In *IROS*, 2010.

[40] S. Sadanand and J. J. Corso. Action bank: a high-level representation of activity in video. In *CVPR*, 2012.

[41] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. NTU RGB+D: a large scale dataset for 3D human activity analysis. In *CVPR*, 2016.

[42] J. Shan and S. Akella. 3D human action segmentation and recognition using pose kinetic energy. In *IEEE Workshop on Advanced Robotics and its Social Impacts*, 2014.

[43] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[44] M. Sun, P. Kohli, and J. Shotton. Conditional regression forests for human pose estimation. In *CVPR*, 2012.

[45] J. Sung, C. Ponce, B. Selman, and A. Saxena. Unstructured human activity detection from RGBD images. In *ICRA*, 2012.

[46] T-K. Kim T.-H. Yu and R. Cipolla. Real-time action recognition by spatiotemporal semantic and structural forests. In *BMVC*, 2010.

[47] D. Tang, T.-H. Yu, and T-K. Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *ICCV*, 2013.

[48] N. Ukita. Iterative action and pose recognition using global-and-pose features and action-specific models. In *ICCV Workshop*, 2013.

[49] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3D skeletons as points in a lie group. In *CVPR*, 2014.

[50] B. C. Vemuri, A. Mitiche, and J. K. Aggarwal. Curvature-based representation of objects from range data. *Image and Vision Computing*, 1986.

[51] A. W. Vieira, E. R. Nascimento, G. L. Oliveira, Z. Liu, and M. F. M. Campos. STOP: space-time occupancy patterns for 3D action recognition from depth map sequences. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, 2012.

[52] C. Wang, Y. Wang, and A. L. Yuille. An approach to pose-based action recognition. In *CVPR*, 2013.

[53] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu. Action recognition by dense trajectories. In *CVPR*, 2011.

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

[55] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Learning actionlet ensemble for 3D human action recognition. *TPAMI*, 2014.

[56] P. Wang, W. Li, Z. Gao, J. Zhang, C. Tang, and P. Ogunbona. Action recognition from depth maps using deep convolutional neural networks. *IEEE Transactions on Human Machine Systems*, 2015.

[57] P. Wei, Y. Zhao, N. Zheng, and S.-C. Zhu. Modeling 4D human-object interactions for event and object recognition. In *ICCV*, 2013.

[58] G. Willems, T. Tuytelaars, and L. V. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008.

[59] S. Wu, O. Oreifej, and M. Shah. Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In *ICCV*, 2011.

[60] L. Xia and J. K. Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *CVPR*, 2013.

[61] L. Xia, C.-C. Chen, and J. K. Aggarwal. View invariant human action recognition using histograms of 3D joints. In *CVPR Workshop*, 2012.

[62] H. Yang and I. Patras. Privileged information-based conditional structured output regression forest for facial point detection. *TCSVT*, 2015.

[63] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *CVPR*, 2010.

[64] X. Yang and Y. L. Tian. Super normal vector for activity recognition using depth sequences. In *CVPR*, 2014.

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

[66] A. Yao, J. Gall, G. Fanelli, and L. V. Gool. Does human action recognition benefit from pose estimation? In *BMVC*, 2011.

[67] T.-H. Yu, T-K. Kim, and R. Cipolla. Unconstrained monocular 3D human pose estimation by action detection and cross-modality regression Forest. In *CVPR*, 2013.

[68] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The moving pose: an efficient 3D kinematics descriptor for low-latency action recognition and detection. In *ICCV*, 2013.

[69] H. Zhang and L. E. Parker. 4-dimensional local spatio-temporal features for human activity recognition. In *IROS*, 2011.

[70] X. Zhang, Y. Wang, M. Gou, M. Sznaier, and O. Camps. Efficient temporal sequence comparison and classification using gram matrix embeddings on a riemannian manifold. In *CVPR*, 2016.

[71] Y. Zhou, B. Ni, R. Hong, M. Wang, and Q. Tian. Interaction part mining: a mid-level approach for fine-grained action recognition. In *CVPR*, 2015.

[72] Y. Zhu, W. Chen, and G. Guo. Fusing spatiotemporal features and joints for 3D action recognition. In *CVPR Workshop*, 2013.