Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Ongoing human action recognition with motion capture

Mathieu Barnachon [a,*], Saïda Bouakaz [a], Boubakeur Boufama [b], Erwan Guillou [a]

[a] *Université de Lyon, CNRS, Université Lyon 1, LIRIS, UMR5205, F-69622, France*
[b] *School of Computer Science, University of Windsor, Windsor, ON, Canada N9B 3P4*

ARTICLE INFO

Article history:
Received 25 July 2012
Received in revised form
9 May 2013
Accepted 11 June 2013

Keywords:
Human action recognition
Histogram
Ongoing recognition
Human–computer interaction

ABSTRACT

Ongoing human action recognition is a challenging problem that has many applications, such as video surveillance, patient monitoring, human–computer interaction, etc. This paper presents a novel framework for recognizing streamed actions using Motion Capture (MoCap) data. Unlike the after-the-fact classification of completed activities, this work aims at achieving early recognition of ongoing activities. The proposed method is time efficient as it is based on histograms of action poses, extracted from MoCap data, that are computed according to Hausdorff distance. The histograms are then compared with the Bhattacharyya distance and warped by a dynamic time warping process to achieve their optimal alignment. This process, implemented by our dynamic programming-based solution, has the advantage of allowing some stretching flexibility to accommodate for possible action length changes. We have shown the success and effectiveness of our solution by testing it on large datasets and comparing it with several state-of-the-art methods. In particular, we were able to achieve excellent recognition rates that have outperformed many well known methods.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

Human action recognition is a challenging research problem in computer vision that, if solved, would enhance numerous applications in areas ranging from Human Computer Interface (HCI) to entertainment. For instance, human action recognition could help to identify suspicious activities. In entertainment applications, recognizing players' actions makes the game more attractive, more user-friendly and increases its potential. Given its importance in numerous applications, the problem of action recognition has attracted a great deal of research works over the last decades. Although ambiguous 2D images have been traditionally used as inputs for action recognition, numerous researchers have started using MoCap data for action recognition [30,44,40]. In particular, newly available low cost depth sensors, such as Microsoft Kinect and its real-time MoCap system [35], can be used to enhance the user's experience with games, serious games, presentation softwares, etc.

This paper proposes a novel examplar-based human action recognition system, that uses MoCap poses as input. We first extend the concept of histograms to the case of poses. Then, spatio-temporal series of poses are clustered to create a statistical representation of actions. Note that when an action consists of cycles, for example walking, its histogram, or part of it, is affected by a scale only. We have also introduced an incremental and memory efficient structure, the integral histogram, to allow for ongoing activity recognition. Finally, a dynamic programming algorithm, inspired from the Dynamic Time Warping (DTW) method [33], is used to compare sub-actions and to compute the recognition score between multiple human action instances. To the best of our knowledge, this paper is the first to propose a solution using histograms of 3D Motion Capture data for action recognition. The efficient formulation of histograms has made it possible to learn and recognize actions during their progress. We have validated our proposed approach with extensive tests on well-known benchmark datasets, and we have compared it to several state-of-the-art methods. The obtained results have clearly shown the success and effectiveness of our solution, even in the presence of noise and/or similar actions in the datasets.

2. Previous works

This section summarizes major previous works in human action recognition, and briefly surveys three related issues: the body skeletonization, the body shape analysis and the extraction of feature points. For an extensive survey on human action recognition, the interested reader may consult [1].

The skeleton is usually easy to extract and is known to make an efficient and compact representation of a shape, like the human body [39]. The first body skeletonization method to analyze actions was proposed by Fujiyoshi et al. [12]. Their method performs a skeletonization of the body contour to identify walking,

[*] Corresponding author. Tel.: +33 426234445.
E-mail addresses: mathieu.barnachon@gmail.com, mathieu.barnachon@liris.cnrs.fr (M. Barnachon), boufama@uwindsor.ca (B. Boufama).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.06.020

running and gait. Their solution is simple to use and depends only on a single 2D image to extract the skeleton. Although the recognition process of this method is very efficient for simple activities, it suffers from the simple "star" skeletonization problem as well as from visual ambiguities. Ziaeefard and Ebrahimnezhad [47] have proposed an improvement for this method. They have introduced a normalized-polar histogram, obtained from the extracted "star" skeleton, that corresponds to a cumulative skeleton during one action cycle. In particular, they have analyzed different skeletonization methods and proposed an SVM classification technique to recognize actions. Tran et al. [37] have used a different skeletonization method and have achieved better results on the same datasets. However, as they have used polar histograms instead of time-based histogram, the temporal information is lost. Lv and Nevatia [22] have proposed a MoCap-based solution where actions are modeled by a set of virtual key-poses. This is somehow similar to the animation key-poses that represent important poses and/or transitions between sub-parts of actions. This solution is limited by the number of extracted key-poses and by their computational complexity.

Cuntoor et al. [9] have suggested that trajectories contain the most discriminative information that is relevant to human action analysis. Inspired by this observation, Li and Fukui [20] have proposed a trajectory-based solution using Motion Capture data to identify human actions. However, they have only tested their solution on simple cases and not on real human data variations. Using a large database, Han et al. [13] have exploited the skeleton hierarchy to compute trajectories, where actions were represented in a manifold space. As they have used not all but a subset of joints, they needed very large samples in the training set and a high intra-class variation. Therefore, the clustering process of similar actions in their approach was complex and time consuming. Baak et al. [3] and Müller et al. [27] have addressed the problem of action recognition using the idea of Motion Template. They extract patterns from a sequence of animation to recognize actions, transforming the recognition problem into a tractable pattern recognition problem. In [3], a method was proposed to improve MoCap extraction, using a database of priors such as, feet on the ground during the walk, etc.

Recently available, cheap and easy to use, depth sensors have opened new perspectives for solving the problem of action recognition. Raptis et al. [30] have used joint angles as features to recognize dance actions in a game-based application. As mentioned by the authors, their method is limited by the number of actions. In particular, when the number of different classes is large, their error rate increases drastically. With similar input data, Wang et al. [40] have introduced the concept of *actionlet*. They cluster joints and depth neighborhoods in order to be more discriminant. For instance, "drinking from a cup" and "eating a peanut" can be discriminated according to the depth data around the hand. However, the MoCap data has to include depth information, which is not always possible. Related works from the animation community are also relevant to action recognition. Barbič et al. [4] have used the Principal Component Analysis (PCA) to extract a known number of similar behaviors. Beaudoin et al. [5] extracted subsequences of animations, that are similar in the proposed "motion space", then used a graph-based solution to create smooth transitions between animations. Given that these solutions were designed to be animation tools, i.e. to produce smooth transitions between animations, they suffer from the lack of efficient interpretation structures.

Instead of using skeleton, many researchers have used human shape analysis, mostly silhouettes, to address human action recognition. Bobick and Davis [7] have introduced Motion Templates from Motion History Image (MHI), where the recognition problem is turned into a matching problem. Although their system is faster than classical machine learning approaches, it is still time consuming and not flexible enough for extending the database. To address the efficiency of the database, Elgammal et al. [11] have used the "examplar" paradigm with silhouettes. In particular, the Markov model and the "examplar" paradigm lead to a light training database. Although their solution is efficient for adding new actions, it suffers from the view dependency problem, which is inherent to silhouettes. The proposed solution is more appropriate for "simple" gesture recognition than "complex" action recognition. Huang and Trivedi [16] have presented the concept of cylindrical histogram, where multiple views are used to construct a 3D histogram of voxels. Weinland et al. [41] have adapted the 3D histogram process to the examplar paradigm for view-independent learning from multiple views. Boulgoris and Chi [8] have proposed a hybrid solution that uses labelled body parts from silhouettes. Even though their solution is efficient for gait analysis, its use for general action recognition is hindered by its labeling process that has to be done separately. Xiong and Liu [43] have also used a Markov model with silhouettes to recognize mainly simple behaviors. Yilmaz and Shah [46] considers a silhouette as a 2D surface and construct a 3D surface from a sequence of spatio-temporal silhouettes. Then, they extract interest points from the obtained 3D surface, creating something like a trace of an action. Unfortunately, their process is not real-time because it requires an expensive stage of silhouette correspondence for computing the 3D surface. In addition, the obtained lengthy volume is dependent on the silhouette quality, which is prone to errors. Ahmad and Lee [2] have proposed an extension of the MHI where they have used an SVM to cluster actions. As they were using too many parameters in their system, it was difficult to draw a strong conclusion from their results. Tseng et al. [38] have developed a silhouette-based approach, where silhouettes were used as characteristic vectors. Their actions were clustered using a dimension reduction method, then the k-nearest-neighbor algorithm was used on a temporal graph in the recognition stage. Because their solution depends on the quality of the extracted silhouettes, the recognition success might suffer from it.

Many other previous research works on action recognition have used feature points in a spatio-temporal framework. Laptev and Lindeberg [18] extended the Harris and Stephen detector [14] to the spatio-temporal case. Dollar et al. [10] have proposed another spatio-temporal feature detector, especially designed for cyclic motion in actions. They introduced the concept of cuboid, widely followed by others, where each cuboid encodes information about a local neighborhood. As the spatial locations of cuboids were ignored, it has lead to the concept of bag of words. Ryoo [31] has used these bags of words to construct histograms and use them for ongoing activity recognition. Their solution uses 2D image features, instead of our 3D primitives, and is not examplar-based. In particular, their training phase requires more processing, making it difficult to add new actions. One can also consider the work of Scovanner et al. [34] where the SIFT detector was extended to the 3D case of action recognition. Their solution is usually used with an extrusion of spatio-temporal volume from 2D images, a complex process that is also sensitive to the background extraction result.

More recently, many researchers have worked on spatial configurations. Wong et al. [29] have proposed an extension to the pLSA space to model spatial relations [42]. Ryoo and Aggarwal [32] introduced the *Spatio-Temporal Relationship match* (STR match) to consider the spatial information with the temporal one. Yao et al. [45] obtained a non-linear latent space to discriminate between complex activities in a kitchen. Although their solution can be considered efficient, their latent space is complex to compute and need a huge training set to be effective. By contrast, our solution can work with smaller training sets. In particular, our proposed method outperformed their recognition rate for the kitchen scene, as shown in the experimental results.

3. Histogram-based comparison

This section describes our proposed histogram-based method to classify actions from MoCap data. Let $\mathcal{P}$ be the set of all poses, extracted from a video stream. Let $A=(p_0,\dots,p_N)$ be an action consisting of a time-ordered sequence of poses, assuming A starts at time t_0 and ends at time t_N. To keep the notations simple and without loss of generality, let us assume that $t_0=0$. A pose is geometrically represented by a simple human skeleton that consists of a set of 3D joints with hierarchical relations. Note that because of noise perturbations and speed variation of movements, two different actions may contain some identical poses and instances of the same action may be slightly different. To overcome this problem, all poses composing an action are grouped, based on similarity criteria of their appearances, into a set of clusters. Then, each of these clusters is defined by a representative element, denoted by $\tilde{p}$ and called delegate. To quantify the similarity between two poses, p_1 and p_2, we have used the well known Hausdorff distance [15], denoted by D_P hereafter, that provides an elegant way to compare two poses. In order to achieve the clustering mentioned above, we have defined the following $\varepsilon-$equivalence between two poses.

Definition 1. Let D_P be a distance between two poses, the $\varepsilon-$equivalence between p_1 and p_2 is given by

$$p_1 \sim p_2 \;\Leftrightarrow\; D_P(p_1,p_2)\le\varepsilon \tag{1}$$

where $p_1,p_2\in\mathcal{P}^2$.

In our case, a delegate $\tilde{p}$ is the median element of its $\varepsilon-$equivalence cluster of poses, where $\tilde{p}\in\tilde{\mathcal{P}}$, and $\tilde{\mathcal{P}}\subset\mathcal{P}$ is the set of delegates. Using the above definition, we can introduce the following cumulative frequency occurrences of a delegate $\tilde{p}$ from an action $A=(p_0,\dots,p_N)$ of length t_N.

$$f_A^{\Delta T}(\tilde{p})=|\{p_t/t\in\Delta T\wedge p_t\sim\tilde{p}\ \forall p_t\in A\}| \tag{2}$$

$$\Delta T=[t_i,t_j]/t_0\le t_i<t_j\le t_N \tag{3}$$

where $|\cdot|$ is the cardinal of poses.

Note that when $t<t_N$, we are considering a restriction of action A to the time interval $[0,t]$. Such a restriction is useful to us as we are interested in recognizing actions even before they are completed. To do so, we need to evaluate the likelihood over time of the ongoing MoCap data to be one of our previously learned actions. We have defined our own integral histogram of actions that we have used to compute this likelihood for the recognition decision process.

Definition 2. A pose-based integral histogram, $\mathcal{H}$ of action A, is a histogram given by

$$\mathcal{H}^{\Delta T}(A,\tilde{\mathcal{P}})=\{f_A^{\Delta T}(\tilde{p})/\tilde{p}\in\tilde{\mathcal{P}}\} \tag{4}$$

In order to measure the similarity between two histograms, we have used the Bhattacharyya distance [6] x with our pose-based integral histograms. This histogram distance, denoted D_H, between two actions A and B is given by

$$D_H(\mathcal{H}^{\Delta T_A}(A,\mathcal{P}),\mathcal{H}^{\Delta T_B}(B,\mathcal{P}))=\sqrt{1-\sum_{\tilde{p}\in\tilde{\mathcal{P}}}M} \tag{5}$$

where

$$M=\frac{\sqrt{f_A^{\Delta T_A}(\tilde{p})\cdot f_B^{\Delta T_B}(\tilde{p})}}{\sqrt{\sum_{\tilde{p}\in\tilde{\mathcal{P}}}f_A^{\Delta T_A}(\tilde{p})\cdot\sum_{\tilde{p}\in\tilde{\mathcal{P}}}f_B^{\Delta T_B}(\tilde{p})}} \tag{6}$$

Using relation (5), we can introduce the following cost function to evaluate the similarity between two actions A and B.

$$Cost(A,B)=D_H(\mathcal{H}^{T_A}(A,\mathcal{P}),\mathcal{H}^{T_B}(B,\mathcal{P})) \tag{7}$$

where T_A and T_B represent the end times (or lengths) of A and B, respectively.

4. Online recognition

4.1. Piece-wise action comparison

Because integral histograms lack the temporal information about poses, we propose to decompose actions into time-ordered sub-actions. Given an action A, defined by a time-ordered sequence of poses $(p_0,\dots,p_n)$, all possible decompositions of A into sub-actions, of lengths ranging from 1 to n+1, can be defined using the following recursive formulation:

- A has 1 pose: $A=(p_0)$
$$Decomp(A)=\{([p_0])\} \tag{8}$$
- A has 2 poses: $A=(p_0,p_1)$
$$Decomp(A)=\{([p_0][p_1]),([p_0])([p_1])\} \tag{9}$$
- A has 3 poses: $A=(p_0,p_1,p_2)$
$$\begin{aligned}Decomp(A)=\{&([p_0][p_1][p_2]),([p_0][p_1])([p_2]),\\&([p_0])([p_1][p_2]),([p_0])([p_1])([p_2])\}\end{aligned} \tag{10}$$

$\vdots$

- A has n+1 poses: $A=(p_0,\dots,p_n)$
$$Decomp(A)=\bigcup_{s\in Decomp(A\backslash\{p_n\})}\{Concat(s,\{p_n\})\}\cup(s,\{p_n\}) \tag{11}$$

where

$$\begin{aligned}&Concat((p_a,\dots,p_b)^\star(p_i,\dots,p_{n-1}),\{[p_n]\})\\&\quad=\{(p_a,\dots,p_b)^\star(p_i\cdots p_{n-1}p_n)\}\end{aligned} \tag{12}$$

where $0\le a\le b<i\le n-1$, and for $A=(p_0,\dots p_{n-1},p_n)$ and $(p_i)^\star$ means that the sequence (p_i) could be repeated 0 to N times, like in regular expressions.

$$A\backslash\{p_n\}=(p_0,\dots,p_{n-1}) \tag{13}$$

When comparing an action A to another action B, we have to find the optimal sub-action decompositions of A and B that yield the minimum score, given by Eq. (7). To this purpose, we build one histogram for each sub-action, yielding a time-ordered sequence of histograms for the whole action. For instance, a 4-pose-based action A decomposed into 3 subactions can be represented by three ordered histograms as follows:

$$A=\underbrace{([p_0])}_{h_0},\underbrace{([p_1][p_2])}_{h_1},\underbrace{([p_3])}_{h_2}\ \Rightarrow\ A\text{ is represented by }(h_0,h_1,h_2) \tag{14}$$

Therefore, an action decomposition can be considered as time series of sub-integral histograms, represented by a vector $(h_0,\dots,h_N)$ of length not greater than N.

To effectively compute the optimal sub-actions decomposition, we have used the dynamic programming paradigm, which is the best choice in this case. In particular, to compare action A to action B, all possible decompositions of A and B are evaluated and the best score $(cost^\star)$ is selected. This is done through the following

Please cite this article as: M. Barnachon, et al., Ongoing human action recognition with motion capture, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.06.020

recursive relations.

$$\begin{aligned} Cost^{\star}(h_0^A, h_0^B) &= Cost(h_0^A, h_0^B) Cost^{\star}(h_i^A, h_j^B) \\ &= Cost(h_i^A, h_j^B) + \min\{Cost(h_{i-1}^A, h_j^B), \\ &\quad Cost(h_i^A, h_{j-1}^B), Cost(h_{i-1}^A, h_{j-1}^B)\} \end{aligned} \tag{15}$$

where, $(h_0^A, \ldots, h_N^A)$ and $(h_0^B, \ldots, h_M^B)$ refer to A and B, respectively.

4.2. Multi-hypothesis

Using a single instance per action as a training set will not yield a good recognition rate. This is because of the occurring natural variations in human activities, due to different body proportions and movement styles. In particular, two instances of the same action can be slightly different with respect to poses. To overcome this problem, the training set of each action should consist of multiple instances of the same action. These multiple instances are translated into multiple histograms. Rather than combining an action's multiple histograms into a single one, which is a challenging task, we propose to use several histograms to represent an action. These different histograms, representing a single action, are referred to as hypotheses. Note however, that if we are given n instances of the same action as a training set, we will end up with fewer than n hypotheses. This is because instances found to be very similar are clustered into a single hypothesis. In practice, we have achieved this using the *K-medoids* algorithm, which has many similarities with the *K-means* algorithm. These two algorithms differ from each other by the choice of the median element. In the *K-means* case, the median is the barycentric center of the cluster, whereas, for the *K-medoids* the median is an element of the cluster, the closest one to the barycentric center of the cluster.

Algorithm 1. *K*-Medoids: finds the best partition of a set into *K* groups and returns the closest element to each center.

```
Input: Number of clusters k and set of all poses $\mathcal{P}$ of a dataset
$\{Medoids\} \leftarrow$ initial k delegates from $\mathcal{P}$, defining k clusters ;
$minDistance \leftarrow distance\ of\ \{Medoids\}\ to\ their\ clusters$ ;
$\{bestMedoids\} \leftarrow \{Medoids\}$ ;
foreach $m \in \{Medoids\}$ do
    foreach $v \in \mathcal{P} \setminus \{Medoids\}$ do
        $\{Medoids\} \leftarrow \{Medoids\} \setminus \{m\} \bigcup \{v\}$ ;
        $distance \leftarrow distance\ of\ \{Medoids\}\ to\ their\ clusters$ ;
        if $distance < minDistance$ then
            $minDistance \leftarrow distance$ ;
            $\{bestMedoids\} \leftarrow \{Medoids\}$ ;
        end
    end
end
```

The distance function used in the above algorithm is the one defined in Eq. (7) and each medoid is a histogram representing a single hypothesis.

In other words, given that "very similar" histograms are not better than a single one, we only keep those histograms that represent "different" instances of the same action. The set of these "different" histograms, call it $\mathcal{H}^A$, represents the multiple hypotheses for the same action in the recognition stage. Hence, our recognition score is computed using the following equation:

$$Cost_{multi}(A, B) = \min_{h^A \in \mathcal{H}^A} \{Cost^{\star}(h^A, h^B)\} \tag{16}$$

where $\mathcal{H}^A$ is the set of multiple hypotheses of action A and h^B is the histogram of the ongoing action to be recognized.

5. Results

We have tested our method on four different datasets, HDM, CMU, TUM and MSR Action3D. Both HDM and CMU are game-oriented where actions were created for a computer animation purpose. These two datasets consist mostly of well defined short actions, making it easy to evaluate the recognition performance. Actions in the TUM and MSR Action3D datasets are more realistic as they consist of real human activities in a complex and natural environment. These two datasets have allowed us to test the robustness of the proposed method and to compare it with others.

5.1. Game-oriented dataset

We have tested our system on the HDM dataset of actions from [26]. This dataset consists of 130 classes, obtained from 2337 actions (cuts of longer capture sets), made by 5 different actors. We have considered the two scenarios, single-hypothesis and multi-hypothesis. In the former case, we have randomly selected one action instance from each class (random execution, random actor) and used it as a training set. Whereas in the multi-hypothesis case, we have randomly selected a few action instances from each class and have kept only three instances to represent each action. We have also constructed another dataset, consisting of 9 classes out of 53 actions, from the very large and complex CMU dataset [24] (see Table 1).

Compared to the results published in [28], where a similar subset of CMU dataset has been used, our solution has performed much better. Their recognition rate was around 75%, whereas ours are 86.63% and 90.92% for the single-hypothesis and multi-hypothesis, respectively.

Fig. 1(a) and (b) presents the results of the single hypothesis as confusion matrices computed with $\varepsilon = 1.0$. These two figures show in particular that our method highly discriminates between different actions. It also highlights similar actions, such as "Boxing", "Drink" and "Eat", as the three of them involve hand activities. The multi-hypothesis solution, shown in Fig. 1(c) and (d), is clearly more discriminant, as it allows more intra-class variations. Note that for the multi-hypothesis solution, the 130 initial classes were manually reduced to 33 different classes, to be semantically consistent with usual approaches dealing with 2D actions. The resulting dataset is actually more challenging for examplar-based approaches. Indeed, with 33 classes there is more intra-class variation than with the original 130 classes. Because we have used only 3 instances per class for training, any increase in the intra-class variation will add extra challenge to our recognition method. In fact, the task would be easier if we simply add more training samples and keep all the 130 initial classes. The difficulty arises from the additional intra-class variations (e.g., number of steps or left/right first step used for walk), a more challenging context for any examplar-based recognition method. On the other hand, classical machine learning approaches use much larger training sets, making them more successful when the number of classes is small (Fig. 2).

More quantitative scores are shown in Table 4, where the proposed method clearly yields excellent results when compared to others. Although the single-hypothesis recognition rate is

Table 1
The training and test sets we have used from the CMU MoCap dataset.

Name	Training sets	Sequences used as testing sets
Walk	02_01	02_02;03_01;05_01;07_01;08_01
Run	02_03	09_01;17_01;35_22;77_10;141_02
Punching	143_23	02_05;111_19;113_13
Boxing	13_17	13_18;14_01;14_02;14_03;14_13;80_10
Jump	13_32	13_39;13_40;16_01;16_03;118_02
Shake hands	18_01	18_02;19_01;19_02;79_06;141_23;80_73
Laugh	13_14	13_15;13_16;14_17;14_18;14_19
Drink	13_09	14_04;14_37;23_13;79_38;79_40
Eat	79_12	79_15;79_42;80_11;80_33

Please cite this article as: M. Barnachon, et al., Ongoing human action recognition with motion capture, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.06.020

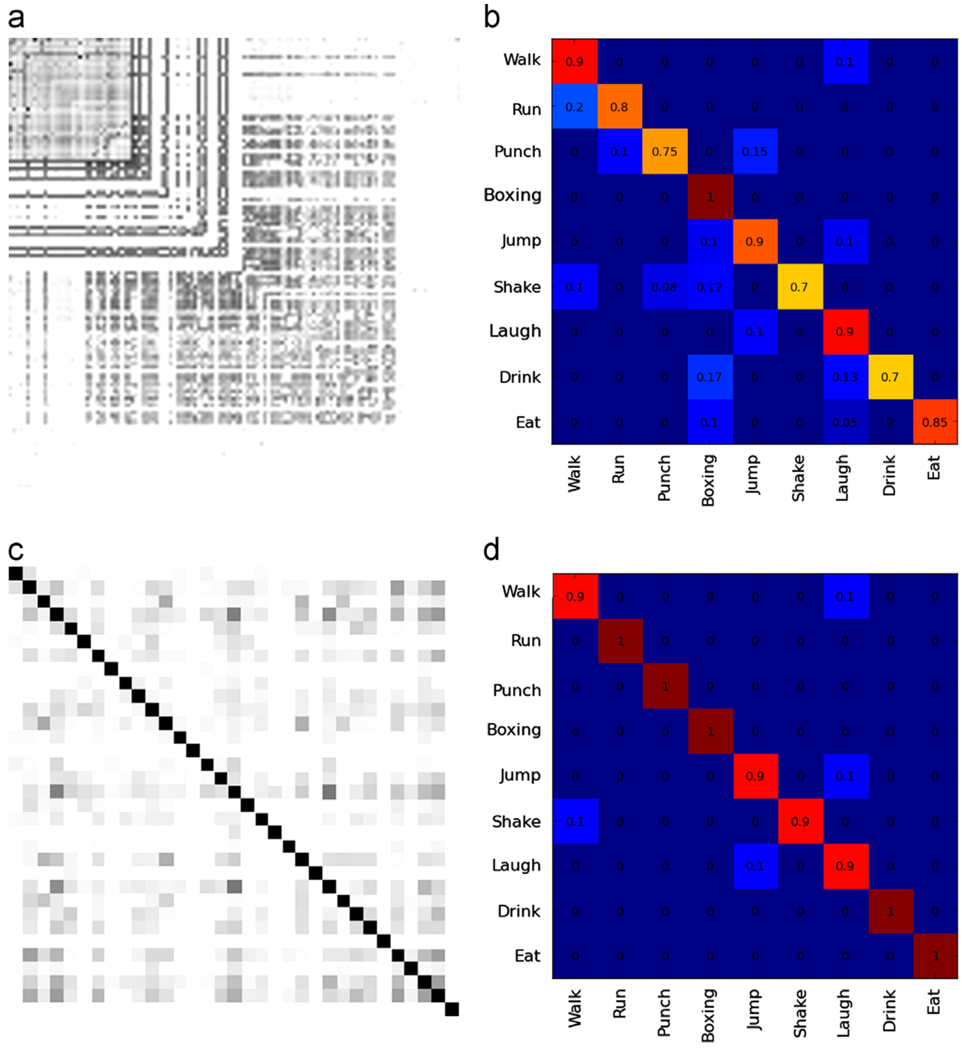

Fig. 1. Confusion matrices for HDM and CMU datasets, where image colors are in logarithmic to enforce contrast. The multi-hypothesis results in (c) and (d) are clearly superior than the single-hypothesis results in (a) and (b). (a) HDM single (130 classes), (b) CMU single, (c) HDM multi (33 classes) and (d) CMU multi. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

outperformed by two out of three previous works, our method is in a disadvantageous situation with its one-vs-one strategy. However, our fully automatic multi-hypothesis solution has outperformed all three methods in a similar context. For example, [28] has an average recognition rate of 80% whereas our multi-hypothesis achieved a rate of 96.67% on the same dataset. Note also that in [28], keyframes, used in the queries, were manually selected.

5.2. *Activity dataset*

5.2.1. *TUM*

In order to compare our solution with activity recognition methods, we have used the TUM dataset [36] that consists of 20 sequences of people setting the table. Like in [44], we have applied two different strategies: using the set of sequences {0–2, 0–4, 0–6, 0–8, 0–10, 0–11, 1–6} as a test set, while the others are used for training. To be consistent with others [17,36,44], we have used 10 different classes, where “standing” and “walking” were separated. As shown in Table 2, the obtained results are very good compared to the 81.5% recognition rate reported in [44], which uses 2D features and MoCap information from the TUM dataset. They are also much better than the 67% of [17], where only 2D features were used. Finally, our results outperformed the recognition rate of 62.77% reported in [36], even with their extra sensors used for doors.

We have also carried out a number of experiments – known as *one-vs-one* or *one-shot learning* – where, we used the first instance of each action from the training set for learning. We have compared our results to three classical machine learning algorithms: a kernel-based SVM with Radial Basis Function, a k-Nearest Neighbors and, a Tree classification. The training set for these methods is the same as in [17,36,44]. An example of the one-vs-one learning is given in Fig. 3 showing the recognition score for the full sequence 0–2, using the actions of sequence 0–12. Although there is less ambiguity for this dataset, some actions were not recognized due to the complexity of their execution and the lack of intra-class variation information.

We have summarized all the results in Table 2. Our solution slightly outperforms the three approaches SVM, kNN and Tree, for the one-vs-one learning, and significantly outperforms six state-of-the-arts methods (SVM, kNN, Tree and [17,36,44]) for the multi-hypotheses learning.

5.2.2. *MSR Action3D*

The MSR Action3D is a newer dataset, created by Microsoft Research in 2010, that aims at providing 3D data for action recognition. This dataset, first used in [19], is made up of twenty actions performed by ten actors, where each action has three

Please cite this article as: M. Barnachon, et al., Ongoing human action recognition with motion capture, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.06.020

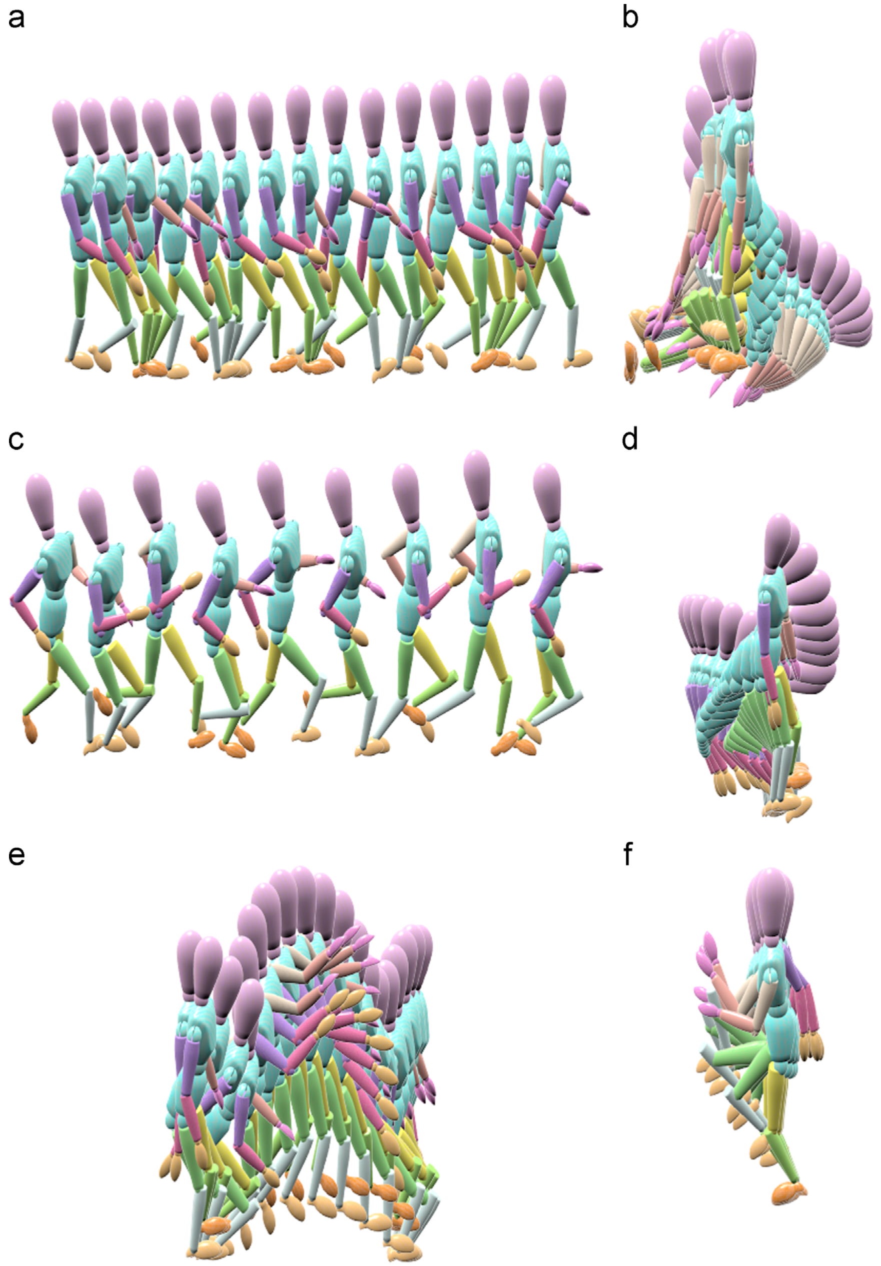

Fig. 2. Samples images of data used in the training and recognition. (a) CMU Walk, (b) HDM Lie Down, (c) CMU Run, (d) HDM Sit Chair, (e) CMU Jump and (f) HDM Elbow to Knee.

Table 2
Comparisons of different solutions on the TUM dataset. The "–" entry means no result was reported by the other methods for the one-vs-one strategy.

	TUM (one-vs-one) (%)	TUM (full) (%)
SVM (RBF+K)	49.52	54.67
kNN	51.41	71.34
Tree	26.67	75.28
[36]	–	62.77
[17]	–	67
[44]	–	81.50
Proposed method	**56.82**	**92.56**

instances. The dataset consists of the following 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*, *side boxing*, *bend*, *forward kick*, *side kick*, *jogging*, *tennis swing*, *tennis serve*, *golf swing*, *pick up & throw*. Actors were requested to use their right arm or leg, when only one arm or one leg is involved in the action. The depth maps of these actions were obtained with Microsoft Kinect sensor and the skeletons were extracted using the method in [35] (see Fig. 4 for examples of actions). This dataset is a challenging one due to the noise in the extracted skeletons. In particular, the obtained skeletons have more noise than the ones in the TUM dataset, and even more than the ones in HDM and CMU.

Using these datasets, we have compared our approach to five state-of-the-art methods [19,21,23,25,40]. Except for [19], where depth maps were used as inputs, the other four methods [21,23,25,40] as well as ours use skeleton inputs, extracted from video streams. with an action graph to model the dynamic of actions. Note that we have used the test results on the MSR Action3D dataset of [21,23,25] that were reported in [40]. Table 3 summarizes this comparison and shows in particular, that our multi-hypothesis approach clearly outperforms the other

Please cite this article as: M. Barnachon, et al., Ongoing human action recognition with motion capture, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.06.020

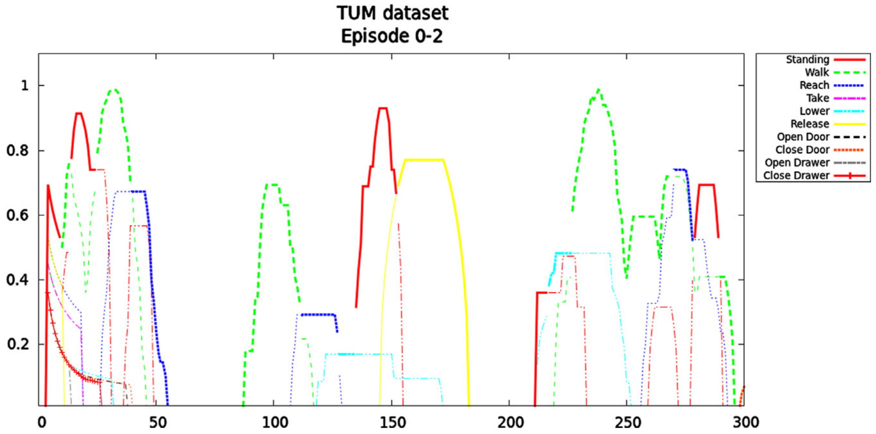

Fig. 3. Evolving recognition score for the sequence 0–2 in one-vs-one learning with sequence 0–12 being the training one. The figure presents the accuracy computed over action's frames.

Fig. 4. Sample frames of the MSR Action3D dataset. Courtesy of Wang et al. [40].

Table 3
Comparisons of different solutions on the MSR Action3D dataset.

Methods	Accuracy (%)	Used primitives
Recurrent Neural Network [23]	42.50	MoCap
Dynamic Temporal Warping [25]	54	MoCap
Hidden Markov Model [21]	63	MoCap
Action Graph on Bag of 3D Points [19]	74.70	Depth Map
Mining of Actionlet ensemble [40]	88.2	MoCap
Our (*one-vs-one*)	**63.92**	MoCap
Our (full)	**90.56**	MoCap

Table 4
Recognition rates on all datasets compared with the best results from the other state-of-the-art methods. Entry "–" means no result was reported by that method.

Dataset	Accuracy	
	Our (%)	Best (%)
HDM (*one-vs-one*)	**67.89**	–
HDM (multi-hypothesis)	**96.67**	80 [28]
CMU (*one-vs-one*)	**86.63**	–
CMU (multi-hypothesis)	**90.92**	75 [28]
TUM (*one-vs-one*)	**56.82**	51.41 (kNN)
TUM (multi-hypothesis)	**92.56**	81.50 [44]
MSR Action3D (*one-vs-one*)	**63.92**	–
MSR Action3D (multi-hypothesis)	**90.56**	88.20 [40]

5 methods. Even when using the one-vs-one strategy, our method is still competitive.

We have also summarized all results for all datasets in Table 4 to provide an overview of our method performance in comparison with others.

These results suggest the followings. When MoCap data (3D skeleton) is available, our method is best suited for action recognition. Furthermore, as our method is examplar-based, it is preferable when we have small training sets and/or when online and fast training is desirable.

5.3. *Early recognition*

As our solution has the potential to recognize actions before their completion, we present here the obtained results for early recognition. Fig. 5 shows the obtained recognition accuracies for all datasets, with the usual learning and the one-vs-one learning, where the progress of actions ranges from 50% to 100% of their completion. For the case of TUM dataset, where actions are complex, the recognition rate does not significantly increase after 50% of the action progress. This is mainly because most of the discriminant information is contained in the beginning of the action for this dataset. However, for the CMU dataset, as the actions were longer and very different, the accuracy increases with the action progress to confirm the recognition over time. In the HDM dataset, where actions are short, the recognition score at 50% of the action's progress is very high. This is a consequence of the matching process, performed by a dynamic programming approach, that almost perfectly match short actions, leading to such an excellent score.

5.4. *Discussion on parameters setting*

Similar to other action recognition methods, our solution depends on two important parameters, the between-pose distance threshold ε of Eq. (1) and the number of hypotheses to be used for an action (described in Section 4.2).

Please cite this article as: M. Barnachon, et al., Ongoing human action recognition with motion capture, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.06.020

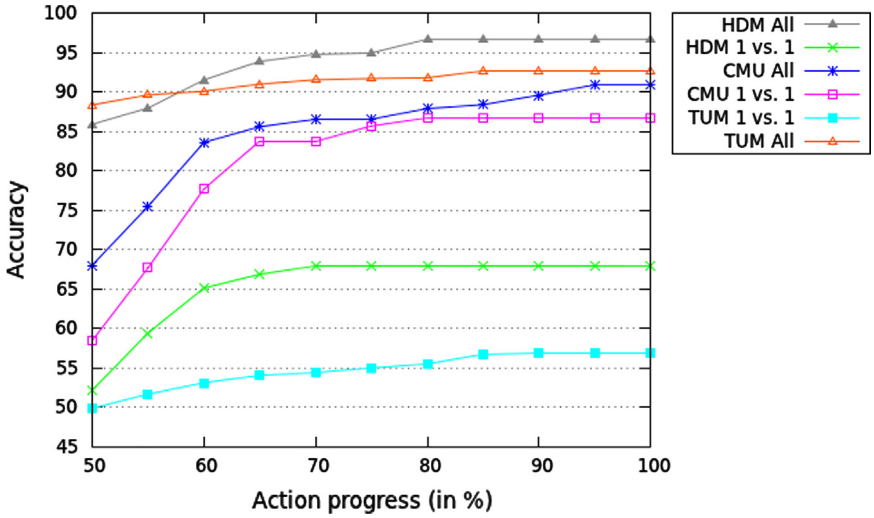

Fig. 5. Early action recognition results with the action progress ranging from 50% to 100% and a progress step of 5%.

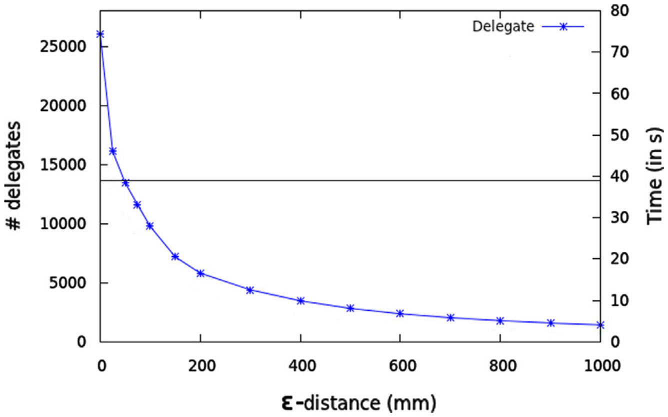

Fig. 6. The effect of ε−distance on the number of delegates and on the recognition time. The graph suggests that if real-time is sought, we should choose ε−distance above 50 mm.

The first parameter ε is a distance threshold that determines the number of representative poses (delegates). This parameter directly affects the computation time of the recognition process as suggested by Fig. 6. The latter shows that our method is always real time, except when ε takes values that are less than 1 mm. However, 1 mm is too low for a distance threshold and yields too many delegates, 26046 for the TUM training set for example. Furthermore, the MoCap data metric accuracy is around a few centimeters [36]. Hence, we have set ε to 20 mm for all our tests, which is well above the data errors and ensures that our recognition process is always real-time. For example, with $\varepsilon = 20$ mm, we have obtained 5799 delegates for the TUM dataset. When taking a particular action from the latter, e.g., the sequence 0–2, made of 957 poses, the recognition process took 12.0756 s, which is less than 1/3 of the sequence length of 39 s.

The second important parameter for our method is the number of hypotheses per action. The distance threshold value, used to decide whether two histograms are different, affects the number of hypotheses to be kept for an action. We have investigated the effect of this parameter for the case of the action “walk”, chosen because it is a very short and variable one, with a high likelihood of over segmentation with respect to hypotheses. We have found that when this distance threshold is below 0.5, the number of hypotheses remains high and constant. On the other hand, when this threshold passes 0.5, the number of hypotheses decreases rapidly. Hence, this threshold could be set between 0.5 and 0.9, depending on the desired level of granularity. In our case, we have used 0.5 for this parameter in all our tests, to keep most of the intra-class variations.

6. Conclusion

This paper proposed a new technique for ongoing human action recognition. We have provided a new histogram-based formulation for recognizing streams of poses from Motion Capture data. The use of histograms has allowed our solution to be more efficient, in terms of required space and computational time, than most existing methods. Furthermore, because histogram structures are flexible, a new action can be added in a straightforward way, by computing and updating its histogram of poses while the action is being performed. In order to overcome the lack of temporal information in histograms, we have proposed an extension of the classical histograms to the integral histograms. Hence, actions can be sliced into sub-actions, represented by sub-histograms, that will be compared for recognition in an incremental way. We have used the examplar paradigm as a learning approach that has made it possible to use very small training sets. Hence, using a few training instances for each action, our proposed solution was able to recognize actions in real-time. The combination of integral histograms and examplars makes it easy to extend the training dataset, even during the recognition process. That is, an unknown action could be easily added to the training set during the recognition process to extend the dataset of actions with new ones. Furthermore, by using integral histograms, we are able to recognize actions even before they are completed. Such ongoing recognition method opens up new possibilities for action recognition, especially in the new Human–Computer Interaction applications, where the user has the freedom to extend its application with new gestures at runtime. The extensive tests made on different datasets have validated our approach for classical human actions, such as walk, run, jump, etc. in simple settings. We have also tested our solution in more challenging settings, with the TUM kitchen-based dataset and Microsoft MSR Action3D, in order to show the high accuracy and discriminative efficiency of the method. Our solution was flexible enough to cope with intra action variations, due to body proportion differences and/or variations in the speed of actions. This flexibility was possible thanks to the use of dynamic alignment of sub-histograms. The obtained results have shown that our approach yielded the best recognition rate when compared with many well known state-of-the-art methods on four different datasets. In particular, when 3D MoCap data are available, our multi-hypothesis examplar-based method is preferable over others, as it is the most accurate and most time-efficient one. Our future work will aim at extending the proposed method to use a more semantic approach of multi-users, reinforcing the recognition of actions by a cross-validation of each user's recognized actions.

Conflict of interest

None declared.

Acknowledgment

This work has been supported by ANR (french National Research Agency) and CNSA (french National Solidarity Fund for Autonomy).

References

[1] Jake Aggarwal, Michael Ryoo, Human activity analysis, ACM Computing Survey 43 (2011) 1–43.

[2] Mohiuddin Ahmad, Seong-Whan Lee, Variable silhouette energy image representations for recognizing human actions, Image and Vision Computing 28 (5) (2010) 814–824.

[3] Andreas Baak, Bodo Rosenhahn, Meinard Müller, Hans-Peter Seidel, Stabilizing motion tracking using retrieved motion priors, in: IEEE 12th International Conference on Computer Vision, September 2009, pp. 1428–1435.

[4] Jernej Barbič, Alla Safonova, Jia-Yu Pan, Christos Faloutsos, Jessica K. Hodgins, Nancy S. Pollard, Segmenting motion capture data into distinct behaviors, in: GI '04: Proceedings of Graphics Interface 2004, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2004. Canadian Human–Computer Communications Society, pp. 185–194.

[5] Philippe Beaudoin, Stelian Coros, Michiel van de Panne, Pierre Poulin, Motion-motif graphs, in: SCA '08: Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Eurographics Association. Aire-la-Ville, Switzerland, Switzerland, 2008, pp. 117–126.

[6] A. Bhattacharyya, On a measure of divergence between two statistical populations defined by their probability distributions, Bulletin of the Calcutta Mathematical Society 35 (1943) 99–109.

[7] Aaron F. Bobick, James W. Davis, The recognition of human movement using temporal templates, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (3) (2001) 257–267.

[8] Nikolaos V. Boulgoris, Zhiwei X. Chi, Human gait recognition based on matching of body components, Pattern Recognition, 2006.

[9] N.P. Cuntoor, B. Yegnanarayana, R. Chellappa, Activity modeling using event probability sequences, IEEE Transactions on Image Processing 17 (April (4)) (2008) 594–607.

[10] P. Dollar, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: Proceedings of the 14th International Conference on Computer Communications and Networks, 2005, pp. 65–72.

[11] Ahmed Elgammal, Vinay Shet, Yaser Yacoob, Larry S. Davis, Learning dynamics for exemplar-based gesture recognition, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, 2003, p. 571.

[12] Hironobu Fujiyoshi, Alan J. Lipton, Real-time human motion analysis by image skeletonization, in: IEEE Workshop on Applications of Computer Vision, vol. 0, 1998, p. 15.

[13] Lei Han, Xinxiao Wu, Wei Liang, Guangming Hou, Yunde Jia, Discriminative human action recognition in the learned hierarchical manifold space, Image and Vision Computing, 2010, 836–849.

[14] C. Harris, M. Stephens, A combined corner and edge detection, in: Proceedings of the Fourth Alvey Vision Conference, 1988, pp. 147–151.

[15] Felix Hausdorff, Grundzüge der Mengeniehre, Von Veit, Leipzig, 1914.

[16] Kohsia S. Huang, Mohan M. Trivedi, 3d shape context based gesture analysis integrated with tracking using omni video array, in: CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)—Workshops, IEEE Computer Society, 2005, Washington, DC, USA, p. 80.

[17] B. Krausz, C. Bauckhage, Action recognition in videos using nonnegative tensor factorization, in: 2010 20th International Conference on Pattern Recognition (ICPR), 2010, pp. 1763–1766.

[18] Ivan Laptev, Tony Lindeberg, Space-time interest points, in: ICCV, 2003, pp. 432–439.

[19] W. Li, Z. Zhang, Z. Liu, Action recognition based on a bag of 3d points, in: Human Communicative Behavior Analysis Workshop (in conjunction with CVPR), 2010.

[20] Xi. Li, .Fukui Kazuhiro, View invariant human action recognition based on factorization and hmms, IEICE – Transactions on Information and Systems E91-D (7) (2008) 1848–1854.

[21] F. Lv, R. Nevatia, Recognition and segmentation of 3d human action using HMM and multi-class adaboost, in: ECCV, 2006.

[22] Fengjun Lv, Ramakant Nevatia, Single view human action recognition using key pose matching and viterbi path searching, in: CVPR, 2007.

[23] J. Martens, I. Sutkever, Learning recurrent neural networks with hessian-free optimization, in: ICML, 2011.

[24] MoCap CMU, The data used in this project was obtained from mocap.cs.cmu.edu. the database was created with funding from nsf eia-0196217. 〈http://mocap.cs.cmu.edu/〉.

[25] M. Muller, T. Röder, Motion templates for automatic classification and retrieval of motion capture data, in: ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2006.

[26] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, A. Weber, Documentation Mocap Database HDM05, Technical Report CG-2007-2, Universität Bonn, June 2007.

[27] Meinard Müller, Andreas Baak, Hans-Peter Seidel, Efficient and robust annotation of motion capture data. in: SCA '09: Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, ACM, New York, NY, USA, 2009, pp. 17–26.

[28] Meinard Müller, Andreas Baak, Hans-Peter Seidel, Efficient and robust annotation of motion capture data, in: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2009, pp. 17–26.

[29] Juan Carlos Niebles, Hongcheng Wang, Li Fei-fei, Unsupervised learning of human action categories using spatial–temporal words, in: Proceedings of the BMVC, 2006.

[30] Michalis Raptis, Darko Kirovski, Hugues Hoppe, Real-time classification of dance gestures from skeleton animation, in: Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation – SCA '11, ACM Press, 2011, p. 147.

[31] Michael Ryoo, Human activity prediction: early recognition of ongoing activities from streaming videos, in: International Conference on Computer Vision, 2007.

[32] Michael S. Ryoo, Jake K. Aggarwal, Spatio-temporal relationship match: video structure comparison for recognition of complex human activities, in: ICCV, 2009, pp. 1593–1600.

[33] Hiroaki Sakoe, Seibi Chiba, Dynamic programming algorithm optimization for spoken word recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing 26 (1978) 43–49.

[34] Paul Scovanner, Saad Ali, Mubarak Shah, A 3-dimensional sift descriptor and its application to action recognition, in: Proceedings of the 15th International Conference on Multimedia, 2007, pp. 357–360.

[35] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, Andrew Blake, Real-time human pose recognition in parts from a single depth image, in: CVPR, 2011.

[36] Moritz Tenorth, Jan Bandouch, Michael Beetz, The TUM kitchen data set of everyday manipulation activities for Motion Tracking and Action Recognition, in: IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS), in conjunction with ICCV2009, 2009.

[37] K.N. Tran, I.A. Kakadiaris, S.K. Shah, Part-based motion descriptor image for human action recognition, Pattern Recognition, 2012.

[38] Chien-Chung Tseng, Ju-Chin Chen, Ching-Hsien Fang, Jenn-Jier James Lien, Human action recognition based on graph-embedded spatio-temporal subspace, Pattern Recognition, 2012.

[39] R.M. Udrea, N. Vizireanu, Iterative generalization of morphological skeleton, Journal of Electronic Imaging, 2007.

[40] Jiang Wang, Zicheng Liu, Ying Wu, Junsong Yuan. Mining actionlet ensemble for action recognition with depth cameras, in: IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[41] Daniel Weinland, Edmond Boyer, Remi Ronfard, Action recognition from arbitrary views using 3d exemplars, in: Proceedings of the International Conference on Computer Vision, Rio de Janeiro, Brazil, 2007, pp. 1–7.

[42] Shu-Fai Wong, Tae-Kyun Kim, Roberto Cipolla, Learning motion categories using both semantic and structural information, in: CVPR, 2007.

[43] Jing Xiong, ZhiJing Liu, Human motion recognition based on hidden Markov models, in: Advances in Computation and Intelligence, 2007, pp. 464–471.

[44] Angela Yao, Juergen Gall, Gabriele Fanelli, Luc Van Gool, Does human action recognition benefit from pose estimation?, in: Proceedings of the British Machine Vision Conference, 2011, pp. 67.1–67.11.

[45] Angela Yao, Juergen Gall, Luc Van Gool, Raquel Urtasun, Learning probabilistic non-linear latent variable models for tracking complex activities, in: Neural Information Processing Systems (NIPS), 2011.

[46] Alper Yilmaz, Mubarak Shah, A differential geometric approach to representing the human actions, Computer Vision and Image Understanding 109 (3) (2008) 335–351.

[47] Maryam Ziaeefard, Hossein Ebrahimnezhad, Hierarchical human action recognition by normalized-polar histogram, in: 2010 20th International Conference on Pattern Recognition, 2010, pp. 3720–3723.

Mathieu Barnachon received his Ph.D. in computer science from the University of Lyon 1 in 2013. He is currently research assistant at the Claude Bernard University of Lyon, France. His received his M.Sc. in 2008 from the University of Lyon 1. His research interests include human action recognition and their application to Human computer interaction.

Saïda Bouakaz received her Ph.D. from Joseph Fourier University of Grenoble, France. She is currently a professor in the Computer Science department, at the Claude Bernard University of Lyon, France. Her current research concerns computer vision and graphics including motion capture and analysis, gesture recognition, computer animation of characters.

Boubakeur Boufama received his M.Sc. degree in Computer Science from the University of Nancy I, France, in 1991. He received his Ph.D., also in Computer Science, from the Institut National Polytechnique de Grenoble, France, in 1994. He is currently a full professor at the School of Computer Science of the University of Windsor, Canada. He has published over 70 papers in various journals and proceedings related to computer vision and graphics. His research interests include computer vision, image processing and computer graphics.

Erwan Guillou received his Ph.D. in computer science from the University of Rennes in 2000. He is currently associate professor in the Computer Science department, at the Claude Bernard University of Lyon, France. His research interests include computer vision, 3D reconstruction, Markerless Motion Capture and Human action recognition.

Please cite this article as: M. Barnachon, et al., Ongoing human action recognition with motion capture, Pattern Recognition (2013), http://dx.doi.org/10.1016/j.patcog.2013.06.020