Online RGB-D Gesture Recognition with Extreme Learning Machines

Xi Chen and Markus Koskela
Department of Information and Computer Science
Aalto University School of Science
PO Box 15400, FI-00076 AALTO, Finland
xi.chen@aalto.fi, markus.koskela@aalto.fi

ABSTRACT

Gesture recognition is needed in many applications such as human-computer interaction and sign language recognition. The challenges of building an actual recognition system do not lie only in reaching an acceptable recognition accuracy but also with requirements for fast online processing. In this paper, we propose a method for online gesture recognition using RGB-D data from a Kinect sensor. Frame-level features are extracted from RGB frames and the skeletal model obtained from the depth data, and then classified by multiple extreme learning machines. The outputs from the classifiers are aggregated to provide the final classification results for the gestures. We test our method on the ChaLearn multi-modal gesture challenge data. The results of the experiments demonstrate that the method can perform effective multi-class gesture recognition in real-time.

Categories and Subject Descriptors

I.4.7 [**IMAGE PROCESSING AND COMPUTER VISION**]: Feature representation; I.4.7 [**IMAGE PROCESSING AND COMPUTER VISION**]: Applications; I.5.4 [**PATTERN RECOGNITION**]: Computer vision—*gesture recognition*

General Terms

Algorithms, Experimentation, Performance

Keywords

Online gesture recognition; extreme learning machine; RGB-D; skeleton model; HOG

1. INTRODUCTION

Human action and gesture recognition has been a popular research topic for the last few decades [18, 13, 19]. The general aim in the field is to provide automated analysis of various kinds of human activities. Action and gesture recognition has many practical applications in real life, such as in surveillance, human–computer interfaces, gaming, medical rehabilitation, and analysis of sign language. Previously most of the developed approaches were based on RGB video data [15, 14, 26]. Another important data source are the motion capture (mocap) systems which capture human motion with high frequency and accuracy by calculating the joints' coordinates and the angular information of the human skeleton using a system setup consisting of multiple calibrated cameras in a dedicated space [16]. Motion capture is often used in fields such as filmmaking, computer animation, sports science, and game development. The skeletal data generated by mocap systems is also used for action recognition to facilitate the data retrieval for reuse due to the expensiveness of mocap data generation [16].

On the other hand, the existing commodity RGB-D (RGB and depth) sensors, such as the Microsoft Kinect, provide depth information along with the standard RGB video and are now widely used e.g. in gaming, human–computer interfaces, and robotics due to their portability and low cost. The depth modality provides a lot of extra information compared to the original RGB data, which gives new perspectives for researchers to solve many traditional problems in computer vision, such as object recognition, segmentation, and gesture recognition. Several algorithms have been developed to extract the human skeleton from the depth images in real-time [24, 30]. Essentially, these algorithms classify a large 3D point cloud into about a dozen human skeleton joint coordinates and thus provide data analogous to mocap. This enables the classification methodology developed for mocap skeletons to be applied for RGB-D data as well.

In this paper, we use multi-modal data obtained from a Kinect sensor for online gesture recognition. The aim is to develop a robust camera-based (RGB and depth) method

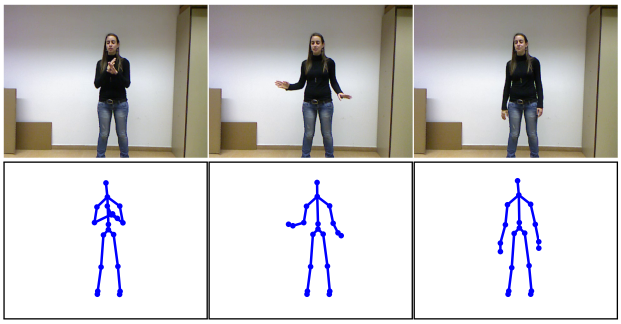


Figure 1: Frames from an example gesture from the ChaLearn dataset and associated skeleton models.

ICMI'13, December 9–13, 2013, Sydney, Australia
Copyright 2013 ACM 978-1-4503-2129-7/13/12 ...$15.00.
http://dx.doi.org/10.1145/2522848.2532591.

that can recognize several gestures in real-time using a standard desktop or laptop computer. The method is based on the skeleton model extracted from the depth images [24] (see Figure 1 for an example) and a method for full-body action recognition we have previously applied to both mocap and RGB-D data [3]. The method is extended in this paper by extracting the hand regions from the RGB data and extracting histogram of oriented gradients (HOG) [5] features from them. As the classifier, we use Extreme Learning Machines (ELM) [9], which can provide high accuracy and, at the same time, both classification and the training of the models are very fast compared to many other non-linear classification methods. The outputs from each modality are fused together to get the final classification results. We test our method on data from the ChaLearn Multi-modal Gesture Recognition Challenge 2013 [1]. Our setup here differs from that of the common challenge as we consider here gesture recognition only, i.e. we assume that the start and end points of the gestures are known, and we do not use the audio modality.

2. RELATED WORK

Action and gesture recognition have been researched for several decades based on different data sources, with videos being the most traditional data source. In [15], low level visual features are extracted from videos and the concept of sub-actions during complex human actions are utilized. A dynamic Bayesian model is applied based on a language-motivated approach. In [14] videos of different actions are considered as third order tensors and imposed on a product manifold. A regression model based on latent geometry is then used for action recognition. This method is applied in the ChaLearn one-shot learning gesture challenge [1]. In [26], hand gestures are recognized in continuous video streams using a dynamic Bayesian network. The method employs skin models for hands and for face detection, and each gesture is associated with a certain gesture model.

Data from a human skeletal model is also often used for gesture recognition. In [4], the angles from the vertical axis of the hip-center to the rest of the joints are measured and combined as the feature vector of each frame. The feature vectors are clustered by a Gaussian mixture model, and the motion streams are segmented and recognized by a threshold model with a conditional random field (CRF). In [31], the joints' features including static posture, motion, and offset are classified by a Naive-Bayes nearest-neighbor classifier. In [29], the histogram of 3D joints' locations is used to represent the posture. The postures are classified into posture-visual words and modeled by a discrete hidden Markov model (HMM). In [23], dance gestures are classified from Kinect skeletal data. An angular representation of the skeleton is extracted for each frame, and a cascaded correlation-based max-likelihood multivariate classifier is used for the classification, along with a distance metric based on dynamic time warping.

In addition to the skeletal data, some methods directly use the depth information for activity recognition. These methods typically extract four-dimensional features from the depth data regarding both 3D spatial and temporal information. In [32], the depth data is projected into three orthogonal planes through the whole motion sequences to generate depth motion maps. HOG features are then computed from the three maps and concatenated to represent the whole action. In [21], the depth features are derived as histograms of normal orientations in the 4D space of depth, time, and spatial coordinates. The features are quantized using the vertices of a 600-cell polychoron in order to get the distributions of the features for the depth sequences.

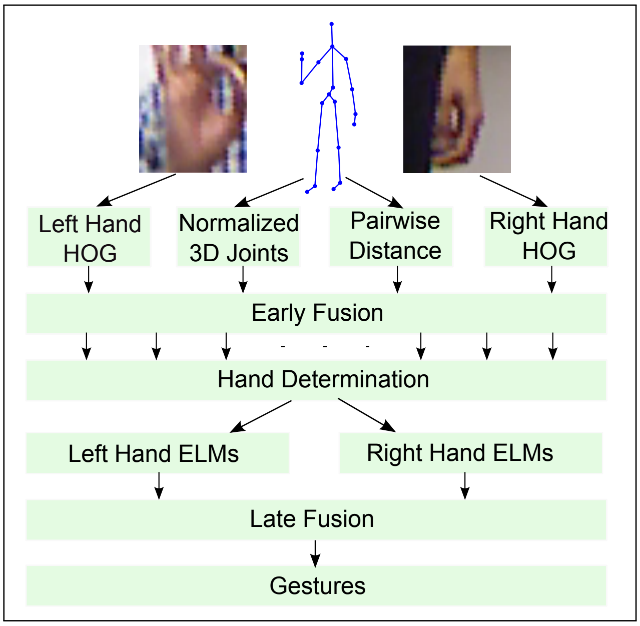
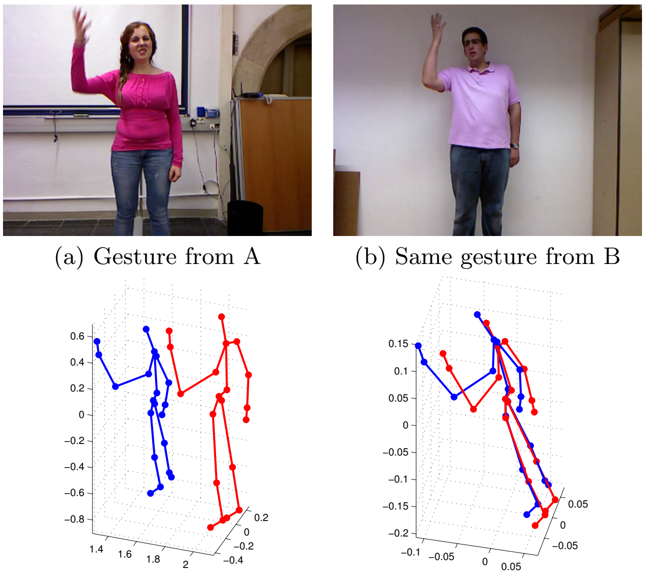


Figure 2: An overview of the proposed gesture recognition system.

Instead of using a single modality of the sensor data, many works [28, 25, 20] combine data from multiple sources to improve the recognition accuracy. In [8], actions from the database with 14 classes are classified by using both the RGB and depth data. The RGB data is incorporated with the depth information to detect interest points. HOG and HOF (histogram of optical flow) features and histograms of oriented depth gradients and a relative motion descriptor are extracted to form a bag of visual words, and a SVM is used as the classifier. The Berkeley Multimodal Human Action Database (MHAD) [20] contains data from a motion capture system, stereo cameras, depth sensors, accelerometers, and microphones. Features are extracted from each modality, and several classifiers are learned by various combination of the features through Multiple Kernel Learning. The experiments show that with more modalities combined the recognition accuracies are higher. In [28], the 3D joint coordinates are used to extract features, called local occupancy patterns, around the joints in the depth data. The features can capture the relations between the body parts and the objects in the interaction.

The selection of the classifier directly influences the accuracy, speed, and the computational complexity of the system. HMMs and CRFs are widely applied for sequential gesture recognition [7]. Another approach is to start from static posture recognition and to apply standard methods used for unsequential data, such as SVMs, on the frame level, and subsequently combine the results into the sequence level.

In this work, we use Extreme Learning Machines (ELM) [9] as frame-level classifiers, and include temporal information by utilizing feature differencing with a fixed time offset.



(a) Gesture from A

(b) Same gesture from B

(c) Original coordinates of A and B

(d) Normalized 3D joint Position of A and B

Figure 3: RGB frames and the corresponding skeleton information for the same gesture from two different performers.

ELM has recently been applied to many classification problems, and it often produces almost as good accuracies as kernel SVMs but with orders of magnitude faster training and testings times. For example, in [17] ELMs were used to recognize human activities from video data.

3. OVERVIEW

In our work, we use the skeletal data and the RGB video data for gesture recognition. We extract two kinds of features from the skeletal data: normalized 3D joint positions and pairwise distances between joints. In addition, based on the given hand joint positions on the skeleton model, we extract regional HOG features for the left and right hand based from the RGB frames. These features are concatenated in the early fusion stage to form different kinds of combinations. Next, by calculating the maximal scope of the hands' movements, we determine whether each gesture is left or right hand dominant. Based on this decision, the extracted features are classified with a classifier trained either for the left or the right hand as dominant. The classification stage contains multiple ELM classifiers learnt for combinations of features. The outputs of the multiple ELMs are fused again and aggregated to provide the final classification result for a gesture sequence. See Figure 2 for an overview of the proposed method.

4. FEATURE EXTRACTION

We extract features both from the skeleton models and from the hand regions of RGB frames.

4.1 Skeletal Features

Normalized 3D Joint Positions (NP). The skeletal data provides 3D joint positions of the whole body. The 3D coordinates of these joints are, however, not invariant to the position and size of the actors. Therefore we transform all skeletons into the same orientation by aligning the plane formed by the root and the hips from all frames into the same plane. The hips centers are overlapped and the sum of the distances between the corresponding hips are minimized, as shown in Figure 3. Figure 3a and Figure 3b show the same gesture in two different videos. The original skeletons of both gestures are drawn in Figure 3c where it is clearly visible that the two skeletons are not overlapping. The corresponding normalized 3D joint positions can be found in Figure 3d. After the transformation, the hip center of the skeleton is translated to the origin of the coordinates, and the planes formed by the hip center and hips are all rotated onto the same plane, which is randomly selected from one skeleton model. To eliminate the effect of different sizes of the performers, the transformed skeletons are normalized so that the sum of the distances of all connected joints is one.

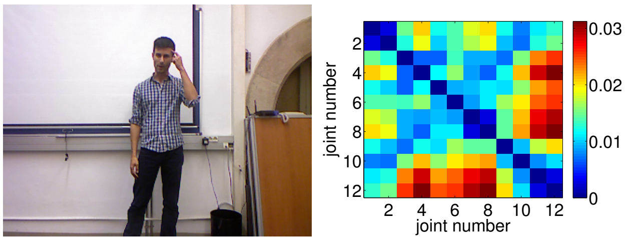


Figure 4: An RGB frame (left) and visualization of the corresponding pairwise distance (right).

For gestures related to whole body movement, the whole set of joints from the above feature can be used; for gestures only with partial body movement, such as hand and arm gestures, a subset can be selected. In this work, we use only the following upper-body joints: the spine, shoulder center, head, shoulders, elbows, wrists and the hands.

Pairwise Distances (PD). Another feature extracted from the skeletal data is the pairwise distances between joints. The distances form a vector which is then normalized to one. The elements of the feature vector can be calculated by

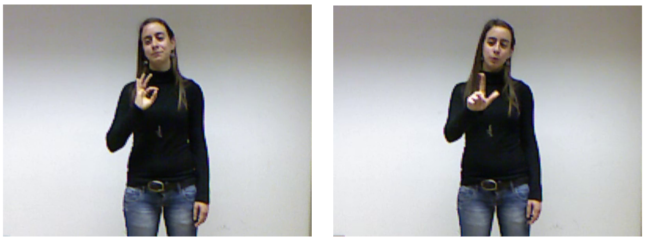
$$\frac{\|\mathbf{p}_i - \mathbf{p}_j\|}{\sum_{i \neq j} \|\mathbf{p}_i - \mathbf{p}_j\|} \ , \quad i \neq j \tag{1}$$

where $\mathbf{p}_i$ and $\mathbf{p}_j$ are the 3D positions of joints i and j. In this work, the joints used in (1) include the above 11 joints and the hip center. Figure 4 visualizes the distances between the upper-body joints for the gesture in the left frame.

Temporal Differencing. A gesture is formed by a sequence of frames. In order to preserve temporal information in the sequence, the temporal difference of features $\mathbf{x}_k^{td}$ for the kth frame in the sequence is calculated by

$$\mathbf{x}_k^{td} = \begin{cases} \mathbf{x}_k^d & 1 \leq k < k' \\ \dfrac{\mathbf{x}_k^d - \mathbf{x}_{k-k'+1}^d}{||\mathbf{x}_k^d - \mathbf{x}_{k-k'+1}^d||} & k' \leq k \leq K \end{cases} \tag{2}$$

where $\mathbf{x}_k^d$ is the NP or PD feature, k' is the temporal offset parameter, $1 < k' < K$, and K is the total number of frames. The final feature $\mathbf{x}_k$ of the frame k is a concatenation of $\mathbf{x}_k^d$ and $\mathbf{x}_k^{td}$, $\mathbf{x}_k = [\,(\mathbf{x}_k^d)^T \ (\mathbf{x}_k^{td})^T\,]^T$.



(a) OK (#11) (b) Non ce n'è più (#15)

Figure 5: Different ChaLearn 2013 hand gestures with similar skeleton alignment.

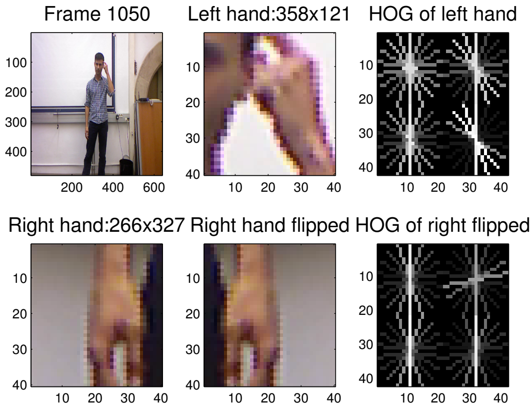


Figure 6: HOG features extracted from the left and (flipped) right hand region.

4.2 RGB Image Features

The skeletal features capture the movement of the whole body or a body region but are not able to capture hand gestures, which often present meaningful linguistic symbols. Examples from the ChaLearn 2013 dataset shown in Figure 5 suggest that skeletal features are not sufficient alone to distinguish all gestures in the dataset.

Histograms of oriented gradients (HOGs) [5], originally proposed for human pedestrian detection, have recently been successfully used in and many other applications, e.g. part-based object detection [6]. In this work we extract HOG features from the left and right hand separately. The right hand image is first flipped in the horizontal direction, which enables us to use a common classifier for both hands. We obtain the 2D pixel coordinates of the hand joints' positions from the skeletal data and extract grayscale HOG features using a grid of 2×2 cells and a cell size of 20×20 pixels. See Figure 6 for an example. We use the HOG implementation available in the VLFeat library [27].

5. CLASSIFICATION

Let us assume there are M gestures $\mathcal{A} = \{A_1, \ldots, A_M\}$ and let us define $c_m \in \{0, 1\}$, $1 \leq m \leq M$. If c_m is one, then the sequence belongs to the gesture A_m, otherwise it does not. The row vector $\mathbf{y} = [c_1 \ \ldots \ c_m \ \ldots \ c_M]$ indicates the gesture that the sequence belongs to. For each gesture there are multiple motion sequence examples, and each sequence s is represented by the features of its frames. That is, $s = \{\mathbf{x}_1, ..., \mathbf{x}_k, ..., \mathbf{x}_K\}$, where K is the number of frames. Now, $(\mathbf{x}_k, \mathbf{y})$ form K training input–output pairs for the classifier.

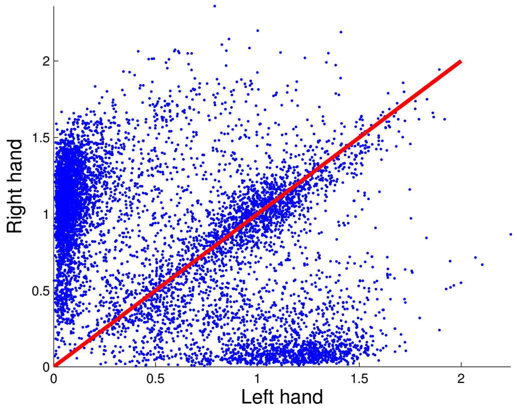


Figure 7: The total scopes of 3D movement of the left and right hands on the training sequences.

In this section, we briefly describe multi-class classification with the ELM algorithm and our methods to obtain sequence-level classification results and to feature fusion.

5.1 Determination of the Dominant Hand

We start the classification stage by determining the dominant hand of the actor. The used ChaLearn 2013 data contains several gestures that can be performed with either hand as the dominant one. In some gestures, the actors do use both hands but generally in a symmetric way. Therefore, we select the dominant hand for each gesture by measuring the the total scope of movement in 3D of both the left and right hand. The hand with a larger movement scope is marked as the dominant hand. See Figure 7 for a visualization, where the clusters of left dominant hand, right dominant hand, and both hands equally active, are clearly visible. We train separate ELMs for the cases where the left and the right hands are the dominant one, and during classification select the used ELM models based on similar scope analysis of the current gesture.

5.2 Extreme Learning Machine

The Extreme Learning Machine (ELM) [10] belongs to the class of single-hidden layer feed-forward neural networks (SLFNs). Traditionally such networks have been trained using a gradient-based method such as the backpropagation algorithm. In ELM, the hidden layer weights and biases do not need to be learned but are assigned randomly, which makes the learning extremely fast. The only unknown parameters are the output weights which can be obtained by finding a least-squares solution.

Given P training samples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^{P}$, where $\mathbf{x}_i \in \mathbb{R}^n$ and $\mathbf{y}_i \in \mathbb{R}^M$, the standard ELM model with L hidden neurons can be represented as

$$\mathbf{y}_i = f(\mathbf{x}_i) = \sum_{j=1}^{L} \boldsymbol{\beta}_j g(\boldsymbol{\omega}_j \cdot \mathbf{x}_i + b_j) \ , \qquad (3)$$

where $g(\cdot)$ is a nonlinear activation function, $\boldsymbol{\beta}_j \in \mathbb{R}^M$ are the output weights, $\boldsymbol{\omega}_j \in \mathbb{R}^n$ is the weight vector connecting the input layer to the jth hidden neuron and b_j is the bias of the jth hidden neuron. Both $\boldsymbol{\omega}_j$ and b_j are assigned randomly during the learning process. With $\mathbf{Y} = [\mathbf{y}_1^T\ \mathbf{y}_2^T \cdots \mathbf{y}_P^T]^T \in \mathbb{R}^{P\times M}$ and $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^T\ \boldsymbol{\beta}_2^T \cdots \boldsymbol{\beta}_L^T]^T \in \mathbb{R}^{L\times M}$, Eq. (3) can be written compactly as

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{Y}\ , \tag{4}$$

where the hidden layer output matrix $\mathbf{H}$ is

$$\mathbf{H} = \begin{bmatrix} g(\boldsymbol{\omega}_1 \cdot \mathbf{x}_1 + b_1) & \cdots & g(\boldsymbol{\omega}_L \cdot \mathbf{x}_1 + b_L) \\ \vdots & \ddots & \vdots \\ g(\boldsymbol{\omega}_1 \cdot \mathbf{x}_P + b_1) & \cdots & g(\boldsymbol{\omega}_L \cdot \mathbf{x}_P + b_L) \end{bmatrix}_{P\times L} . \tag{5}$$

If $L = P$, the matrix $\mathbf{H}$ is square and invertible, and the model can approximate the P training samples with zero error. However, in most cases the number of hiddes neurons is much smaller than the number of training samples, i.e. $L \ll P$, and we obtain the smallest norm least-squares solution of (4) as

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^\dagger \mathbf{Y}\ , \tag{6}$$

where $\mathbf{H}^\dagger$ is the Moore-Penrose generalized inverse of $\mathbf{H}$ [9].

5.3 Sequence Classification

Given a test sequence $s = \{\mathbf{x}_1, \ldots, \mathbf{x}_q, \ldots, \mathbf{x}_Q\}$, ELM provides the output weight (Eq. (3)) of each class m for each frame. We convert the outputs into probabilities with the logistic sigmoid function

$$p(c_m = 1|\mathbf{x}_q) = \frac{1}{1 + \exp(-\gamma y_{qm})}\ . \tag{7}$$

where y_{qm} is the mth component of $\mathbf{y}_q$.

If the sequence s belongs to an action A_m, every frame in the sequence also belongs to A_m. We therefore aggregate the frame-level probabilities of A_m to form the sequence-level classification result using a function $d_m : \mathbb{R}^Q \rightarrow \mathbb{R}$.

A straightforward approach would be to use the joint probability of the frames in s to determine the gesture

$$\begin{aligned} d_m &= p(c_m = 1 \mid s) \\ &= p(c_m = 1 \mid \mathbf{x}_1, \ldots, \mathbf{x}_q, \ldots, \mathbf{x}_Q) \\ &= \prod_{q=1}^{Q} p(c_m = 1 \mid \mathbf{x}_q) \end{aligned} \tag{8}$$

where temporal independence among the frames in a sequence is assumed on the last row.

We can also use any other function of the frame-wise probabilities as d_m. We found out empirically that using e.g. arithmetic mean improves the results over the joint probability. In this paper, we use the weighted arithmetic mean

$$d_m = \sum_{q=1}^{Q} w_q\, p(c_m = 1 \mid \mathbf{x}_q) \tag{9}$$

where the weights w_q are obtained from a normalized Gaussian distribution, $w_q = \frac{1}{Z}\mathcal{N}(q; \frac{Q}{2}, \sigma^2)$, normalized so that $\sum_{q=1}^{Q} w_q = 1$.

Finally, we classify the sequence s by

$$\hat{c}_m = \begin{cases} 1 & \text{if} \quad m = m' \quad \text{where } m' = \arg\max_i d_i \\ 0 & \text{otherwise}\ . \end{cases} \tag{10}$$

5.4 Fusion

We utilize both early and late fusion of the features (see Figure 2). In the early fusion stage, we concatenate either all three used features or pairs of two features before the ELM classification. In late fusion, we use the geometric mean to fuse the classification outputs of the different subsets of the feature-wise and early-fusion classification results.

6. EXPERIMENTS AND DISCUSSION

In this section, we describe gesture recognition experiments performed with the ChaLearn Multi-modal Gesture Recognition Challenge 2013 data [1]. The gestures in the dataset are $M = 20$ Italian cultural or anthropological signs, as shown e.g. in Figure 1 and Figures 3–5.

6.1 Setting

The ChaLearn 2013 dataset is split into three parts: training (7754 gestures), validation (3362 gestures), and test data (2742 gestures). We use about 6000 gesture sequences from the training data for learning our models and partition the videos in the validation data provided in the challenge evenly into a *validation set* and a *test set*. This is due to the lack of start and end points for the gestures in the provided test data. The validation set is used for parameter optimization and the test set is used to obtain the final results.

For ELM, the number of hidden neurons L is the only parameter we need to tune. In addition, we have the parameters k', γ and σ^2, corresponding to the temporal offset in (2), slope of the logistic sigmoid in (7), and the variance of the Gaussian weighting function in (9), respectively. In these experiments, we use k' corresponding to an offset of 300 milliseconds, $L = 1500$, and $\sigma = \frac{Q}{5}$. We optimize the value of γ for each feature vector separately.

As the performance measure, we use the standard measure of the challenge, i.e. the Levenshtein distance between the ground truth and the predicted gestures. In the gesture recognition setup, that is assuming that the start and end points of the gestures are known, this corresponds to the error rate of the classification algorithm.

6.2 Classification Based on Dominant Hand

We determine the dominant hand for each gesture as described in Section 5.1 and train separate classifiers for the cases where the left or the right hand is dominant. To illustrate the advantage of this approach, we also train classifiers without the dominant hand determination. The classification error rates on the validation set for the three features can be seen in Figure 8. The NP and PD features are based on the whole upper part of the body, so the feature vectors are identical in these two sets of experiments. The HOG features are extracted for each hand separately, and in the dominant hand experiment HOG features from only the dominant hand are used. In the experiment without the determination of the dominant hand, HOG features from both hands are concatenated to form a single feature vector.

From Figure 8 we can see the classification error rates with the dominant hand determination are lower for all features. The difference is largest for HOG, which can be explained by the increased noise caused by the inclusion of the often non-informative non-dominant hand. Furthermore, the NP and PD features also benefit from the division of the training data into two sets based on the dominant hand determination. In this case, each classifier can learn a more

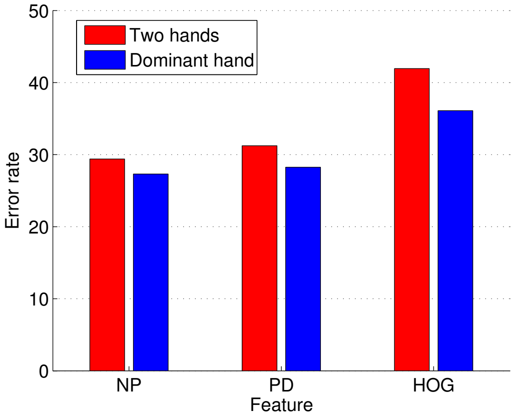


Figure 8: The error rates of the features when using both hands and using only the dominant hand.

Table 1: Error rates of different features and types of fusion on the validation and test sets.

features	validation set	test set
NP	0.273	0.293
PD	0.283	0.305
HOG	0.361	0.441
early fusion	0.171	0.187
late fusion	0.230	0.259
e+l fusion	0.178	0.217
optimal fusion	0.157	0.174

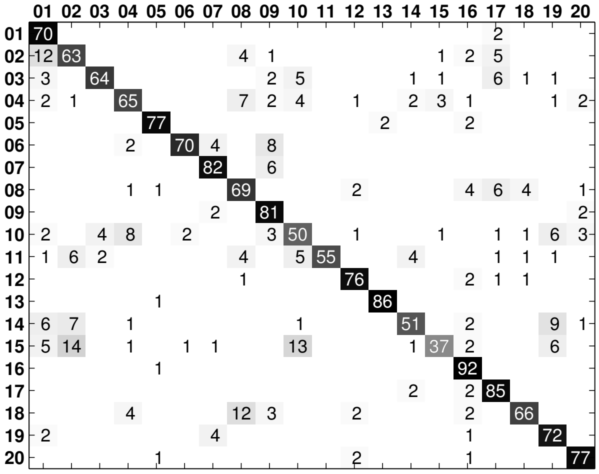
accurate model from the separated data. Therefore in our further experiments we use two sets of classifiers based on the dominant hand determination.

6.3 Results

Table 1 shows the error rates of the individual features and feature fusions on the validation and test sets. We see that the skeleton-based features outperform HOG and that normalized 3D joint positions (NP) feature has the lowest error rate of the individual features. The results can then be improved by both early and late fusion. The row *e+l fusion* combines the early fusion of all features to all individual features in the late fusion stage. Early fusion of all features performs particularly well, compared to both the best individual feature and to using late fusion.

The lowest overall error rate is achieved by considering all possible combinations of early and late fusion and selecting the best one based on the validation set. This *optimal fusion* is shown on the last row in Table 1, and consists of fusing all three features as well as all pairs of features in early fusion, and fusing these four features on the late fusion stage.

The error rates for different gestures are shown in Figure 9. It can be observed that the skeletal and hand features provide complementary information to each other: there are several gestures for which either the skeletal or the hand features clearly outperform the other modality. The power of the feature fusion can also be observed in Figure 9 as in most cases the error rate of the fusion is lower or at the level of the best feature even with considerable performance differences of the features. Finally, the full confusion matrix of the optimal fusion results on the test set are shown in Figure 10. The gesture classes in both figures correspond to the numbering used in the challenge [1].



	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20
01	70																2			
02	12	63						4	1						1	2	5			
03	3		64						2	5				1	1		6	1	1	
04	2	1		65				7	2	4		1		2	3	1			1	2
05					77								2			2				
06				2		70	4		8											
07							82		6											
08				1	1			69				2				4	6	4		1
09							2		81											2
10	2		4	8		2			3	50		1			1		1	1	6	3
11	1	6	2					4		5	55			4			1	1	1	
12								1				76				2	1	1		
13					1								86							
14	6	7		1						1				51		2			9	1
15	5	14		1		1	1			13				1	37	2			6	
16					1											92				
17														2		2	85			
18				4				12	3			2				2		66		
19	2						4									1			72	
20					1							2				1				77

Figure 10: Confusion matrix of the optimal fusion results on the test set.

With our current implementation written in Matlab, we can easily perform the recognition in real-time. The feature extraction takes 1.6 ms, 0.026 ms, and 24 ms per frame for the NP, PD, and HOG features, respectively. Moreover, the HOG computation could be speeded-up by orders of magnitude using GPUs (e.g. [22]). Classification with a single ELM takes about 0.1 milliseconds per frame. Training the ELMs is also fast, taking only around 1-3 minutes per model with the full ChaLearn 2013 training dataset. This makes it possible to retrain the system or to learn new gestures with a reasonable delay within an online application. All the experiments are conducted on a Intel(R) Xeon(R) CPU at 3.3 GHz and 16 GB of memory.

6.4 Discussion

The requirement of supporting online recognition poses a challenge to the feature extraction and classification algorithms. The distinctiveness of the feature significantly influences the effectiveness of the system, whereas simpler features make the processing of the feature extraction easier, reducing the complexity and speeding-up the computation. On the other hand, especially with multiple features, the evaluation time of the used classification algorithm should fast, as e.g. with linear classifiers, SLFNs, or decision trees.

A common approach to model spatio-temporal signals such as human actions or gestures is to use statistical temporal models such as HMMs, CRFs, or dynamic time warping. In this work, we approach the problem from the viewpoint of static pose recognition and use ELM as a standard multi-class classifier for frame-level classification. We bring in temporal information by differential features with a fixed time offset and then aggregate the results into the sequence level. This approach can provide an adaptive and fast method for action recognition that has been successfully applied to full-body mocap data classification with a large number of classes [3].

The ChaLearn 2013 dataset used in this work contains gestures that are difficult to separate based on the skele-

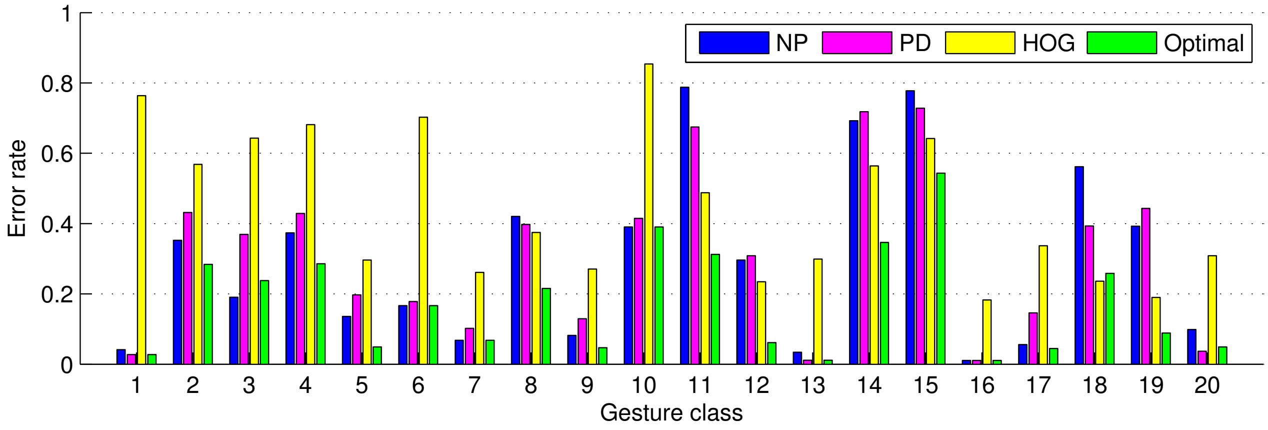


Figure 9: Error rates of the used features and the optimal fusion for each gesture on the test set.

ton model alone, so we introduce HOG-based hand features to obtain additional information about the hand configurations. The fusion experiments show that the HOG features provide a valuable addition that can reduce the overall error rate even though the skeletal features are more accurate on average. There are also numerous ways that the hand features could be improved, such as utilizing the depth images, tracking the hands to get a more accurate localization, and incorporating spatio-temporal features such as histograms of optical flow (HOF) [11].

Some gestures such as the Italian anthropological signs in the ChaLearn 2013 dataset can be performed with either hand as the dominant one. In this work, we take this into account by determining the dominant hand by measuring the total scopes of hand movement and training separate classifiers for gestures where the left and the right hand are dominant. Other approaches such as mirroring the skeleton models are also possible, but the separate classifiers permit us to model any potential systematic differences of gestures when performed with the left or the right hand. In the online setup, this however requires that we run both sets of classifiers in parallel.

In this work, we have limited the discussion to closed gesture recognition, that is, we have assumed that the start and end points for the gestures are known and that each performed gesture belongs to exactly one of the prespecified gesture classes. Generally, in an online setup, this is not the case and we have to perform both temporal gesture segmentation or gesture spotting and thresholding to reject non-gestures and gestures that do not belong to any of the known gesture classes. This is also the setup used in the ChaLearn gesture challenges. There are many proposed approaches for both problems, e.g. the threshold model for HMMs [12], but they are still largely unsolved. Our approach for the segmentation of the gesture sequences in the ChaLearn 2013 evaluation is based on hand movement. We detect the lowest position of the hands in the sequence and consider that frame as the reference frame. The distances to the reference frame are smoothed by a Gaussian filter. The gestures are segmented from the sequence by two adjacent minima, conditioned with several other conditions, such as the minimum number of frames in a gesture and the minimum difference between the maximum and minimum of the distance function. Our main contributions in this work are however in gesture recognition, so we omitted the gesture segmentation results from this paper.

The included audio modality plays an important role in the ChaLearn 2013 challenge for both the recognition accuracy and gesture segmentation [2]. This makes the direct comparison of methods utilizing audio to those that do not rather fruitless. In general, audio data can be very noisy or not available in many applications, so purely camera-based methods for gesture recognition are also needed.

7. CONCLUSIONS

In this work, we propose a simple, effective, and easily implementable gesture recognition system for RGB and skeletal data. We extract two kinds of features, normalized 3D joint positions and pairwise distances, from the skeletal data for each frame in the gesture sequence. These features are easy to obtain but still provide distinctive information about the posture. To capture the hand gestures which present meaningful linguistic symbols, HOG features are extracted from the hands regions of RGB data. We use multiple extreme learning machines as the classifiers for the left hand and right hand dominant gestures separately. The outputs from the multiple ELMs are then fused and aggregated to the sequence level to provide the final classification results. ELMs were shown to provide high recognition accuracy and, at the same time, both classification and the training of the models are very fast. Together with our computationally light features this makes our system readily applicable for online recognition applications.

8. ACKNOWLEDGMENTS

This work was funded by Aalto MIDE programme (project UI-ART) and grants Multimodally grounded language technology (254104) and Finnish Center of Excellence in Computational Inference Research COIN (251170) of the Academy of Finland.

9. REFERENCES

[1] ChaLearn. Multi-modal Gesture Recognition Challenge 2013. *http://gesture.chalearn.org/*.

[2] ChaLearn. Results of ChaLearn multi-modal gesture recognition challenge 2013.

http://iselab.cvc.uab.es/CHALEARN-MMGesture-ChallengeResults2013.pdf.

[3] X. Chen and M. Koskela. Classification of RGB-D and motion capture sequences using extreme learning machine. In *Proceedings of the 18th Scandinavian Conference on Image Analysis*, volume 7944 of *LNCS*, Espoo, Finland, June 2013. Springer Verlag.

[4] H. Chung and H.-D. Yang. Conditional random field-based gesture recognition with depth information. *Optical Engineering*, 52(1):017201–017201, 2013.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[6] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[7] I. Guyon, V. Athitsos, P. Jangyodsuk, H. J. Escalante, and B. Hamner. Results and analysis of the ChaLearn Gesture Challenge 2012. In *Proceedings of 21st International Conference on Pattern Recognition*, Tsukuba, Japan, November 2012.

[8] S. Hadfield and R. Bowden. Hollywood 3d: Recognizing actions in 3d natural scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, Portland, Oregon, USA, June 2013.

[9] G. Huang, H. Zhou, X. Ding, and R. Zhang. Extreme learning machine for regression and multiclass classification. *Systems, Man, and Cybernetics, Part B, IEEE Transactions on*, 42(2):513–529, 2012.

[10] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew. Extreme learning machine: theory and applications. *Neurocomputing*, 70(1):489–501, 2006.

[11] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, 2008.

[12] H.-K. Lee and J. H. Kim. A HMM-based threshold model approach for gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):961–973, October 1999.

[13] W. Li, Z. Zhang, and Z. Liu. Expandable data-driven graphical modeling of human actions based on salient postures. *Circuits and Systems for Video Technology, IEEE Transactions on*, 18(11):1499–1510, 2008.

[14] Y. M. Lui. Human gesture recognition on product manifolds. *Journal of Machine Learning Research*, 13:3297–3321, 2012.

[15] M. R. Malgireddy, I. Nwogu, and V. Govindaraju. Language-motivated approaches to action recognition. *Journal of Machine Learning Research*, 14:2189–2212, 2013.

[16] A. Menache. *Understanding motion capture for computer animation and video games.* Morgan Kaufmann Publishers, 2000.

[17] R. Minhas, A. Baradarani, S. Seifzadeh, and Q. Jonathan Wu. Human action recognition using extreme learning machine based on visual vocabularies. *Neurocomputing*, 73(10):1906–1917, 2010.

[18] S. Mitra and T. Acharya. Gesture recognition: A survey. *IEEE Transactions on Systems, Man and Cybernetics, Part C*, 37(3):311–324, May 2007.

[19] M. Müller and T. Roder. Motion templates for automatic classification and retrieval of motion capture data. In *Proceedings of the Eurographics/ACM SIGGRAPH symposium on Computer animation*, volume 2, pages 137–146, Vienna, Austria, 2006.

[20] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Berkeley mhad: A comprehensive multimodal human action database. In *Applications of Computer Vision (WACV), IEEE Workshop on*, 2013.

[21] O. Oreifej, Z. Liu, and W. Redmond. Hon4d: Histogram of oriented 4d normals for activity recognition from depth sequences. In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

[22] V. Prisacariu and I. Reid. fastHOG - a real-time GPU implementation of HOG. Technical Report 2310/09, Oxford University, 2009.

[23] M. Raptis, D. Kirovski, and H. Hoppe. Real-time classification of dance gestures from skeleton animation. In *Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 147–156. ACM, 2011.

[24] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proc. Computer Vision and Pattern Recognition*, June 2011.

[25] Y. Song, D. Demirdjian, and R. Davis. Continuous body and hand gesture recognition for natural human-computer interaction. *ACM Transactions on Interactive Intelligent Systems*, 2(1):5, 2012.

[26] H.-I. Suk, B.-K. Sin, and S.-W. Lee. Hand gesture recognition based on dynamic bayesian network framework. *Pattern Recognition*, 43(9):3059–3072, 2010.

[27] A. Vedaldi and B. Fulkerson. VLFeat: A library of computer vision algorithms. *http://www.vlfeat.org/*.

[28] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *Computer Vision and Pattern Recognition, 2012 IEEE Conference on*, 2012.

[29] L. Xia, C.-C. Chen, and J. Aggarwal. View invariant human action recognition using histograms of 3d joints. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012.

[30] L. Xia, C.-C. Chen, and J. K. Aggarwal. Human detection using depth information by Kinect. In *Workshop on Human Activity Understanding from 3D Data in conjunction with CVPR (HAU3D)*, Colorado Springs, USA, 2011.

[31] X. Yang and Y. Tian. Eigenjoints-based action recognition using naive-bayes-nearest-neighbor. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012.

[32] X. Yang, C. Zhang, and Y. Tian. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *Proceedings of the 20th ACM International Conference on Multimedia*, 2012.