Gesture Recognition using Template Based Random Forest Classifiers

Necati Cihan Camgöz, Ahmet Alp Kindiroglu, Lale Akarun

Bogazici University,
Computer Engineering Department, Istanbul
{cihan.camgoz, alp.kindiroglu, akarun}@boun.edu.tr

Abstract. This paper presents a framework for spotting and recognizing continuous human gestures. Skeleton based features are extracted from normalized human body coordinates to represent gestures. These features are then used to construct spatio-temporal template based Random Decision Forest models. Finally, predictions from different models are fused at score level to improve overall recognition performance.
Our method has shown competitive results on the CHALEARN 2014 Looking at People dataset. Trained on a dataset of 20 gesture vocabulary and 7754 gesture samples, our method achieved a Jaccard Index score of 74.6% on the test set, reaching 7th place among contenders. Among methods that exclusively used skeleton based features, our method obtained the highest recognition performance.

Keywords: Template Based Learning, Random Decision Forest, Gesture Recognition

1 Introduction

Gestures are natural and expressive tools of human communication. As computers take a greater role in daily life, creating natural human computer interaction methods, such as gesture interfaces, has become a necessity. Especially hand and arm gestures, which people commonly use to communicate with each other, have now become commonly used human computer interaction methods [12]. However, there are still limitations in sensing, detecting and modelling gestures. Recent developments such as the emergence of consumer depth cameras and the availability of large annotated corpora have turned automatic gesture recognition to a competitive and active research field.

Automatic Gesture Recognition aims to spot and distinguish gestures from a gesture vocabulary given a sensory input. However, imperfect human pose detection and recognition coupled with spatio-temporal variability of the gestures makes distinguishing between gestures a challenging task [20].

This research was supported by the joint Bogazici University and Netaş SANTEZ project (0341.STZ.2013-2) from the Turkish Ministry of Science, Industry and Technology.

Many state-of-the-art gesture recognition systems use depth cameras to capture gestures [26]. Video-based gesture recognition deals with challenging tasks, such as the difficulty of locating hands in the presence of rapid arm movements and lighting changes [12, 7]. Depth cameras alleviate some of these difficulties as they are able to operate under difficult lighting conditions where RGB cameras fail [28].

In the literature, video-based gesture recognition methods differ according to two criteria: gesture cues and learning methods for training gesture recognition systems.

Once a gesture has been sensed, it is described via meaningful mathematical features. The chosen features often depend on the elements of the gesture being detected. In a typical gesture learning module, features like joint locations, angles between joints, hand locations, trajectories and hand shape parameters are used. These features can be obtained from modalities such as motion, color and depth. In conjunction with statistical learning methods, these features are then used to distinguish classes of gestures from each other.

Classification of human gestures relies on learning temporal information as well as spatial information. Due to the spatio-temporal nature of gestures, learning the temporal structure of human actions is crucial in building successful gesture recognition models. In the literature, three common approaches are used to learn the temporal structure of models [18]:

The first of these approaches omits temporal dependencies and models gestures using either individual key frames or histogram of feature sequences. In vocabularies where the temporal aspect of gestures is static (meaning there is not much variation in appearance during the gesture), using a single representative image may be sufficient. In [27], Carllson and Sullivan use differences in edge templates to classify key frame images. Likewise, using features of multiple frames in a histogram setting, such as the temporal bag of words approach [21], builds effective classifiers by modelling the frequencies of different features. However, such models fail to distinguish among similar gestures with different temporal ordering.

A more popular approach to temporal modelling is using action grammars. In these approaches, features are grouped into certain configurations, such as states. Changes among these states are modelled using graphical models. Hidden Markov Models [19] are the most popular representation method among these probabilistic methods. Since the works of Starner and Pentland [25] in recognizing American Sign Language letters and Yamato et al. [30] in recognizing tennis gestures, they have been used extensively for gesture learning. Other approaches, such as Conditional Random Fields [14] or Autoregressive Models [1] have also been used.

Another approach to temporal modelling is by using gesture templates. Instead of modelling frame features into clusters and representing the interactions of these clusters, these models deal with learning static sequential groups of features called templates. Models for these approaches are often constructed by either stacking a sequence of features together or by stacking a sequence of images together to learn features in the spatio-temporal domain. Techniques such as motion history images [3] are popular approaches of this technique.

While these approaches model blocks of features over a temporal domain, they have no mechanism for detecting temporal changes such as slower execution of a gesture. To handle such changes, the model should be trained with either temporally similar samples or temporally normalized samples using approaches, such as Dynamic Time Warping [22].

Since templates are obtained by concatenating spatial features onto fixed sized vectors, non-temporal machine learning techniques, such as support vector machines, nearest neighbour methods or ensemble methods can be used to learn such representations [2].

In this paper, we present a continuous gesture recognition framework for recognizing continuous Italian gestures [23]. We extract skeleton based features from human body part annotations provided for the ChaLearn 2014 Looking at People Competition [8] . We use template based Random Decision Forest [4] methods for continuous per-frame gesture recognition. We concatenate a temporal sequence of features to form our template; and experiment with different sampling strategies. In Section 2, we outline the Chalearn competition dataset. In Section 3, we describe our gesture recognition methodology. Then we present our experimental setup and results in Section 4 and share our conclusions in Section 5.

2 ChaLearn 2014 Italian Gestures Dataset

The Italian Gestures dataset [23], featured by ChaLearn 2014, was designed to evaluate user independent continuous Gesture Recognition performance. The dataset consists of 13,858 gestures from a vocabulary of 20 Italian cultural/ anthropological signs performed by 27 unique users. The list of Italian gestures in the dataset can be seen in Table 1.

Table 1. List of Italian Gestures in the dataset

Italian Gestures			
vattene	ok	vieniqui	cosatifarei
perfetto	basta	furbo	prendere
cheduepalle	noncenepiu	chevuoi	fame
daccordo	tantotempo	seipazzo	buonissimo
combinato	messidaccordo	freganiente	sonostufo

The dataset was recorded by Microsoft Kinect sensors, and it includes skeleton model [24], user mask, RGB and depth images. A visualization of dataset modalities can be seen in Figure 1. The dataset consists of 450 development, 250 validation, and 240 test videos in which there are a total of 7754, 3362, and 2742 individual gestures, respectively.

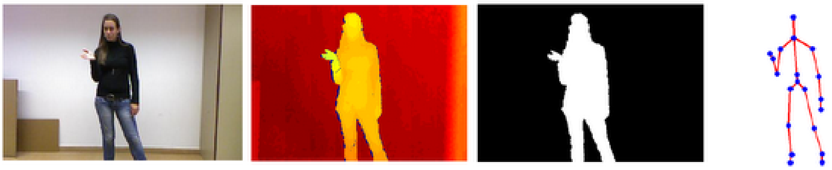


Fig. 1. Data modalities of the dataset. From left to right: RGB Images, Depth Images, User Mask and Skeleton Model

The dataset was featured by ChaLearn 2014 Looking at People competition's Track 3: Gesture Recognition. The emphasis of the gesture recognition track was on multi-modal automatic learning of a set of 20 gestures performed by several different users, with the aim of performing user independent continuous gesture spotting.

3 Method

Our gesture recognition method takes the skeleton model of gesticulating users as input. These models were provided by the dataset and contain 2.5D joint coordinates and their rotations. Given a skeleton model as input, our method goes through the following five stages:

1. Joint coordinates are normalized.
2. Gestures are represented by the skeleton based features that are extracted from the set of normalized coordinates and joint rotations.
3. Gesture Templates are constructed to incorporate temporal information for spatial machine learning methods.
4. Gesture representations are then given to Random Desicion Forests to perform gesture spotting and gesture classification.
5. Score fusion is used to combine predictions of multiple classification models.

The block diagram of our framework can be seen in Figure 2.

3.1 Joint Coordinate Normalization:

The skeleton Model provided by the dataset contains joint world coordinates, joint pixel coordinates and their rotations in each frame of a video. World coordinates represent the global position of a tracked joint in 2.5D space.

We normalize the world coordinates to obtain comparable and user invariant joint coordinates. To do so, we move the hip center to the origin $(0 \quad 0 \quad 0)^T$ in $3D$ space and the shoulder center to $(0 \quad 1 \quad 0)^T$ in all frames. Then, a rotation of the body around the y axis is performed in order to bring the left shoulder to the z=0 plane, thus making all users turn straight towards the camera. Visualization of these preprocessing steps can be seen in Figure 3.

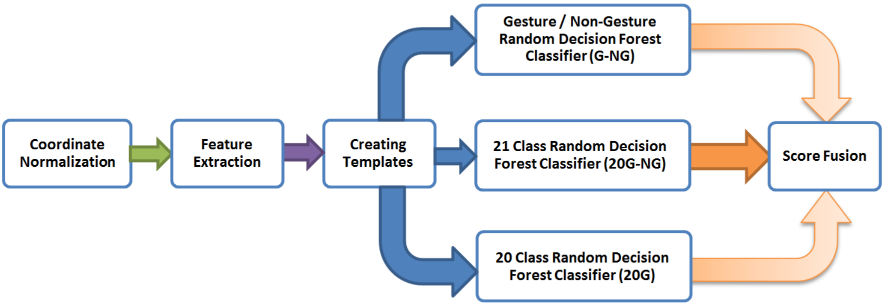

Coordinate Normalization
Feature Extraction
Creating Templates
Gesture / Non-Gesture Random Decision Forest Classifier (G-NG)
21 Class Random Decision Forest Classifier (20G-NG)
20 Class Random Decision Forest Classifier (20G)
Score Fusion

Fig. 2. Our Gesture Recognition Framework

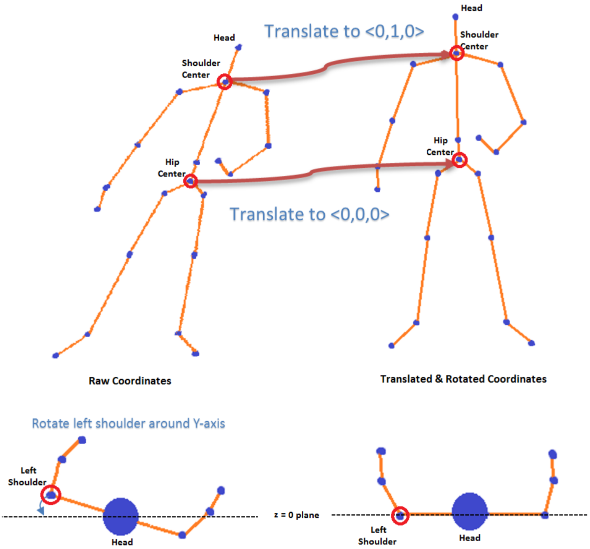


Fig. 3. World coordinate normalization

3.2 Gesture Representation

A total of six groups of features were extracted from the skeleton model for each frame. The features we have used to represent gestures are as following:

Upper Body Joint World Coordinates: The world coordinates represent the global position of a tracked joint in 2.5D space. Each joint coordinate is represented by C_x, C_y, C_z components of the subject´s global position in milimeters [23].

From the upper body joints, we have used Head, Shoulder Center, Left & Right Shoulder, Left & Right Elbow, Left & Right Wrist, Left & Right Hand, Spine and Hip Center's world coordinates thus making 36 features in total.

Normalized Upper Body Joint World Coordinates: We have obtained normalized world coordinates as explained in Section 3.1. Each normalized joint coordinate is represented by N_x, N_y, N_z components of the subject´s global position after normalization. We used the same 12 joints from the unprocessed joint coordinates, thus making another 36 features in total.

Upper Body Joint Rotations: The world rotation contains the orientation of skeleton bones in terms of absolute transformations. Each joint orientation is represented with four quaternion values $\theta_w, \theta_x, \theta_y, \theta_z$. The orientations of a bone is relative to the previous bone, and the hip center contains the orientation of the subject with respect to the sensor.

Skeleton Based Features: Instead of using hand based features, which can be unreliable due to sensor limitations, quantized wrist positions, wrist movements and trajectories were extracted as additional features.

The gesture space is divided into nine regions by using the middle point of shoulder bones and spine as seen in Figure 4. The quantized positions, representing the centroid of the region where the wrists are located, are the features W_R and W_L.

Additionally, wrist trajectories and their displacements between frames are used as features (T_R and T_L & M_R and M_L).

Since the gestures in this dataset mainly differ in shoulder, elbow, and wrist positions; bone orientations are also used as supplementary features ($B_{1:4}$).

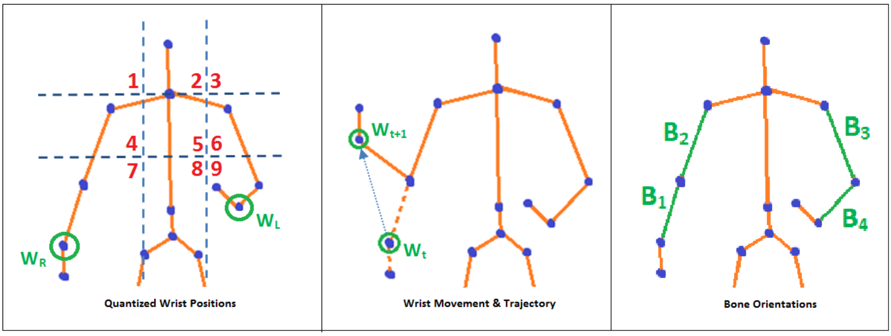


Fig. 4. Skeleton Based Features

3.3 Constructing Gesture Templates

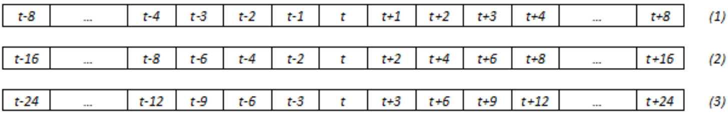
As mentioned in Section 3.2, our feature vector belonging to frame at time t (F_t) consists of the features in Equation 1.

$$F_t =< C_x, C_y, C_z, N_x, N_y, N_z, \theta_w, \theta_x, \theta_y, \theta_z, W_R, W_L, T_R, T_L, M_R, M_L, B_{1:4} > \tag{1}$$

Due to their lack of temporal mechanisms, spatial machine learning methods are not suitable for recognizing gestures. In order to use powerful spatial classifiers, such as ensemble methods with temporal data, temporal features need to be of fixed sizes. In our framework, this is achieved through padding per-frame features (F_t) together in fixed k sized structures called **templates** (T_t) as in Equation 2.

$$T_t =< F_{t-\frac{k-1}{2}}, ..., F_{t-1}, F_t, F_{t+1}, ..., F_{t+\frac{k-1}{2}} > \tag{2}$$

In template based gesture recognition, increasing template size enhances temporal representation. However, increasing the total number of feature dimensions needs to be limited at a point due to memory and computational power restrictions of development systems. To overcome this, selection methods of frames for templates can be altered. We have experimented with the original rate videos, 2x downsampled videos and 3x downsampled videos as shown in Figure 5.



t-8	...	t-4	t-3	t-2	t-1	t	t+1	t+2	t+3	t+4	...	t+8	(1)
t-16	...	t-8	t-6	t-4	t-2	t	t+2	t+4	t+6	t+8	...	t+16	(2)
t-24	...	t-12	t-9	t-6	t-3	t	t+3	t+6	t+9	t+12	...	t+24	(3)

Fig. 5. Frame selection for original rate (1), 2x downsampled (2) and 3x downsampled (3) videos with the template size of 17

3.4 Gesture Recognition with Random Desicion Forests

Random Decision Forest (RDF) is a supervised classification and regression technique that has become widely used due to its efficiency and simplicity. RDF's are an ensemble of random decision trees (RDT) [4]. Each tree is trained on a randomly sampled subset of the training data. This reduces overfitting in comparison to training RDTs on the entire dataset therefore increasing stability and accuracy.

During training, a tree learns to split the original problem into smaller ones. At each non-leaf node, tests are generated through randomly selected subsets of features and thresholds. The tests are scored using the decrease in entropy, and best splits are chosen, and used for each node [4]. Based on these tests, bone leaf nodes separate the data into their left and right child nodes. At a leaf node, only samples that belong to the same class remain.

Classification of a frame is performed by starting at the root node and assigning the pixel either to the left or to the right child recursively until a leaf node is reached. Majority voting is used on prediction of all decision trees to decide on the final class of the gesture.

3.5 Score Fusion

In order to explore the effect of late fusion, four different fusion strategies were used on the dataset.

These methods were used to fuse the predictions from three different models. To predict the label of a frame given its features, a 21 class classifier was used. However, as reported by Kuznetsova et al. [10], we have observed that random forest classifiers perform better when a lower number of classes are classified in a hierarchy. For this reason, the task of separating gestures from non-gestures and separating gestures among each other were handled by training different RDF classifiers.

Three RDF models were trained using the same development dataset: the 2 class Gesture/Non-Gesture (G-NG) model, the 20 class Gesture only model (20G) and the 21 class combined model (20G-NG).

1. **Non-Gesture Suppression:** Using the G-NG and 20G-NG methods, all non-gesture results from the G-NG set were imposed on the 20G-NG set. Remaining class labels were untouched.
2. **Median Filtering:** In addition to Non-Gesture Suppression, median filter of length three was used to suppress single frame anomalies.
3. **Filtering Based Gesture Suppression:** Using the G-NG and 20G-NG methods, gestures from the 20G-NG predictions were replaced using a filtering approach. A frame based majority filtering of size M was applied on each frame that was labeled as a gesture by G-NG and as a nongesture by 20G-NG. Since the correct label could not be predicted from the 20G-NG predictions, the most occuring non non-gesture label in a M size neighborhood was assigned to that frame.
4. **20G Model Based Gesture Suppression:** An additional 20G model was used in conjuction with 20G-NG and G-NG models to perform better fusion. Each frame that was labeled as a gesture by G-NG and as a nongesture by 20G-NG was assigned the value indicated by the 20G model.

4 Experiments & Results

To verify the effectiveness of the proposed approach, we have used the Chalearn Gesture dataset. We performed our parameter optimization on the validation set, and reported a single test result on the test set with our best validation parameters.

All RDF models were trained with 134K features, where 134 is the number of features we have used and K is the template window size. At each node, these features were sampled with replacement from the training set and M features were selected, where $M = \sqrt{134K}$. A total of 100 trees were trained with each model to a maximum tree depth of 100. These values were determined through experimentations with the validation set.

In all the experiments, we use the Jaccard Index (JI) as our evaluation method. JI is a commonly used success criterion for the evaluation of gesture spotting. It is preferred in situations where penalizing false positives is considered as important as rewarding true positives. In this sense, for each frame belonging to one of the $n = 20$ gesture categories, Jaccard Index is defined as:

$$J_{s,n} = \frac{A_{s,n} \bigcap B_{s,n}}{A_{s,n} \bigcup B_{s,n}} \tag{3}$$

$A_{s,n}$ is the ground truth of gesture n at sequence s, and $B_{s,n}$ is the prediction for such a gesture at sequence s. $A_{s,n}$ and $B_{s,n}$ are vectors where entries denote frames in which the n^{th} gesture is being performed[23].

Performance is evaluated based on the mean Jaccard Index among all gesture categories for all sequences, where all gesture categories are independent. In addition, when computing the mean Jaccard Index, all gesture categories have the same importance as indicated by the performance criteria of the Chalearn 2014 Challenge[23].

Using the Jaccard Index, we have tested our system with several parameters such as template size, template selection strategy and fusion techniques.

Template Size Optimization: The size and selection criterion of the temporal gesture templates were crucial parameters of the designed system. After obtaining per-frame spatial features, templates for each frame were formed by stacking features belonging to consecutive frames. We have experimented with template sizes from 1 to 21 while incrementing template size by four at each experiment.

Experiments showed that increasing the template size increased overall recognition performance. While 36.9% performance was obtained by using single frame templates, templates formed by the concatenation of 21 consecutive frames yielded the best results as 76%. The effects of changing the size of the templates for the 20G-NG classification can be seen in Table 2 and Figure 6.

Due to large memory and computation time requirements, further experimenting with templates larger than 17 frames was not feasible. However, the results displayed a positive correlation between recognition performance and the length of represented temporal interval. To represent larger intervals without exceeding the memory limitations, template sampling (or in other words, video downsampling) strategies were applied.

Table 2. Effect of template size on per-frame gesture recognition performance

Template Size:	1	5	9	13	17	21
Jaccard Index:	36,9%	41,3%	70,2%	73,0%	74,8%	76,0%

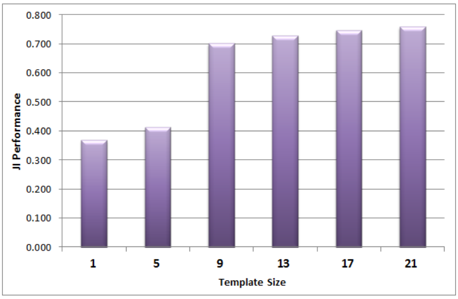


Fig. 6. Effect of template size on per-frame gesture recognition performance

We have experimented with original rate videos, 2x downsampled videos and 3x downsampled videos. By adding every 2 and 3 consecutive frames, temporal intervals of length 33 and 49 were trained as 17 frame templates. Compared to the 74.8% accuracy obtained without downsampling, adding 2x downsampling yielded an accuracy of 77.3%. The results can be seen in Table 3.

Table 3. Effect of template length on performance using 17 frame feature vectors

Downsampling rate:	none	2	3
Represented Interval Size:	17	33	49
Jaccard Score:	74,83%	77,34%	77,24%

Score Fusion: In order to explore the effect of late fusion, four different fusion strategies were used on the dataset. Experiments were performed by training three separate models on the development set using different sets of labels. These are:

- 2 class G-NG method
- 20 class 20G method
- 21 class 20G-NG method

To decide on fusion strategy, baseline performances using 17 template size with no downsampling were obtained. The G-NG method was the most accurate with a 2 class accuracy of 93%. The 20G-NG model had a 21 class accuracy of 88% with a hugely imbalanced class distribution favouring non-gestures. The 20G achieved the lowest accuracy with 80% performance. As co-articulation from non-gesture frames aid in the detection of gestures, the lack of nongesture samples in training may have resulted in the lower performance of the 20G model.

As a result, in order to boost our performance on 21 class prediction of frame labels, the G-NG and 20G methods were used to boost the recognition performance of the 20G-NG classifier using different approaches.

Experimental results showed that 21 sized templates cumulatively using NG suppression, median filtering and filtering based gesture suppression approaches achieved $77,6\%$ recognition accuracy. The results of different fusion methods based on this strategy can be seen in Figure 7 and Table 4.

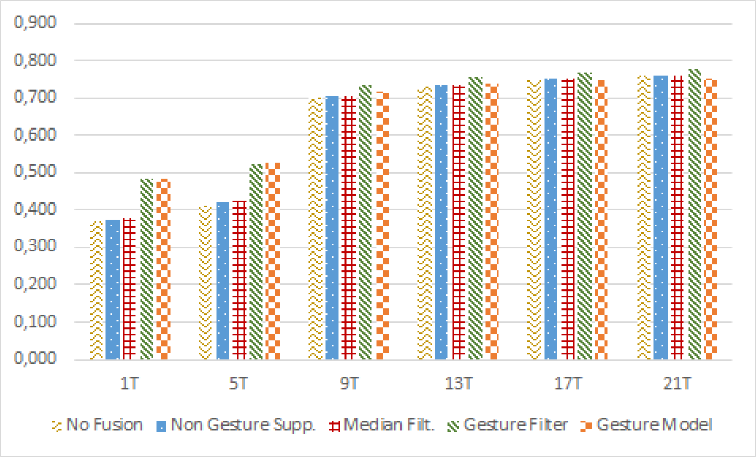


Fig. 7. Evaluation of fusion methods with different template sizes

Table 4. Evaluation of fusion methods on 17 size gesture templates

Method:	No Fusion	NG Supp.	Median Filt.	Gesture Filter	Gesture Model
1T	36,9%	37,5%	38,0%	48,4%	48,6%
5T	41,3%	41,9%	42,3%	52,2%	52,8%
9T	70,2%	70,7%	70,7%	73,6%	71,9%
13T	73,0%	73,3%	73,4%	75,6%	73,8%
17T	74,8%	75,0%	75,1%	76,9%	74,8%
21T	76,0%	76,1%	76,2%	**77,6%**	75,4%

The effectiveness of the fusion methods on eliminating gesture-nongesture misclassifications can be seen by examining the confusion matrices in Figure 8.

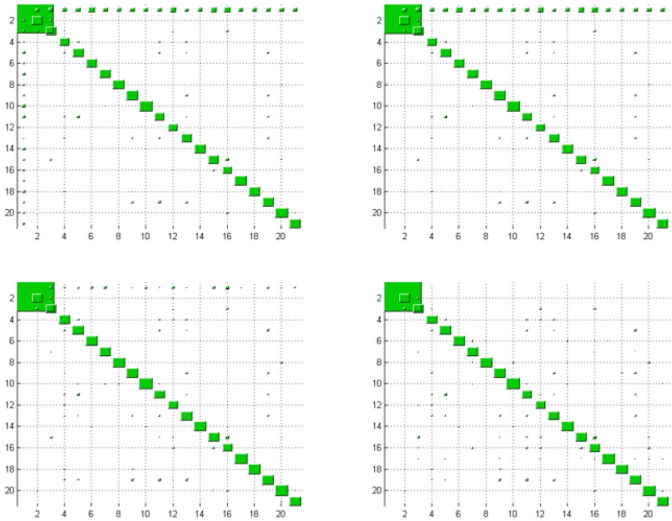


Fig. 8. Confusion matrices for different fusion metrics: No Fusion(top left), NG Supp.(top right), Gesture Filter(bottom left), Gesture Model(bottom right). Large boxes on top left corner represents the numerous non-gesture class.

Combination of Fusion and Downsampling: By combining five different fusion methods with 3 different down-sampling strategies, we have obtained the best results of our method on the validation set. Using 2x Downsampling with NG suppression, median filtering and filtering based gesture suppression, we have achieved 78.75% accuracy compared to the 74.83% accuracy of our baseline method. The results of these models are presented in Figure 9 and Table 5.

Table 5. Effect of template length on performance using 17 frame feature vectors

Fusion Method:	none	NG Supp.	Median Flt.	Gesture Flt.	Gesture Mod.
17T no Downsampling:	74,83%	75,04%	75,12%	76,91%	74,81%
33T 2x Downsampling:	77,34%	77,15%	77,40%	**78,75%**	76,35%
49T 3x Downsampling:	77,24%	77,07%	77,29%	78,69%	76,27%

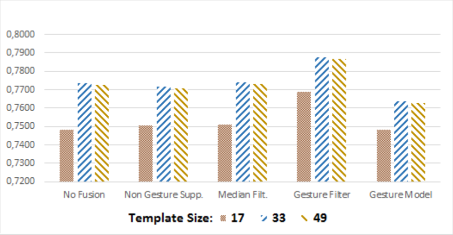


Fig. 9. Effect of template size on per-frame gesture recognition performance

Comparison of Test Results with other Methods: The overall performance evaluation and comparison of the system was done using the evaluation framework of the ChaLearn competition [8] . Due to timing and complexity considerations, we have submitted our template based random forest (**tbRF**) method results with no downsampling and 17 template sized feature vectors. A summary of the challenge results can be seen in Table 6.

Table 6. Comparison of Recognition performance on Chalearn test set with similar studies [8]

Method	J.Score	Modality	Features	Classifier
[15]	84,99%	rgb, depth, skeleton	Raw, Skeleton Joints	DeepNN
[13]	83,39%	rgb, depth, skeleton	HOG, skeleton	Adaboost
[5]	82,67%	rgb, skeleton	HOG, skeleton	MRF, KNN
[16]	79,19%	rgb	HOG, HOF, VLAD	SVM
[17]	78,80%	rgb, depth	Raw, Skeleton Joints	CNN
[29]	78,73%	depth, skeleton	Raw	HMM, DeepNN
tbRF	**74,66%**	**skeleton**	Skeleton Based	RDF
[9]	74,54%	skeleton	Skeleton / Fisher Vector	SVM
[6]	64,89%	rgb, depth, skeleton	STIPS	RDF
[11]	59,71%	mask, depth, skeleton	HOG , Skeleton	SVM,HMM

Looking at the results, we can claim that we obtain the best results among the papers that only use skeleton based features. Observation of the close performance of [9] in Table 6 may suggest the limits of skeleton based features. However, we were able to show that performance was increased through the exploitation of the temporal characteristics.

5 Conclusions

The paper has described a system for the visual recognition of gestures. The system takes body coordinates extracted by Microsoft Kinect, and performs feature normalization and template based RDF learning to automatically recognize gestures.

We have achieved a 78.75% Jaccard Score on the evaluation dataset using 2x downsampled video based templates, and 76.9% with original rate video based templates. These results were justified as we achieved a final Jaccard index score of 74.6% on the ChaLearn 2014 challenge test set using original rate video based templates. This score placed our team at the 7th place among 17 submitters in the third track of the competition. From the submitted fact sheets, it appears that the method presented in this paper was the highest performing among other methods that exclusively used features based on skeleton data. We also note that the 74.6% score on the test set did not include downsampling approaches as these methods were not implemented before the challenge deadline.

Furthermore, the recognition models were only trained on the training set. We were unable to increase the size of our development set with validation samples, as the 31GB memory required by the random forest algorithm made training impractical. While we were able to verify that expanding the template size to 21 frames improved overall recognition performance, we were unable to perform additional experiments due to computational limitations. Therefore, reducing the memory requirements through better memory management or more efficient feature representations may allow the method to achieve even higher results.

Possible future works to improve continuous recognition performance include changing random forest feature sampling strategies and incorporating depth/color based features to increase the discriminative power of the gesture representation methods. Furthermore, using transfer learning methods to increase user independence may also benefit overall recognition performance.

References

1. Agarwal, A., Triggs, B.: Tracking Articulated Motion Using a Mixture of Autoregressive Models. In: Proceedings of the 8th European Conference on Computer Vision. LNCS, vol. 3023, pp. 54–65. Springer (2004)
2. Bishop, C.M.: Pattern Recognition and Machine Learning, vol. 4 (2006), http://www.library.wisc.edu/selectedtocs/bg0137.pdf
3. Bobick, A.F., Davis, J.W.: The recognition of human movement using temporal templates. Pattern Analysis and Machine Intelligence, IEEE Transactions on 23(3), 257–267 (2001), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=910878
4. Breiman, L.: Random forests. Machine learning 45(1), 5–32 (2001)
5. Chang, J.Y.: Nonparametric Gesture Labeling from Multi-modal Data. In: European Conference on Computer Vision (ECCV) 2014 Chalearn Workshop. Zurich (2014)
6. Chen, G., Clarke, D., Weikersdorfer, D., Giuliani, M.: Multi-modality Gesture Detection and Recognition With Un-supervision , Randomization and Discrimination. In: European Conference on Computer Vision (ECCV) 2014 Chalearn Workshop. Zurich (2014)
7. Erol, A., Bebis, G., Nicolescu, M., Boyle, R.D., Twombly, X.: Vision-based hand pose estimation: A review. Computer Vision and Image Understanding 108(1-2), 52–73 (Oct 2007), http://linkinghub.elsevier.com/retrieve/pii/S1077314206002281
8. Escalera, S., Baró, X., Gonzàlez, J., Bautista, M.A., Madadi, M., Reyes, M., Ponce, V., Escalante, H.J., Shotton, J., Guyon, I.: ChaLearn Looking at People Challenge 2014: Dataset and Results. In: ECCV Workshop. Zurich (2014)
9. Evangelidis;, G., Singh;, G., Horaud, R.: Continuous gesture recognition from articulated poses. In: European Conference on Computer Vision (ECCV) 2014 Chalearn Workshop. Zurich (2014)
10. Kuznetsova, A., Leal-Taixe, L., Rosenhahn, B.: Real-time sign language recognition using a consumer depth camera. In: ICCV13 (2013)
11. Liang, B., Zheng, L.: Multi-modal Gesture Recognition Using Skeletal Joints and Motion Trail Model. In: European Conference on Computer Vision (ECCV) 2014 Chalearn Workshop. pp. 1–16. Zurich (2014)
12. Mitra, S., Acharya, T.: Gesture Recognition: A Survey. IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews) 37(3), 311–324 (May 2007), http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=4154947
13. Monnier, C., German, S., Ost, A.: A Multi-scale Boosted Detector for Efficient and Robust Gesture Recognition. In: European Conference on Computer Vision (ECCV) 2014 Chalearn Workshop. Zurich (2014)
14. Mori, G.: Max-margin hidden conditional random fields for human action recognition. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 872–879. IEEE (2009)
15. Neverova, N., Wolf, C., Taylor, G.W., Nebout, F.: Multi-scale deep learning for gesture detection and localization. In: European Conference on Computer Vision (ECCV) 2014 Chalearn Workshop. Zurich (2014)
16. Peng, X., Wang, L.: Action and Gesture Temporal Spotting with. In: European Conference on Computer Vision (ECCV) 2014 Chalearn Workshop. Zurich (2014)
17. Pigou, L., Dieleman, S., Kindermans, P.j., Schrauwen, B.: Sign Language Recognition Using Convolutional Neural Networks. In: European Conference on Computer Vision (ECCV) 2014 Chalearn Workshop. Zurich (2014)

18. Poppe, R.: A survey on vision-based human action recognition. Image and Vision Computing 28(6), 976–990 (2010)
19. Rabiner, L., Juang, B.: An introduction to hidden Markov models. ASSP Magazine, IEEE (1986), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1165342
20. Rautaray, S.S., Agrawal, A.: Vision based hand gesture recognition for human computer interaction: a survey (2012)
21. Schmldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local SVM approach. In: Proceedings - International Conference on Pattern Recognition. vol. 3, pp. 32–36 (2004)
22. Sempena, S., Maulidevi, N.U., Aryan, P.R.: Human action recognition using Dynamic Time Warping. Proceedings of the 2011 International Conference on Electrical Engineering and Informatics pp. 1–5 (2011)
23. Sergio Escalera, Gonzàlez, J., Baró, X., Reyes, M., Lopés, O., Guyon, I., Athitsos, V., Escalante, H.J.: Multi-modal Gesture Recognition Challenge 2013: Dataset and Results. In: Chalearn Multi-Modal Gesture Recognition Workshop, International Conference on Multimodal Interaction, ICMI (2013)
24. Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: CVPR. vol. 2 (2011), http://www.stat.osu.edu/ dmsl/BodyPartRecognition.pdf
25. Starner, T., Pentland, A.: Real-time american sign language recognition from video using hidden markov models. Computer Vision, 1995. Proceedings., (1995), http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=477012
26. Suarez, J., Murphy, R.R.: Hand gesture recognition with depth images: A review. In: Proceedings - IEEE International Workshop on Robot and Human Interactive Communication. pp. 411–417 (2012)
27. Sullivan, J., Carlsson, S.: Recognizing and Tracking Human Action. Computing 1(100), 629–644 (2002), http://www.springerlink.com/index/tl5m8dnjjurrmwef.pdf
28. Wachs, J.P., Kölsch, M., Stern, H., Edan, Y.: Vision-based hand-gesture applications (2011)
29. Wu, D., Shao, L.: Deep Dynamic Neural Networks for Gesture Segmentation and Recognition. In: European Conference on Computer Vision (ECCV) 2014 Chalearn Workshop. Zurich (2014)
30. Yamato, J., Ohya, J., Ishii, K.: Recognizing human action in time-sequential images using hidden Markov model. In: Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on. pp. 379–385 (1992)