A Non-temporal Approach for Gesture Recognition Using Microsoft Kinect

Mallinali Ramírez-Corona, Miguel Osorio-Ramos, and Eduardo F. Morales

Instituto Nacional de Astrofísica, Óptica y Electrónica
Luis Enrique Erro No. 1, Sta. Ma. Tonantzintla, Puebla, 72840, México
{mallinali.ramirez,mjoramos,emorales}@inaoep.mx

Abstract. Gesture recognition has become a very active research area with the advent of the Kinect sensor. The most common approaches for gesture recognition use temporal information and are based on methods such as Hidden Markov Models (HMM) and Dynamic Time Warping (DTW). In this paper, we present a novel non-temporal alternative for gesture recognition using the Microsoft Kinect device. The proposed approach, Recognition by Characteristic Window (RCW), identifies, using clustering techniques and a sliding window, distinctive portions of individual gestures which have low overlapping information with other gestures. Once a distinctive portion has been identified for each gesture, all these sub-sequences are used to recognize a new instance. The proposed method was compared against HMM and DTW on a benchmark gesture's dataset showing very competitive performance.

Keywords: Machine Learning, Gesture Recognition, Kinect.

1 Introduction

Advances in computer vision technology provide us with a large number of tools that give us different types of information, making the data manipulation and extraction easier and more precise. A trending device is the Kinect sensor, a technology developed by Microsoft mainly for movement recognition and tracking. It integrates an RGB camera, a depth sensor consisting of an infrared laser projector, and a multi-array of microphones. The Kinect sensor has triggered an increased interest in gesture recognition.

Most gesture recognition systems use temporal information for building their models and for classifying new gestures. Common techniques include Hidden Markov Models (HMM) and Dynamic Time Warping (DTW). The rationale is that taking into account the temporal information from the gesture a better classifier can be build.

In this paper, we take an alternative approach where we train a classifier using "static" information. The advantage is that there is a large number of off-the-shelf robust algorithms that can be directly applied.

Our approach, Recognition by Characteristic Window (RCW), is based on the idea that for each gesture there is a sub-sequence of frames (window) distinct

J. Ruiz-Shulcloper and G. Sanniti di Baja (Eds.): CIARP 2013, Part II, LNCS 8259, pp. 318–325, 2013.

from all other gestures. In this paper, we implement a novel approach that scans a gesture with a sliding window to find, using clustering, a distinctive sub-sequence of that gesture. The generated windows for each gesture are used as input to a classifier to recognize new instances.

We trained different classifiers and compared different classification policies. Our proposed method was also compared against Dynamic Time Warping (DTW) and Hidden Markov Models (HMM) on a benchmark dataset. It is shown that RCW obtained very competitive results when compared against DTW and HMM models.

The remainder of the paper is organized as follows. Section 2 summarizes the most relevant related work for this research. Section 3 describes how the data is pre-processed to obtain new attributes which are robust to translations and rotations. In Section 4 our method is described, Section 4.1 describes the clustering phase of the method where the best windows are found for each gesture and Section 4.2 explains the way the classifier is trained and how the classification is produced. Section 5 describes the performed experiments and results and Section 6 provides conclusions and future research directions.

2 Related Work

Several approaches have been recently proposed for gesture recognition using the Kinect sensor. Kurakin et al. [5] propose a real-time system for hand-gesture recognition using an action graph which shares similar robust properties with standard HMM.

Raptis et al. [6] propose a real-time dance gesture recognition system based on an angular skeleton representation, and a cascaded correlation-based max-likelihood multivariate classifier that takes into account that dancing adheres to a canonical time-base to simplify the template matching process. It uses a space-time contract-expand distance metric to compare the input with an oracle (the ideal movement).

Biswas and Basu [1] propose a method to recognize human gestures using the Kinect® depth camera. First they isolate the human figure from the background and create a region of interest (ROI) by placing a grid on the extracted foreground, the gesture is parametrized using depth variation and motion information content of each cell of the grid.

Wu et al. [4] propose an actionlet ensemble model to represent each action and to capture the intra-class variance. An actionlet is a particular conjunction of the features for a subset of the joints that are important for each gesture. They also add new features called local occupancy pattern (LOP), these features are robust to noise, invariant to translational and temporal misalignment, and capable of characterizing both the human motion and the human-object interactions

Yang et al. [7] choose 3-dimensional feature vector for 3D gesture recognition from consecutive hand coordinates in a spherical coordinate. They propose a hand tracking algorithm that detects a moving object, if it moves like a wave

motion the algorithm decides the object is a hand. Gestures are recognized by a HMM using Baum-Welch algorithm to estimate the parameters.

Carmona and Climent [2] discussed about the best technique for hand gesture recognition: HMM or DTW using Kinect® skeleton. The first step in gesture recognition is the selection of the features; usually, these features are location, orientation and velocity. For HMM they used Baum-Welch algorithm to find the model that best describes the spatio-temporal dynamics of each gesture, the probability of the gesture produced by each HMM is evaluated using Viterbi algorithm. DTW calculates the distance between two signals, thus they used a k-NN classifier to determine which is the most likely class. They obtained best results in their dataset using DTW.

Unlike previous approaches, we employ traditional classifiers using a distinctive part of each gesture.

3 Preprocessing

The performance of gestures by a user can be done at different distances and from different orientation angles. In this paper, we transformed the raw data produced by the joints of the "skeleton" generated by the Kinect, into a scheme invariant to translation and rotation. In particular, we simplified the method presented in [6], that transforms the data from joint points to angles. Our approach computes the angles between three consecutive joints (e.g. wrist-elbow-shoulder), using the cosine formula (1). This formula gets the angle between two vectors, in this case represented by the joints coordinates.

$$\cos\theta = \frac{a \cdot b}{\|a\| \cdot \|b\|} \tag{1}$$

From the twenty joint coordinates produced by the "skeleton" from the Kinect, only nine were selected as the most descriptive joints. These selected joints were used to obtain the relative angles between consecutive joints, reducing then the attributes from 20×3 (points x, y and z of each joint) to 9, producing a representation invariant to translation and rotation. The attributes are shown in Figure 1.

4 Recognition by Characteristic Window

Our method, RCW, is divided in two phases, the first (Section 4.1) finds the most representative section for each gesture and the second (Section 4.2) classifies the frames and returns a prediction based on the information obtained in the first phase.

4.1 Clustering

Given a set of gestures $G = \{g_1, g_2, ..., g_k\}$, our hypothesis is that for each gesture there exists a sub-sequence of frames that is different from any subsequence of all the other gestures. We implemented a method to find that subsequence through clustering. The algorithm proceeds as follows: we take a sliding

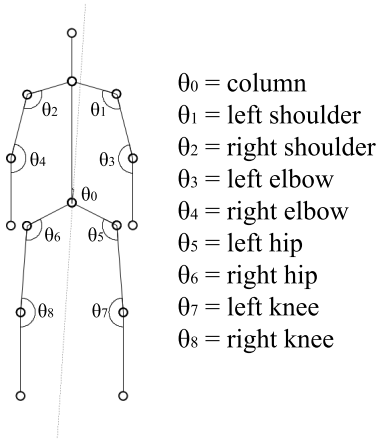

Fig. 1. Skeleton joints showing the most descriptive angles

window with a predefined size relative to the number of frames (percentage) of a gesture. Given a particular window (set of instances) of one gesture (g_i) and the complete sequences of all the other gestures, we run k-means with k equal to the number of classes (different gestures) we want to recognize. If the clustering method generates a cluster whose elements are mostly samples from the selected window, this is returned as a sub-sequence that is distinctive enough from the other gestures. For each sliding window we use the f-score (see Equation (2)) to evaluate how distinctive is this window with respect to the other gestures.

$$F1 = 2 \cdot \left(\frac{precision + recall}{precision \cdot recall} \right) \tag{2}$$

4.2 Classification

We trained a classifier using either the complete sequences of the gestures or using only the distinctive identified windows for all the gestures, with the nine angles as attributes (Section 3). The trained classifier is used to assign one of the possible gestures to each frame of a testing gesture.

For testing, we implemented two decision policies:

1. We classify each frame from the testing gesture and return the class of the longest set of consecutive frames classified equally. We call this policy, *longest sequence* (LS).
2. The second policy takes advantage of the positions of the identified windows in the clustering process. The testing gesture is evaluated only in the windows that were selected during the clustering phase. For each window we obtain a percentage of coincidence and return the class belonging to the window with the highest value. We call this policy *window verification* (WV).

Algorithm 1. RCW clustering algorithm, where *windowSizes* are a pre-defined set of percentages of windows to tried, *step* is the percentage of how much a window is slid each time and *currentScore* is a temporal variable that stores the accuracy of the clusterization for an specific window.

Require: $G, windowSizes, step \geq 0$
Ensure: $bestWindowSize, bestWindowPosition : \forall g_i \in G$
for all $g_i \in G$ **do**
  $maximumScore \leftarrow -1$
  for all *size* such that $size \in windowSizes$ **do**
    for $position = 0$ to $position \leq (100-\text{step})$ **do**
      $datasetToCluster \leftarrow (\forall frame | frame \in window(g_i, position, size) \bigcup (\forall frame \in g_k | g_k \neq g_i)$
      $currentScore \leftarrow eval(kNN(datasetToCluster))$
      if $currentScore > maximumScore$ **then**
        $maximumScore \leftarrow currentScore$
        $bestWindowSize \leftarrow size$
        $bestWindowPosition \leftarrow position$
      end if
      $position \leftarrow$position + $step$
    end for
  end for
end for

Since the windows are selected as percentage of the gesture, its use still works with longer or shorter gesture instances.

5 Results

We tested RCW on the dataset *Microsoft Research Cambridge-12* (MSRC-12) which consists of 594 sequences of movements of an skeleton characterizing the human body. These sequences were collected from 30 persons doing 12 gestures having a total of 6244 instances. The set of files contains the tracking of 20 joints presented as points in the space $< x, y, z >$; each of these files contains around ten instances per gesture performed one after the other. The gestures can be categorized into two abstract categories: Iconic gestures, those that imbue a correspondence between the gesture and the reference, and Metaphoric gestures, those that represent an abstract concept. For the experiments we used the subset of iconic gestures.

- Gesture 2: *Crouch or hide* [500 instances]
- Gesture 4: *Put on night vision goggles* [508 instances]
- Gesture 6: *Shoot a pistol* [511 instances]
- Gesture 8: *Throw an object* [515 instances]
- Gesture 10: *Change weapon* [498 instances]
- Gesture 12: *Kick* [502 instances]

The accuracy from the clustering and the classification phases were measured using the f-score (see Equation (2)).

In the training phase, different window sizes and positions were tested for each gesture. We slid each window 2% of the total gesture each time. For the evaluation phase we used 10-cross fold validation.

Table 1 shows the different values in terms of window size of the precision results and the window position for the six gestures. The best results are marked in bold face. Figure 2 depicts the best windows found for the training data.

Table 1. Best window starting and window length for each gesture, where AC is Accuracy (%) and WP is Best Window position in percentage

Gesture	Window size							
	10%		15%		20%		25%	
	AC	WP	AC	WP	AC	WP	AC	WP
Duck	**96.96**	**90**	93.44	84	89.77	80	82.03	74
Googles	29.29	6	42.58	4	49.83	2	**60.41**	**0**
Shoot	33.22	90	41.80	84	47.65	80	**49.91**	**74**
Throw	15.05	76	20.58	74	24.78	70	**27.87**	**68**
Change Weapon	14.93	90	20.34	0	**32.08**	**78**	27.68	0
Kick	8.04	84	9.75	48	**30.12**	**80**	15.64	74

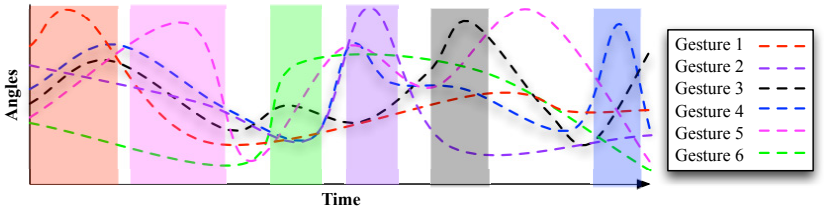

Fig. 2. Graphical representation of the sections found by the clustering phase, the colored columns represent an example of the most representative part of each gesture

Once the distinctive windows were identified for the gestures, we tried four different classifiers from Weka: Naïve Bayes, SVM, C4.5 and Random Forest. After training the classifier, the classification of new gestures was carried out using *longest sequence* (LS) and *window verification* (WV) policies.

We performed tests with two training datasets:

1. Pre-processed dataset (PP-MSRC), which uses the whole transformed sequence of frames to train a classifier.
2. Pre-processed dataset which uses only the frames that belong to the window for each example of gesture (W-MSRC).

The obtained results are shown in Table 2 using a 10-fold cross-validation; the best results are marked in bold face. The overall best is marked with an asterisk.

As can be seen from the results, considering only the distinctive window for evaluation (the WV policy) increases the accuracy in all cases.

Table 2. Obtained results with different classification schemes for each dataset using Longest Sequence (LS) and Window Verification (WV) policies

Classifier	PP-MSRC						W-MSRC					
	LS			WV			LS			WV		
	Prec	Rec	Acc	Prec	Rec	Acc	Prec	Rec	Acc	Prec	Rec	Acc
C4.5	80.13	80.34	80.23	89.35	89.35	**89.35**	67.48	69.70	69.90	85.51	86.70	86.10
SVM	62.09	63.45	62.76	83.71	83.86	83.78	39.73	61.12	48.15	90.61	91.11	**90.86**
Naïve Bayes	48.26	58.05	52.74	80.78	82.13	81.45	41.15	62.15	49.52	90.65	91.24	**90.94**
Rand. Forest	85.79	86.18	85.99	91.82	91.84	**91.82***	75.98	77.39	76.67	91.10	91.85	91.47

RCW (WV policy, PP-MSRC dataset and Random Forest classifier) was compared against two typical methods of gesture recognition: DTW and HMM. As in the previous experiment the accuracy was measured with f-score. The experiment was evaluated using 10-fold cross-validation.

A HMM for each gesture was learned using the Baum-Welch algorithm, then the probability for the frames sequence is computed using Viterbi algorithm, the returned prediction is the one with the best predicted probability. We tried with different number of hidden nodes and report only the best results, that were obtained using three nodes.

To calculate the most probable gesture using DTW, the distance to a subset of examples of each of the gestures (50 examples for this experiment) was computed using the mean of the calculations, the predicted gesture was the one where the distance was smaller.

The results of these experiments are shown in Table 3. A paired t-test was carried out to find statistical significance in the results (marked with an arrow). As can be seen RCW is very competitive against temporal-based approaches and it is statistically better (with 95% of confidence value) against DTW. Apart from that, the small difference between HMM and RCW shown in the results suggests that RCW is a suitable substitute of HMM for this specific problem.

Table 3. Comparing accuracy of RCW against DTW and HMM (percentage)

	Overall	Duck	Googles	Shoot	Throw	Ch. Weapon	Kick
DTW	82.74 ↓	97.11±1.26	71.83±0.89	97.14±0.79	76.89±1.53	75.55±2.17	55.74±2.56
HMM	91.81	97.73±1.42	88.06±1.30	87.45±2.66	90.14±2.41	90.82±0.75	93.95±2.39
RCW	**91.82**	95.49±1.89	85.25±1.88	93.71±1.43	95.43±1.24	82.07±3.2	97.71±0.75

↓ Statistically inferior result with respect to RCW.

6 Conclusions

This article described a novel non-temporal approach to classify gestures from information obtained by a Kinect sensor. RCW identifies distinctive portions of each gesture using a sliding window and a clustering technique. Each window is given as input to a classifier and a new gesture is classified using also a window-based approach. It is shown that our non-temporal approach is very competitive against standard temporal approaches normally used for gesture recognition. As future work we would like to perform more tests involving a larger set of gestures. We would also like to combine more than one discriminatory window for each gesture to improve performance.

References

1. Biswas, K.K., Basu, S.K.: Gesture recognition using microsoft kinect. In: 2011 5th International Conference on Automation, Robotics and Applications (ICARA), pp. 100–103 (2011)
2. Carmona, J.M., Climent, J.: A performance evaluation of hmm and dtw for gesture recognition. In: Alvarez, L., Mejail, M., Gomez, L., Jacobo, J. (eds.) CIARP 2012. LNCS, vol. 7441, pp. 236–243. Springer, Heidelberg (2012)
3. Fothergill, S., Mentis, H., Kohli, P., Nowozin, S.: Instructing people for training gestural interactive systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2012, pp. 1737–1746. ACM, New York (2012)
4. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012, pp. 1290–1297. IEEE Computer Society, Washington, DC (2012)
5. Kurakin, A., Zhang, Z., Liu, Z.: A real time system for dynamic hand gesture recognition with a depth sensor. In: 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), pp. 1975–1979 (2012)
6. Raptis, M., Kirovski, D., Hoppe, H.: Real-time classification of dance gestures from skeleton animation. In: Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation, SCA 2011, pp. 147–156. ACM, New York (2011)
7. Yang, C., Jang, Y., Beh, J., Han, D., Ko, H.: Gesture recognition using depth-based hand tracking for contactless controller application. In: 2012 IEEE International Conference on Consumer Electronics (ICCE), pp. 297–298 (2012)
8. Zhang, H., Du, W., Li, H.: Kinect gesture recognition for interactive system (2012)