Frontal-Standing Pose Based Person Identification Using Kinect

Kingshuk Chakravarty and Tanushyam Chattopadhyay

Innovation Lab, Tata Consultancy Services Ltd., Kolkata, India
{kingshuk.chakravarty,t.chattopadhyay}@tcs.com

Abstract. In this paper we propose a person identification methodology from frontal standing posture using only skeleton information obtained from Kinect. In the first stage, features related to the physical characteristic of a person are calculated for every frame and then noisy frames are removed based on these features using unsupervised learning based approach. We have also proposed 6 new angle and area related features along with the physical build of a person for the supervised learning based identification. Experimental results indicate that the proposed algorithm is able to achieve 96% recognition accuracy and outperforms all the stat-of-the-art methods suggested by Sinha et al. and Preis et al.

1 Introduction

Biometric Person identification in an image or video is of crucial importance and it is critical to determine the presence of a particular person for applications where automatic person recognition is a key enabler such as security and surveillance, elderly people care etc. People are mainly identified based on different physical and behavioral features e.g. iris, fingerprint, speech, face etc. But biometric identification based on these modalities are intrusive as they require direct human interaction. In addition, extracting face, iris or fingerprint characteristic from a large distance or in poor lighting condition are indeed a challenging job. This paper aims at developing a novel person identification algorithm based on only physical build characteristic of a person. As the overall physical structure of a person can be extracted at a large distance and it is very difficult to imitate or hide, the method has clear advantages over the exiting ones. One approach to determine physical characteristics of person is to capture skeleton joint co-ordinates over time. But to accomplish this, we need to have multiple positional cameras to obtain skeleton information. Fortunately, Microsoft provides us a 3D (RGB-D) sensor platform called "Kinect" which can directly provide the 20 skeleton joint co-ordinates. As we are only using skeleton information instead of video or RGB-D image, our proposed method can properly ensure user's privacy and security issue.

After obtaining the skeleton information, the physical build (features) of a person like body dimensions, height, length of two legs, arms etc. can be easily computed from the data. As human being is capable of identifying a person from

M. Kurosu (Ed.): Human-Computer Interaction, Part II, HCII 2014, LNCS 8511, pp. 215–223, 2014.

his/her physical or structural build, any standard statistical learning method (supervised or unsupervised) can be used to map these unique features to a particular object class repressing a person. It needs to be mentioned that person identification using skeleton information already exists in the literature. Preis et al. [1] used physical build of a person like height, length of torso etc. and dynamic gait information like step length and velocity for person identification from constrained side walking pattern. Adrian et al. [2] proposed an unsupervised learning (K-Means) based identification algorithm based on dynamic angular information related to the gait pattern using Kinect and obtained 43.6% accuracy for 4 subjects. Manual gait cycle extraction used by Adrian et al. is not possible in any realtime system. While Naresh et al. [3] tried to model arbitrary walking pattern using only physical build characteristics, Sinha et al. [4] proposed a robust pose and sub-pose based modeling approach for the same. But none of them tried to identify a person from their only static posture using skeleton information obtained from Kinect.

For some applications like TV viewership monitoring or monitoring blackboard activity in school or college, it is very much important to recognize a person from his/her static posture. The static posture may be interpreted as standing, sitting, lying or anything else. To address the above usecases, this paper aims at proposing a novel framework for supervised learning based person identification using only frontal standing pose. We have done the frame level performance analysis as well as comparison our proposed method with respect to existing solution. The key contributions of the paper are given below

- Frontal standing pose based person identification using skeleton data.
- New area and angle related features are proposed for person identification.
- Noisy skeleton data removal using physical characteristic of the person.
- Multiclass Support Vector Machine (SVM) with RBF kernel [2] is employed for supervised learning based person identification.

Rest of the paper is organized as follows. The proposed methodology is described in the Section 2. The detailed results are provided in Section 3 followed by conclusion in Section 4.

2 Proposed Methodology

In this paper, we have presented a frontal standing posture based person identification using only skeleton data obtained from Microsoft Kinect sensor [5]. Kinect provides human skeleton data for 20 skeleton joints at 25 frames per second in real time. The framework shown in the flowchart (Fig. 1) has mainly five modules as given below.

- Acquisition of skeleton data
- Feature extraction
- Noisy frames removal
- Decision Making using Supervised Learning

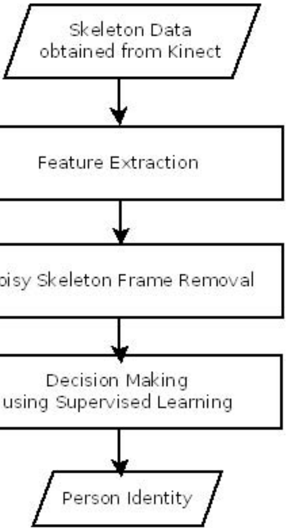
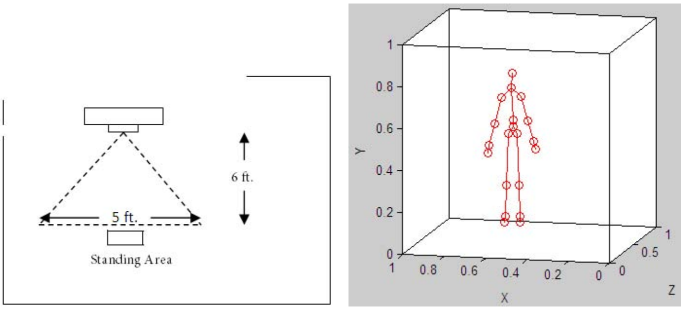

Skeleton Data obtained from Kinect
Feature Extraction
Noisy Skeleton Frame Removal
Decision Making using Supervised Learning
Person Identity

Fig. 1. Flowchart of Our Proposed Algorithm

2.1 Acquisition of Skeleton Data

For data-capture we have marked a fixed position in front of the Kinect where an individual is requested to stand for training and testing. We have used the 20 joints of skeleton data for a person captured at 30 frames per second in frontal-standing posture (figure 2). Each joint consists of 3D world co-ordinates i.e. {x,y,z} tuple in meters considering the Kinect camera as origin of the world coordinate system. We have used Microsoft SDK version 1.5 for the data-capture.



(a) Recording Setup

(b) 3D world co-ordinate points for 20 joints of a subject

Fig. 2. Kinect experimental setup

2.2 Feature Extraction

Feature extraction is one of the main steps for any machine learning based approach. In this case, we have tried to model physical build or structure of a person using a feature vector $\boldsymbol{f}$ which includes

Area Feature ($\boldsymbol{f}_{area}$) - Area occupied by the polygon formed by the joints 1. shoulder left, shoulder right and shoulder center, and 2. hip left, hip center and hip right are unique features for any individual because they do not vary with pose or time. We have considered both of these as one of our candidate features i.e. $\boldsymbol{f}_{area} \in \mathbf{R^2}$. If co-ordinates of i^{th} $(i \in 20)$ joint is (x_i, y_i), then the area A enclosed by the N joints can be computed using eqn. 1

$$A = \frac{1}{2}\sum_{i=0}^{N-1}(x_i * y_{i+1} - x_{i+1} * y_i) \tag{1}$$

Angle Feature ($\boldsymbol{f}_{angle}$) - We have calculated four angles mentioned below

1. angle between shoulder left, shoulder center and spine.
2. angle between shoulder right, shoulder center and spine.
3. angle of the shoulder center and spine with respect to the vertical axis.
4. angle between hip left, hip center and hip right.

As these four angles are unique for any individual and also invariant to pose or posture, we have used the same as one of the candidate features ($\boldsymbol{f}_{angle} \in \mathbf{R^4}$).

Features Related to the Physical Build - We consider height, length of upper and lower legs, length of arms etc to describe physical build of a person. For this, we have used all the static features ($\boldsymbol{f}_{static} \in \mathbf{R^{12}}$) mentioned in [1].

We have used feature vector $\boldsymbol{f} = \{\boldsymbol{f}_{area}, \boldsymbol{f}_{angle}, \boldsymbol{f}_{static}\} \in \mathbf{R^{18}}$ for training and testing using SVM.

2.3 Noisy Frames Removal

The skeleton data obtained from Kinect is itself very much noisy. So noise cleaning is required to achieve good recognition accuracy. The noisy frames are identified and removed based on the static feature vector $\boldsymbol{f}_{static}$. For noise cleaning, we assume that the mutual Euclidean distance between two joints should not vary with time. So if it varies significantly from one frame to another, we mark those frames as noisy frames and remove all the features ($\boldsymbol{f}$) corresponding to those frame for further processing. In our implementation, this is done using unsupervised clustering algorithm [4]. The cluster or group with sparsely distributed points (representing the static feature vectors) is identified as a noisy one, and the frames associated with the sparsely distributed static feature vectors are referred to as the noisy frames. The cluster centers are initialized in the following manner.

– We compute A histogram of B bins on $\boldsymbol{f}_{static}$ where each bin is defined by (2) where k represents the bin-index, for i^{th} static features and j^{th} frame level data points (D).

$$Bin_k^{ij}, 1 \le k \le B, 1 \le i \le 12, 1 \le j \le D \tag{2}$$

- The bin k^i_{max} containing the maximum number of data points for i^{th} feature is used to calculate the i^{th} dimension of the first center.
- An mean of all the points belonging to the k^i_{max} bin represents the center $(C^i_1)$of the first cluster for the feature i and it can be defined as (3), where P^i is the number of feature points $\in k^i_{max}$. The first center is defined as $\boldsymbol{C_1} \in R^{12}$.

$$C^i_1 = \frac{\sum\limits_{j \in k^i_{max}} \boldsymbol{f}^i_{static_j}}{P^i}, 1 \leq i \leq 12 \tag{3}$$

- The second cluster center $(\boldsymbol{C_2})$ is the data point $(\boldsymbol{f}_{static_j})$ representing the static feature vector that is at a furthest distance with respect to the first center $(\boldsymbol{C_1})$. We have selected $\boldsymbol{C_2}$ based on the initialization of the K-Means++ [6] algorithm.

2.4 Decision Making Using Supervised Learning

We have used multicalss Support Vector Machine as supervised learning algorithm for decision making process.

Given N-class training data in the form of D = $(\boldsymbol{f_1}, y_1)$, $(\boldsymbol{f_2}, y_2)$, $(\boldsymbol{f_3}, y_3)$, ..., $(\boldsymbol{f_n}, y_n)$ where $\boldsymbol{f_i} \in \mathbf{R^n}$ is feature vector representing a class, a supervised learning algorithm [7] [8] requires a function g which maps the input/feature space (X) into decision or output space (Y) g:X $\rightarrow$ Y. Here g is the element of hypothesis space G. Some times g is also expressed as scoring function f(x,y): $X \times Y \rightarrow$ R, such that g is defined as g(x) = $\arg\max_y f(x, y)$. For probabilistic learning model g is defined either by conditional probability g(x) = P(Y|X) or by joint probability model f(x,y) = P(x,y). Empirical risk minimization (ERM) and structural risk minimization (SRM) [9] are commonly used for choosing g and f. In structural risk minimization based approaches the problem of over fitting is prevented by incorporating regularization penalty.

Support Vector Machine (SVM) [10] [11] [12] a very well known supervised learning algorith was first proposed by Vapnik. SVM is developed based on the structural risk minimization [9] principle derived from computational learning theory. SVM separates objects into different classes by defining a hyper plane in multidimensional space. SVM employs mathematical operator ϕ for mapping the training datapoints from input space to higher dimensional space. These mathematical opertor are often referred as Kernel. Then iterative training algorithm is used to define the separating hyperplane in that higher dimensional space by optimizing (minimizing) an error function. Based on the selection of the error function, SVM can also be categorized as i) C-SVM ii) nu-SVM iii) epsilon-SVM regression and iv) nu-SVM regression. For example, C-SVM has the error function

$$e = \frac{1}{2} * w^T w + C \sum_{i=1}^{N} \epsilon_i \tag{4}$$

subject to the constraints:

$$y_i(w^T\varphi(x_i) + b) \geq 1 - \epsilon_i \quad and \quad \epsilon_i \geq 0, \quad i = 1,..,N \quad (5)$$

Though linear hyperplane was originally proposed by Vapnik, but in 1992, Bernhard E. Boser, Isabelle M. Guyon and Vladimir N. Vapnik also modified the kernel function [13] to maximum-margin hyperplanes [14] for building a nonlinear classifier. Various types of kernel functions already exist in the literature for different applications e.g linear Kernel, Radial Basis Function (RBF), polynomial function etc.
Polynomial (homogenius) kernel- k$(x_i, x_j) = (x_i.x_j)^d$
Polynomial (inhomogenius) kernel- k$(x_i, x_j) = (x_i.x_j + 1)^d$
Gaussian radial basis function (RBF) kernel - k$(x_i, x_j) = \exp(-\gamma||x_i - x_j||^2)$, for $\gamma > 0$. Sometimes $\gamma = 1/(2\sigma^2)$, where σ is described as the area of influence occupied by the support vectors over input data space. Several approach had already been proposed for multiclass SVM, few of them include

- one of the class label with respect to rest (one-versus-all)
- between each and every pair of classes (one-versus-one)
- Directed Acyclic Graph SVM [15]
- error-correcting output codes [16]

Crammer and Singer also proposed a multiclass SVM by considering entire classification objective as a single optimization problem rather than dividing it into multiple binary classification problems.

3 Experimental Results

We have taken 10 persons (7 male + 3 female subjects) dataset for training and testing. Initially, a single kinect is positioned at a fixed position to record the skeleton information at a distance of 6 feet from the subject. Then feature vector $\boldsymbol{f} \in \mathbf{R^{18}}$ is extracted at frame level for 10 subjects (A-J). As discussed earlier then we perform the noise removal using $\boldsymbol{f}_{static}$. After removing noisy frames, we store the feature vector $\boldsymbol{f}$ for rest of the frames in a dataset D. The dataset D is used for training model generation using multiclass SVM.
We have done the frame level performance analysis as well as comparison on the basis of F-score (6), which is defined as the harmonic mean of precision and recall. Here N is the number of subjects.

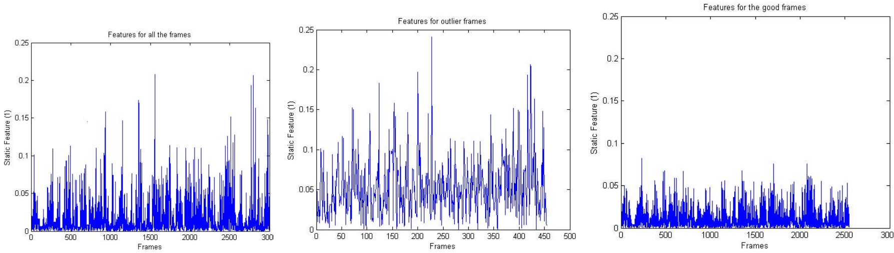
$$Fscore_i = \frac{2 * precision_i * recall_i}{(precision_i + recall_i)} \quad \forall i, 1 \leq i \leq N \quad (6)$$

The performance analysis is done in 2 sections

- Effect of outlier removal
- Comparison with state-of-the-art systems.

3.1 Effect of Outlier Removal

As discussed in the section 2.3, skeleton information obtained from Kinect is very noisy [17]. Euclidean distance based outlier detection algorithm [4] is used to remove noisy frames. It is based on the fact that mutual distance between two physical joints should be constant over frames. Thus if the the joint-distance varies significantly from one frame to another frame, we remove those noisy frames from further processing. K-Means++ [6] algorithm is used for clustering $\boldsymbol{f}_{static}$ into two clusters - one containing noisy frames and other containing clean data. Figure 3 shows the cluster based analysis of noisy skeleton data. Figure 3(a), 3(b) and 3(c) represent the skeleton information for all the frames, for noisy frames and outcome of our proposed noise removal algorithm (i.e. noise clean skeleton data), respectively.



(a) for all frames (b) for noisy frames (c) noise clean data

Fig. 3. Sample static feature for different frames of a subject. The horizontal axis represents different frames and the vertical axis represents the normalized feature values.

3.2 Comparison with State-of-the-Art Systems

Performance evaluation of our proposed system is done at frame level with 30 seconds training data and 20 seconds testing data. A sample confusion matrix for 10 subjects marked as 'A' to 'J' is shown in the table 1. The diagonal entries of the matrix (shaded in grey) indicate correctly identified frames. Performance comparison is also performed with [18], [4] [1]. The results are tabulated in table 2. From table 2, it is very much clear that our proposed algorithm with noise removal technique is able to achieve 95.75% in real time and outperforms all the state-of-the-art methods.

4 Conclusion

Results indicate that our proposed angle and area related features with SVM based classification technique are having good contribution as the recognition accuracy has increased to 96% when only frontal standing pose is used for person identification. We are planing to do Kinect calibration to identify a multiple person in arbitrary poses using supervised learning.

Table 1. Confusion matrix for the proposed algorithm with 10 subjects

Subjects	A	B	C	D	E	F	G	H	I	J
A	2644	0	0	0	0	0	0	0	0	0
B	0	2647	0	0	0	0	0	0	0	0
C	0	17	2297	0	0	0	0	0	0	350
D	0	57	0	2608	0	0	0	0	2	1
E	0	0	0	0	2610	0	0	0	7	0
F	0	0	0	0	0	2654	0	0	0	0
G	0	0	0	0	0	0	2655	0	483	0
H	0	1	0	0	0	0	0	2653	0	0
I	0	0	0	1	0	0	0	0	2653	0
J	0	309	0	4	0	0	0	0	0	2334

Table 2. Performance (F_{score} in %) comparison with [18], [4] and [1]

Our Proposedwith frontal-standing	Using [18]	Using [4]	Using [1]
95.75	56	69	29

References

1. Preis, J., Kessel, M., Werner, M., Linnhoff-Popien, C.: Gait recognition with kinect. In: 1st International Workshop on Kinect in Pervasive Computing (2012)
2. Ball, A., Rye, D., Ramos, F., Velonaki, M.: Unsupervised clustering of people from skeleton data. In: 2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pp. 225–226. IEEE (2012)
3. Kumar, M., Babu, R.V.: Human gait recognition using depth camera: a covariance based approach. In: Proceedings of the Eighth Indian Conference on Computer Vision, Graphics and Image Processing, p. 20. ACM (2012)
4. Sinha, A., Chakravarty, K.: Pose based person identification using kinect. In: 2013 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 497–503. IEEE (2013)
5. Microsoft: Kinect SDK (2012), `http://www.microsoft.com/en-us/kinectforwindows/develop/developer-downloads.aspx` (accessed February 6, 2014)
6. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, pp. 1027–1035 (2007)
7. Mohri, M., Afshin Rostamizadeh, A.T.: Foundations of Machine Learning. The MIT Press (2012)
8. Wikipedia: Supervised learning — wikipedia, the free encyclopedia, `http://en.wikipedia.org/wiki/Supervised_learning/` (accessed February 6, 2014)
9. Vapnik, V.N.: The nature of statistical learning theory. Statistics for engineering and information science. Springer, Berlin (1999)
10. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
11. Cortes, C., Vapnik, N.V.: Support-vector networks. Machine Learning, 20

12. Wikipedia: Support vector machine — wikipedia, the free encyclopedia, `http://en.wikipedia.org/w/index.php?title=Support_vector_machine&oldid=548461902/` (accessed February 6, 2014)
13. Aizerman, M.A., Braverman, E.M.: Rozonoer: Theoretical foundations of the potential function method in pattern recognition learning. Automation and Remote Control 25, 821–837 (1964)
14. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory. ACM (1992)
15. Platt, J.C., Cristianini, N., Shawe-taylor, J.: Large margin dags for multiclass classification. Advances in Neural Information Processing Systems 12(3), 547–553 (2000)
16. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research 2, 263–286 (1995)
17. Newcombe, R.A., et al.: Kinectfusion: Real-time dense surface mapping and tracking. In: 2011 10th IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 127–136. IEEE (2011)
18. Sinha, A., Chakravarty, K., Bhowmick, B.: Person identification using skeleton information from kinect. In: ACHI 2013, The Sixth International Conference on Advances in Computer-Human Interactions, pp. 101–108 (2013)