Background Invariant Static Hand Gesture Recognition based on Hidden Markov Models

Radu-Laurențiu Vieriu[1,2], Ionuț Mironică[3], Bogdan-Tudor Goraș
[1] TrentoRise, Trento, Italy
[2] Dept. of Information Eng. and Computer Science, University of Trento, Italy
[3] LAPI, University Politehnica of Bucharest, Romania

***Abstract* — This paper addresses the problem of Static Hand Gesture Recognition (SHGR) and proposes a fast yet simple solution based on Discrete Hidden Markov Models (DHMMs) that use features extracted from the hand contours. In addition to previous work, the use of depth information ensures robustness to the overall system, making it background invariant. Experiments carried on a challenging noisy dataset reveal the superior discriminating as well as generalizing abilities of statistical models, when compared to *state-of-the-art* methods.**

I. Introduction

Gesture recognition is a major player in the field of computer vision, regardless of the nature of the gestures (face, hand, body). It aims at inferring behavioral cues and uses this information to facilitate human behavioral understanding or different aspects of human-computer interaction. When it comes to hand gestures, such aspects may refer to browsing through menus, interpreting different messages or posting various commands to intelligent systems.

Recent developments in depth sensing, along with the fact that technologies that record this type of data have become more affordable (e.g. MS Kinect [1], Asus Xtion [2]), opened new perspectives in solving the inherent 2D problems, like loss of valuable information due to projection, occlusions, background extraction etc.

A successful hand recognition system requires a fruitful association between discriminative features that are fast and easy to extract and efficient classifiers that are able to value the chosen features. The literature offers various combinations, each having strengths and weaknesses. High level features are preferred because of their compact representation and ease of describing gestures from a structural approach. In some studies [3] [4] [5] anchor points, color gloves and other sensors are used to extract different features, but these methods are rather invasive and reduce considerably the naturalness of gestures. More recently, in [6] fingertips are obtained by analyzing curvature segments extracted from contours, using an approach similar to that in [7]. While the features themselves seem promising, they do require appropriate classification algorithms in order to use the considerable amount of information they provide. Low level features (e.g. appearance cues, contours, edges) on the other hand are certainly more efficient in this regard and, according to [8], are present in vast majority of studies.

Generally, hand gestures can be divided into two main categories: static ones, for which only the configuration and posture of the hand are of interest and dynamic gestures, where more attention is paid to the trajectory described by the hand over time. While in the static case the recognition can be performed using standard pattern recognition tools [9] [10] [11] [12], the dynamic domain requires techniques that use temporal information like time-compressing templates [13] or Hidden Markov Models [14] [5] [15]. One intensely studied application of HGR is sign language recognition (SLR), which treats both static as well as dynamic problems. There are many published papers on SLR, some of the most relevant being the work of Sarkar et al. [16] [17] [18] and Sclaroff et al. [19] [20]. As stated in [21], HMMs are often used for dynamic HGR and this is mostly attributed to their success on speech processing.

Depth sensors have also been included in gesture recognition systems, as a simple and convenient way of isolating the object of interest from the background. Many recent studies [22] [23] use such device for this purpose.

To our knowledge, this is the first work to employ HMMs for hand posture recognition using a Kinect sensor. We propose a robust and efficient approach using both depth information as well as the color data obtained from the sensor. The robustness spans several areas, making the resulted system immune to background properties and invariant to changes in scale and small rotations. Moreover, each individual frame is processed (i.e. feature extraction and classification) in less than 32 ms, fast enough for most applications that pose real time constraints.

The remainder of the paper is organized as follows: Section II presents the proposed approach, while in Section III we report the experimental results. Finally, Section IV concludes the paper.

II. Proposed approach

This work describes a background invariant Discrete Hidden Markov Model based recognition system where hand gestures are modeled using Markovian chains along with observation sequences extracted from the contours of the gestures. The idea of using HMMs for static hand postures in the first place came from the dynamic domain. It is well known that Markov Models are suited for dynamic processes, especially when one can point to parts of these processes that share similar statistical properties. Imagining such a process in the static field can extend the conventional use of the HMM. One can easily isolate different parts like finger segments by exploring the gesture's contour. Therefore, it makes sense to use HMMs, even though there are no gesture dynamics involved. The success of the statistical models relies heavily on the ability of highlighting all the contour segments "seen" in the training stage.

As pointed in [24], DHMMs work very well with "clean" (background dependent) data, resulting in high recognition accuracies. As will be shown in this paper, they also cope well with more challenging samples, as ones obtained from a cheap sensor like Kinect. The processing flow starts with synchronizing the two channels, followed by small adjustments applied to the images in order to overlap and fit seamlessly. This was solved by a linear mapping between the depth frame and the corresponding colored one.

A. Isolating the gestures

Assuming that the hand is the only object closest to the sensor, the isolating stage follows a fairly easy routine by using two types of segmentation: one is applied to the depth image and consists of an adaptive threshold that separates the close range field from the background. The other consists of a simple and fast skin color detector, applied to the RGB frame.

The final isolated object is obtained by combining the two segmentation images in a logical sum, as can be seen in Fig. 1. This procedure ensures that the hand is extracted from the scene even though it overlaps other skin colored background objects (like for instance the face).

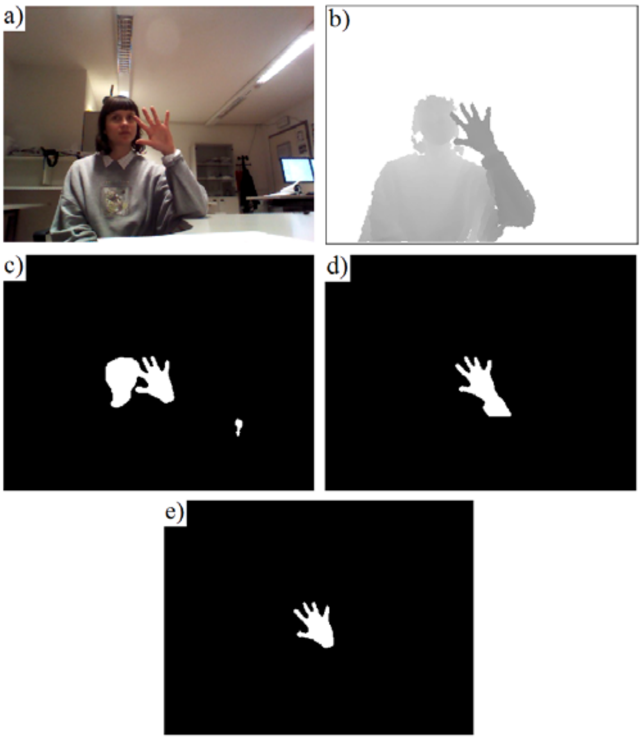

Figure 1. Example of segmenting one frame from the test set: a) – original RGB image, b) – corresponding depth frame, c) – skin color segmentation result, d) – depth segmentation result and e) – logical summation of c) and d)

B. Feature extraction

After isolating the hand from the background, a median filtering is performed for smoothing the boundaries and then these are extracted by tracing the outline of the object in the binary image. In case of multiple unlinked objects, only the one with the longest boundary is kept.

Given the contour, a starting point is adopted (in this case the most southern and then eastern is chosen) and afterwards the curve is re-sampled as to meet a specified length. Besides aligning all the contours to the same length, the re-sampling process (applied to the 2D points that describe the boundary) ensures a most welcome robustness to changes in scale. This means that, regardless of the size of the hand gesture, the extracted contour will always have the same dimension and, after the next processing step, roughly the same feature values will be obtained. The 2D points are used to compute the angles that the tangent to the contour determines with the horizontal axis during the exploring process. Finally a rescaling stage maps the angle values to the dictionary symbols of the DHMM, resulting in a T sized observation sequence, as seen in (1).

$$O=\{o_t, t=\overline{1,T}\} \qquad (1)$$

The feature extraction process is briefly outlined in Fig. 2. Given the re-sampling process and that the angles are calculated with respect to a reference set to the aforementioned starting point i.e. on the hand itself, the resulted features are robust also against changes in scale and translation. This offers the freedom of naturally moving the hand inside the frame. Additional robustness to small changes in rotation along the normal vector to the camera plane will be incorporated by the models.

C. Training the models

Each gesture is associated a left-to-right Hidden Markov Model that "learns" the class specifics during training and stores them in the form of probabilities and probability distributions. The particular structure of HMMs used here allows transitions either from one state to itself or to the next one. This is consistent with the exploring process, even though observation symbols may have repeatable values in different parts of the contour.

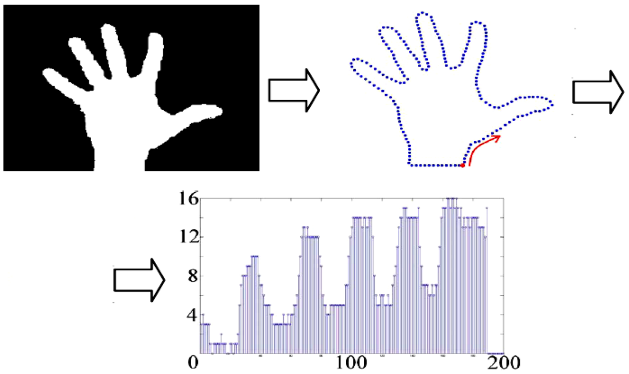

Figure 2. Feature extraction starting from segmented image, followed by extracting the contour and ending up with the angle sequence made by the tangent to the contour with the horizontal axis.

HMMs are oriented graphs described by three main parameters: $A = \{a_{ij}\}$, the state transition matrix, which holds the probabilities of jumping from one state to another, $B = \{b_j(k)\}$, the state probability distribution, responsible with modeling the observation symbols associated to each state j from the graph and $\pi = \{\pi_j\}$, the initial state probability vector, that models the uncertainty of having a given state as the first one. The training process assumes an initial segmentation of the training sequences, in order to define and initialize the state parameters of the graph. This segmentation was performed by observing that each gesture could be represented by a finite number of concatenated segments. These segments were chosen in order to group observation symbols that share similar values. In this manner, not only the number of states for each model was fixed, but also the premises for initializing all other parameters were obtained.

In the second part of the training stage, the model parameters are tuned by re-estimation formulas, in order to fit the data from the training set. Once a model is trained, it is able to evaluate the likelihood of each observation sequence to belong to its class using (2):

$$P(O \mid \lambda) = \sum_{all\,Q} b_{q_1}(o_1) \cdot a_{q_1 q_2} \cdot b_{q_2}(o_2) \cdot \ldots \cdot a_{q_{T-1} q_T} \cdot b(o_T), \quad (2)$$

where $\{q_1, q_2, ..., q_T\}$ represents any state sequence out of all possible Q, $q_i \in \{1, ..., N\}$, with N being the number of states and $\lambda = (A, B, \pi)$ the trained model. In practice though, (2) is not used due to computational costs. Instead, other more efficient ways of computing $P(O \mid \lambda)$, such as the *forward* or *backward* procedures, are to be preferred. There is also a third option, namely the *Viterbi* algorithm, which does not compute exactly the desired likelihood, but the probability of the most likely state sequence. This algorithm is of high interest for our scenario as it also provides the path itself along with the associated probability for each observation symbol and this is efficiently used to discriminate models in a post-processing stage, consistently improving overall accuracy.

III. EXPERIMENTAL RESULTS

A. Experimental setup

The training dataset is identical to that used in [24]. The same 450 images representing 9 gestures recorded in clean strictly supervised conditions are used to train 9 Markovian models with discrete probability densities for the states. The set describes gestures of a single person and includes small variations in scale, translation and 0x, 0y and 0z alternative rotations for each gesture.

Additionally we tuned the models on a separate validation set. All images here (around 7300) belong to the same person and present similar clean conditions as in training.

Our test set is recorded in more challenging setup, including illumination changes, uncontrolled background and a wider range of rotation angles. It sums over 8500 images recorded by 6 persons in front of a Kinect camera. The subjects were permitted to freely express the gestures within a range of 1–1.5 meters away from the sensor. Some relevant examples are presented in Fig. 3.

At testing stage, each trained model was used to generate a score proportional to the probability that a given sample belongs to that particular class. Even though the score corresponds to the most probable state path (obtained by the *Viterbi* algorithm), the model hierarchy does not change, which allows us to directly compare scores and assign the winning class to the highest one.

Each test sample was assigned one of the 9 gesture classes and for each class a mean precision value was computed, by summing all correct classifications and dividing the result by the number of samples.

B. Tunning the parameters

The resulted scores were subject to further processing, by using the state paths that come along as a byproduct with the *Viterbi* procedure. By fixing empirical margins with respect to the mean state distribution obtained in the training process, we were able to model state consistency (i.e. how many discrete time measures pass through a particular state) and reduce the scores accordingly. Therefore, given that all the states for each model were consistent at training, the winning score would at least have passed the consistency check. Additional constraints (e.g. imposing similar lengths for states that belong to symmetrical models, like in the case of the second gesture) boost the accuracies even further, ending up with a mean recognition rate of 93.38% on the test set (95.52% on validation set), which is more than satisfactory, given that the models were trained and tuned on clean data. This result enforces the generalizing abilities of statistical models that manage to keep high accuracies when moved from validation set to test data.

C. Comparison to state-of-the-art

We compared our results with two recent approaches [6] [12] that share similar or close enough scenarios. The first one uses a structural description of each gesture based on higher level

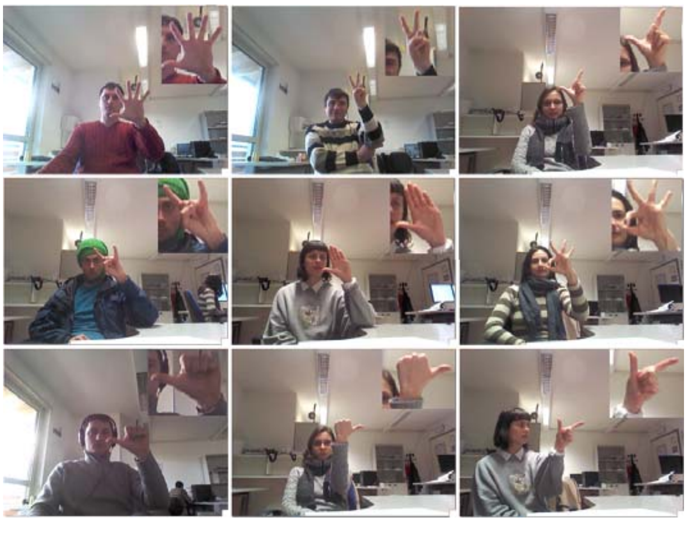

Figure 3. Examples from the test set showing each of the 9 static gestures recorded. Classes are numbered from top to bottom and from left to right

features like finger tips, segments and their position within the hand posture. Classification is performed using decision trees. The second approach extracts the angle count, skin color angle and non-skin color angle in combination with Hu invariant moments features and uses *k*-nearest neighbor algorithm (*k*-NN) as a classifier. In order to prove that HMMs are suitable for this task, we compare our results against state-of-the-art SVM classifier.

Fig. 4 shows the results obtained by all considered methods on the test set. Each bar displays the precision value for each gesture class, as described in section *A* of this paragraph. Our proposed approach outperforms the other three on average, managing to obtain high accuracies on all classes, except the last one, where, due to noisy samples as well as high rotation angles, the precision did not pass 80%.

The method from [6] relies heavily on detecting finger tips, which in many samples from our dataset are not entirely well defined. The second approach is also having difficulties in distinguishing gesture classes, mostly because the Hu features are invariant to rotation changes and of all 9 gestures, the 9th and the 8th are obtained by rotating the 3rd and the 7th respectively. On average the system from [6] recognized 72.3% of all test samples, whereas the one from [12] obtained a mean recognition rate of 69.22%. By combining our features with SVM and Histogram Intersection kernel, we obtain an accuracy of 88.31%, which is still more that 5% less that our mean result.

IV. Conclusions

In this paper we present an efficient system that addresses Static Hand Gesture Recognition using discrete HMMs and angle features extracted from gestures' silhouettes. The system is robust against changes in scale, translation and small rotations and also against person specifics. Additional robustness is brought by using a Kinect sensor, which allows separating the close range hand object from virtually any type of background.

Experimental results confirm the discriminative power of the chosen features as well as the flexibility and generalizing ability of the statistical models.

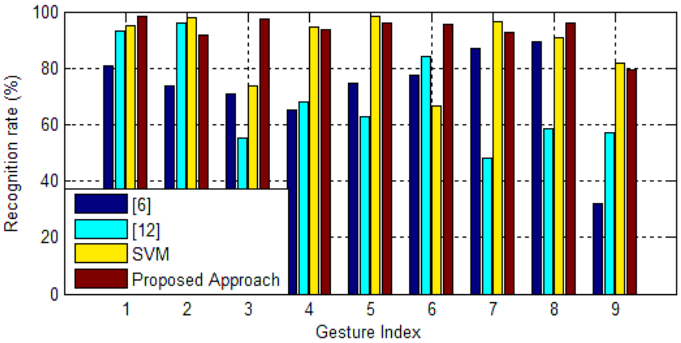

Figure 4. Recognition rates obtained by different approaches from literature in comparison with our method for each gesture

References

[1] http://www.microsoft.com/en-us/kinectforwindows/

[2] http://www.asus.com/Multimedia/Xtion_PRO_LIVE/

[3] A. Just, O. Bernier, S. Marcel, "HMM and IOHMM for the recognition of mono- and bi-manual 3d hand gestures", Proc. of the British Machine Vision Conference, pp. 28.1-28.10, 2004.

[4] R.Y. Wang, J. Popovic, "Real-time Hand-Tracking with a Color Glove", ACM Trans. On Graphics, vol. 28(3), pp.63.1-63.8, 2009.

[5] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, J. Yang, "A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors", Trans. on Systems, Man and Cybernetics, vol.41(6), pp.1064-1076, 2011.

[6] S. Oprisescu, C. Rasche, S. Bochao, "Automatic static hand gesture recognition using ToF cameras", in Proc. of the 20th European Signal Processing Conference (EUSIPCO), pp.2748-2751, 2012.

[7] C. Rasche, "An Approach to the Parameterization of Structure for Fast Categorization", International Journal of Computer Vision, vol. 87(3), pp.337-356, 2010.

[8] A. Erol, G. Bebis, M. Nicolescu, R.D. Boyle, X. Twombly, "Vision based hand pose estimation: A review", in Computer Vision and Image Understanding, vol. 108, pp.52-73, 2007.

[9] V. Atitsos, S. Sclaroff, "Estimating 3D hand pose from a cluttered image", Conf. on Computer Vision and Pattern Recognition, pp.432-442, 2003.

[10] X. Teng, B. Wu, W. Yu, C. Liu, "A hand gesture recognition system based on local linear embedding", Visual Language and Computing, vol. 16(5), pp.442-454, 2005.

[11] S. S. Ge, Y. Yang, T. H. Lee, "Hand gesture recognition and tracking based on distributed locally linear embedding", Image and Vision Computing, vol. 26(12), pp.1607-1620, 2008.

[12] L. Yun, Z. Lifeng, Z. Shujun, "A Hand Gesture Recognition Method Based on Multi-Feature Fusion and Template Matching", in Procedia Engineering, Volume 29, Pages 1678-1684, 2012.

[13] A.F. Bobick, J.W. Davis, "The Recognition of Human Movement Using Temporal Templates", Trans. Pattern Analysis and Machine Inteligence, vol. 23(3), pp. 257-267, 2001.

[14] M. Elmezain, A. Al-Hamadi, B. Michaelis, "Hand Gesture Spotting Based on 3D Dynamic Features Using Hidden Markov Models", Signal Processing, Image and Pattern Recognition, vol. 61, pp.9-16, 2009.

[15] T. Schlömer, B. Poppinga, N. Henze, S. Boll, "Gesture recognition with a Wii controller", Proc. of Intl. Conf. on Tangible and Embedded Interaction, pp.11-14, 2008.

[16] R. Yang, S. Sarkar, B. Loeding, "Enhanced Level Building Algorithm for the Movement Epenthesis Problem in Sign Language Recognition", Conf. on Computer Vision and Pattern Recognition, pp.1-8, 2007.

[17] R. Yang, S. Sarkar, "Detecting Coarticulation in Sign Language using Conditional Random Fields", Intl. Conf. on Pattern Recognition, pp.108-112, 2006.

[18] R. Yang, S. Sarkar, B. Loeding, "Handling Movement Epenthesis and Hand Segmentation Ambiguities in Continuous Sign Language Recognition Using Nested Dynamic Programming", Trans. on Pattern Analysis and Machine Intelligence, vol.32(3), pp.462-477, 2010.

[19] H. Yang, S. Sclaroff, S. Lee, "Sign Language Spotting with a Threshold Model Based on Conditional Random Fields", Trans. on Pattern Analysis and Machine Intelligence, vol.31(7), pp.1264-1277, 2009.

[20] A. Thangali, S. Sclaroff, "An alignment based similarity measure for hand detection in cluttered sign language video", Computer Vision and Pattern Recognition Workshops, pp.89-96, 2009.

[21] S. Mitra, T. Acharya, "Gesture Recognition: A Survey", Trans. on Systems, Man, and Cybernetics, vol.37(3), pp.311-324, 2007.

[22] M. Van den Bergh, D. Carton, R. de Nijs, N. Mitsou, C. Landsiedel, K. Kuehnlenz, D. Wollherr, L. Van Gool, M. Buss, "Real-time 3D hand gesture interaction with a robot for understanding directions from humans", RO-MAN, pp.357-362, 2011.

[23] Z. Ren, J. Meng, J. Yuan, Z. Zhang, "Robust hand gesture recognition with Kinect sensor", Proc. of ACM Multimedia, pp. 759-760. 2011.

[24] R.L. Vieriu, B. Goraş, L. Goraş, "On HMM static hand gesture recognition," Intl. Symp. on Signals, Circuits and Systems, pp.221-224, 2011.