World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:7, No:11, 2013

Vision Based Hand Gesture Recognition Using Generative and Discriminative Stochastic Models

Mahmoud Elmezain, Samar El-shinawy

Open Science Index, Computer and Information Engineering Vol:7, No:11, 2013 publications.waset.org/17283/pdf

Abstract—Many approaches to pattern recognition are founded on probability theory, and can be broadly characterized as either generative or discriminative according to whether or not the distribution of the image features. Generative and discriminative models have very different characteristics, as well as complementary strengths and weaknesses. In this paper, we study these models to recognize the patterns of alphabet characters (A-Z) and numbers (0-9). To handle isolated pattern, generative model as Hidden Markov Model (HMM) and discriminative models like Conditional Random Field (CRF), Hidden Conditional Random Field (HCRF) and Latent-Dynamic Conditional Random Field (LDCRF) with different number of window size are applied on extracted pattern features. The gesture recognition rate is improved initially as the window size increase, but degrades as window size increase further. Experimental results show that the LDCRF is the best in terms of results than CRF, HCRF and HMM at window size equal 4. Additionally, our results show that; an overall recognition rates are 91.52%, 95.28%, 96.94% and 98.05% for CRF, HCRF, HMM and LDCRF respectively.

Keywords—Statistical Pattern Recognition, Generative Model, Discriminative Model, Human Computer Interaction.

I. Introduction

THe hand gesture recognition is an active area of research in the vision community, mainly for the purpose of sign language recognition. Sign language recognition is an application area for Human Computer Interaction (HCI) to communicate with computers. The goal of pattern interpretation is to push the advanced human-computer communication to bring the performance of HCI close to human-human interaction. In the last decade, several methods of potential applications [1], [2], [3], [4], [5], [6] in the advanced hand gesture interfaces have been suggested but these differ from one another in their models. Some of these models are Neural Network [7], Hidden Markov Model (HMM) [1], [8], [9], Dynamic Time Warping (DTW) [10] and Conditional Random Field [3], [4], [5]. Elmezain *et al.* [1] proposed a real-time system to recognize American Sign Language (ASL) and numbers (0-9) using HMM. The experiment was performed with isolated hand motion trajectory that was employed for HMM for recognition and achieved 94.72% accuracy. Yang *et al.* [2] introduced an ASL recognition system based on a time-delay neural network. This system used the motion information to extract hand position where the recognition rate was 96.2%. Yang *et al.* [3] introduced a method for designing threshold model in a CRF model, which performs an adaptive threshold to distinguish between signs and non-sign patterns. The experiments were performed with isolated and continuous data set according to extracted six features. Sminchisescu *et al.* [4] applied CRF model to recognize human motion activities and showed improvement over an HMM technique. The difference between HMM and CRF is that HMM is generative model that defines a joint probability distribution to solve a conditional problem thus focusing on modeling the observation to compute the conditional probability $p(y|x)$. Moreover, one HMM is constructed per label (i.e. pattern) where HMM assumes that all the observation are independent. CRF uses an undirected graphical model to overcome the weakness of Maximum Entropy Markov Model (MEMM) [5]. CRF uses a single model of the joint probability of the labels sequence given the observation sequence. Therefore, there is trade-off in the weights of occurrences number of a feature value for each state [3]. Hidden Conditional Random Field (HCRF) is the extension of CRF that include hidden variables [11], [12]. HCRF can automatically model the local interconnection between labels (i.e. states) with hidden variables, but it cannot model dynamics among states. On the other sides, Latent-Dynamic Conditional Random Field (LDCRF) can model the sub-structure of a state and learn dynamic among states [6]. The LDCRF model combines the strengths of CRF and HCRF. Furthermore, it can detect and recognize states from un-segment data (Fig.1). The main contribution of

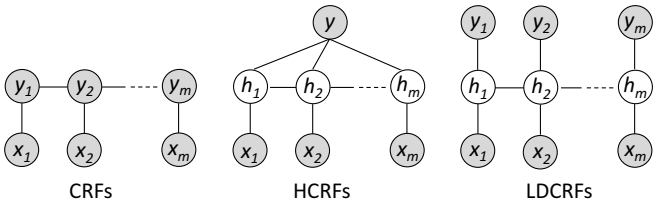


Fig. 1. Different type of discriminative models: CRFs, HCRFs and LDCRFs. In these models, x_j refers to the j^{th} corresponding observation value, h_j is a hidden states that assigned to x_j. y_j is the label of x_j where the gray circles represent the observed variables.

this paper is to investigate the generative and discriminative models for recognizing the patterns of alphabets characters (A-Z) and numbers (0-9). The experimental results discover that LDCRF is the best in terms of results than CRF, HCRF and HMM. Additionally, LDCRF can automatically recognize hand patterns (i.e. gestures) with 98.05%. The rest of this paper is organized as follow; Section II reviews the basic classification generative and discriminative techniques. The main difference between generative and discriminative models

Mahmoud Elmezain was the Depatrment of Mathematics, Tanta University, Egypt. He is now with Institute for Information Technology and Communications (IIKT), Otto-von-Guericke-University Magdeburg, Gemany.
(e-mail: mahmoud.elmezain@ieee.org).
Samar El-shinawy is with Mathematics Department (Computer Science Division), Faculty of Science, Tanta University, Egypt.
(e-mail: samar_ahmedd99@yahoo.com).

Open Science Index, Computer and Information Engineering Vol:7, No:11, 2013 publications.waset.org/17283/pdf

is described in Section III. Experimental results are described in Section IV. Section V concludes this paper.

II. CLASSIFICATION

In computer vision, a good choice for classification approaches helps the success of any system and makes it suitable for real-world applications. Classification of symbols in pattern recognition assigns them to respective classes. An application of hand gesture-based interaction with alphabet characters and numbers is implemented to demonstrate the coactions of suggested components and the effectiveness of gesture (i.e., hand motion) recognition. The isolated hand gesture is handled according to two different classification techniques: generative model such as HMM and discriminative model like CRF, to decide which one is the best in terms of performance. The following two subsections discuss how HMM and CRF are employed for the classification of alphabets character and numbers.

A. Generative Model: HMM

The most widely used recognition algorithm for pattern recognition is HMM [13], [14]. HMM is mathematical model of the stochastic process which generates a sequence of observations according to the previously stored information. In the Markov chain, every state of the model can only observe a single symbol. However, all states in Hidden Markov Model topology can observe one symbol out of a distinct gesture. The probability of observing a symbol for each state is stored in the observation probability distribution matrix. Furthermore, HMM states are called hidden for the following reasons. Firstly, the decision of observing a symbol represents the second process. Secondly, the emitter of an HMM only emits the observed symbol. Finally, the emitting states are unknown since the current states are based on the previous states. HMM have many advantages that are rich mathematical framework, powerful learning and decoding methods, good sequences handling capabilities, and flexible topology for the statistical phonology and the syntax. The disadvantages lie in the poor discrimination between the models and in unrealistic assumptions that must be make to construct the HMM theory, namely the independence of the successive feature frames (i.e. input vectors) and the first order Markov process [15].

1) Elements of HMM: A Hidden Markov Model can be symbolized with $\lambda = (A, B, \pi)$ and is characterized by the following elements [1], [13];

- The set of states $S = \{s_1, s_2, ..., s_N\}$. N represents the number of states in the model.
- An initial probability distribution for each state π such that;

$$\pi_i = P(s_i), \;\; 1 \leq j \leq N \tag{1}$$

- An *N*-by-*N* transition matrix $A = \{a_{ij}\}$, which is given by;

$$a_{ij} = P(s_j|s_i), \;\; 1 \leq i, j \leq N \tag{2}$$

where a_{ij} is the probability of the transition from state s_i at time t to s_j at time $t+1$. The sum of the entries in each row of matrix *A* must be 1 because it is the sum of the probabilities of making a transition from a given state to each other states.

$$\sum_j a_{ij} = 1 \tag{3}$$

- The set of possible emission (an observation) $O = \{o_1, o_2, ..., o_T\}$ in which *T* is the length of gesture path.
- The set of discrete symbols $V = \{v_1, v_2, ..., v_M\}$, where M represents the number of distinct observation symbols per state (i.e. the size of a codeword).
- An *N*-by-*M* observation matrix $B = \{b_j(m)\}$, where

$$b_j(m) = P(v_m|s_j), \;\; 1 \leq j \leq N, \;\; 1 \leq m \leq M \tag{4}$$

$$\sum_m b_j(m) = 1 \tag{5}$$

where $b_j(m)$ gives the probability of emitting symbol v_m at state s_j. The sum of the entries in each row of matrix *B* must be 1 for the same pervious reason.

In short, a complete specification of the HMMs contains two model parameters (N and M). Additionally, it also includes the observation symbols and the three probabilistic parameters A, B and π. Thus, a compact notation of HMM is as follows;

$$\lambda = P(\pi, A, B) \tag{6}$$

Here, λ refers to the parameters set of the model.

2) HMMs Basic Problems: Mathematically, three factors control the use of HMMs. These factors lie in their topologies, the selected features to be emitted and their observation probabilities. The feature selections are based on the observation task. There are three main problems for HMMs; and their solutions helps to employ transitions and observation probabilities in a good way for real-world applications. The problems are:

- **Evaluation problem**: Given the observation sequence O and the model parameter λ, how to compute the probability of observed sequence given the model parameter (i.e. $P(O|\lambda)$)?
- **Decoding problem**: Given the observation sequence O and the model parameter λ, how to determine the best path through λ that generates $O = \{o_1, o_2, ..., o_T\}$ with maximum likelihood (i.e. best explains the observations)?
- **Estimation problem**: Given the observation sequence O, how to adjust or re-estimate the model $\lambda = P(\pi, A, B)$ to generate $O = \{o_1, o_2, ..., o_T\}$ with maximum likelihood?

3) Model Size: Before the HMMs training starts, the size of HMMs must be decided. How many states do we need?

The number of states must be estimated by considering the complexity of the various patterns that HMMs will be used to distinguish. In other words, the number of segmented parts in the graphical pattern is taken into consideration when we represented it. When the number of training data samples is insufficient, the use of excessive state numbers causes the

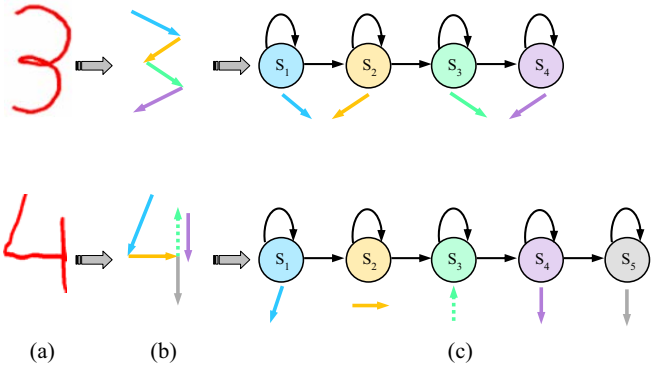


Fig. 2. Straight-line segment for HMMs topologies (a) Gesture number from hand motion trajectory (b) Line segment of gesture number (c) LRB model with line segmented codewords.

over-fitting problem[1]. In addition, the discrimination power of the HMMs is decreased when insufficient number of states is used because more than one segmented part of graphical pattern is modeled on one state. The number of states in our gesture recognition system is determined by mapping each straight-line segment into a single HMM state (Fig. 2). To represent various graphical patterns, we must look at the possible patterns and estimate how many distinguishable segments are contained in a pattern. It may be a good idea to use different numbers of states in the different HMMs, which used to represent separate classes of patterns. For example, to represent a graphical pattern 'L', only two states are needed, whereas six states are required for a graphical pattern 'E', and four states for graphical pattern '3'.

4) Initializing a Left-Right Banded Model: Before starting the iterative Baum-Welch algorithm, the initial values of all parameters in the HMMs must be assigned. There is only one general requirement; the initial model must indicate, somehow, what we want to represent different model states. However, this requirement has different consequences, depending on the type of HMMs. In practice, the LRB model is considered because each state in Ergodic topology has many transitions than LR and LRB topologies, so, the structure data can be easily lost. On the other hand, LRB topology has no backward transition so, the state index either increases or remains the same as time increases. In addition, LRB topology is more restricted than LR topology and simple for training data, which can match the data to the model [1].

An intuitively observation is that, a good initialization for HMMs parameters (A, B, π) achieves better results. Matrix A is the first parameter, where it is determined using Eq. 7.

$$A = \begin{pmatrix} a_{11} & 1-a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & 1-a_{22} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{pmatrix} \tag{7}$$

The diagonal elements a_{ii} of the transition matrix can be chosen to indicate approximately the average state durations d such that;

$$a_{ii} = 1 - \frac{1}{d} \tag{8}$$

and

$$d = \frac{T}{N} \tag{9}$$

where T is the length of gesture path and N represents the number of states.

This is sufficient for an automatic training procedure in which state 1 is intended to represent the first part of the training data, state 2 the next part, etc. Therefore, all output probability distributions for different states can be initialized with the same parameters for all states. Consequently, the first step in Baum-welch iteration uses the training data to calculate more correct output probability parameters for each state. Since HMMs states are discrete, all elements of matrix B are initialized with the same value for all different states (Eq. 11). Matrix B is an N-by-M observed symbols where b_{im} gives the probability of emitting symbol v_m in state i (Eq. 4).

$$b_{im} = \frac{1}{M} \tag{10}$$

where i, m run over the number of states and the number of discrete symbols, respectively.

$$B = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1M} \\ b_{21} & b_{22} & \cdots & b_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{N2} & \cdots & b_{NM} \end{pmatrix} = \begin{pmatrix} \frac{1}{M} & \frac{1}{M} & \cdots & \frac{1}{M} \\ \frac{1}{M} & \frac{1}{M} & \cdots & \frac{1}{M} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{M} & \frac{1}{M} & \cdots & \frac{1}{M} \end{pmatrix} \tag{11}$$

For each new time sample, the state can jump back by itself, or only to the nearest following state. Therefore, the initial probability vector π should be initialized as;

$$\pi = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}^T \tag{12}$$

It is to ensure that it begins from the first state.

5) Termination of HMMs Training: The Baum-Welch training algorithm is very efficient. Often a good model is reached already after 5-10 iterations. The trained model must be flexible enough to correctly represent a new test sequence that never occurred during training. The training step is repeated until the change of transition and emission matrix converges. The convergence is satisfied if the change is less than 0.001 (i.e. tolerance ϵ = 0.001) as described in Eq. 13, or reaches to the maximum number of iterations (i.e. 500).

$$\sum_{i=1}^{N}\sum_{j=1}^{N} |\hat{a}_{ij} - a_{ij}| + \sum_{j=1}^{N}\sum_{m=1}^{M} |\hat{b}_{jm} - b_{jm}| < \epsilon \tag{13}$$

[1]Over-fitting occurs when HMMs describe random error instead of the underlying relationship. Potential over-fitting problem does not only depend on the number of parameters and data, but also on the compatibility of model structure with the amount of model error and data shape. To avoid the problem of over-fitting, additional techniques (e.g. regularization, early stopping, cross-validation and etc.) are used when further training is not resulting in better generalization.

Open Science Index, Computer and Information Engineering Vol:7, No:11, 2013 publications.waset.org/17283/pdf

Open Science Index, Computer and Information Engineering Vol:7, No:11, 2013 publications.waset.org/17283/pdf

The main motivation behind using tolerance is to control the number steps required by the Baum-Welch algorithm in order to successfully execute its purpose. This algorithm is terminated if all of the following three quantities are less than the tolerance value. First, log-likelihood for a given observation sequence O is generated using the current estimated values of transition matrix A and observation matrix B. Second, change in the normalization of the transition matrix A. At the end, change in the normalization of the observation matrix B. Note that, increasing tolerance reduces the number of steps to execute the Baum-Welch algorithm before it was terminated. In fact, the maximum number of iterations controls the maximum number of steps to execute the algorithm. If the Baum-Welch algorithm executes 500 iterations before reaches to the specified tolerance value, the termination is occurred with a warning. When this occurs, the value of maximum number of iterations should be increased so that the algorithm reaches to the desired tolerance before termination.

It is usually very difficult to provide sufficient amounts of training data. Therefore, some observation may never occur in the limited set of training data, although we may know that they might have occurred with some small probability. If a discrete HMM is trained on a such data, the Baum-Welch will assign *zero* observation probability to some elements of the observation probability matrix. In such case, a very small non-zero value may be assigned and re-normalization of the row matrix is required. A similar problem can occur with the transition probability matrix. For a left-right banded HMM we have intentionally defined many elements of the transition probability matrix exactly *zero* values. These elements still have zero values after the Baum-Welch training, and should remain zero. Furthermore, the adjustment of HMMs parameters is important after performing the training operation.

B. Conditional Random Fields

Conditional Random Field (CRF) is undirected graphical models that were developed for labeling sequential data. CRF is different than HMM in their conditional nature and the dependencies assumptions in their computations to ensure tractable inference. In addition, CRF overcomes the weakness of directed graphical models, which suffer from the bias problem as in Maximum Entropy Markov models (MEMM) [16], [17]. Furthermore, CRF combines the strength of MEMM and HMM on a number of real-world sequence labeling tasks. In our work, each label (state) corresponds to a gesture (e.g. alphabets from A to Z or numbers from 0 to 9). In addition, there is a trade-off for each label according to the weights of each feature function because CRF uses a single exponential distribution to model all reference labels of given observation [3]. The CRF is satisfied by defining the normalized each product of potential function [19]. In the case of chain-structured CRF as depicted in Fig.1, each potential function operates on pairs of adjacent label variables y_i and y_{i+1}.

The probability of label sequence y for a given observation sequence x is calculated by;

$$p(y|x,\theta) = \frac{1}{Z(x,\theta)} \cdot \exp\Big(\sum_{i=1}^{n} F_\theta(y_{i-1}, y_i, x, i)\Big) \qquad (14)$$

where $Z(x,\theta)$ is the normalized factor given by;

$$Z(x,\theta) = \sum_{y} \exp\Big(\sum_{i=1}^{n} F_\theta(y_{i-1}, y_i, x, i)\Big) \qquad (15)$$

where parameter $\theta = (\lambda_1, \lambda_2, ..., \lambda_{N_f}; \mu_1, \mu_2, ..., \mu_{N_g})$, N_f represents the number of transition feature function, N_g refers to the number of state feature function and n is the length of observation sequence x. F_θ is defined as follows;

$$F_\theta(y_{i-1}, y_i, x, i) = \sum_{f} \lambda_f t_f(y_{i-1}, y_i, x, i) + \sum_{g} \mu_g s_g(y_i, x, i) \qquad (16)$$

where $t_f(y_{i-1}, y_i, x, i) \simeq t_f(y_{i-1}, y_i, x)$ is a transition feature function of the entire observation sequence and labels at positions i and $i-1$ in the label sequence. $s_g(y_i, x, i) \simeq s_g(y_i, x)$ refers to a state feature function of the label at position i and the observation sequence. λ_f and μ_g represent the weights of the transition and state feature functions respectively, which can be estimated from training data.

From Eq. 14 and Eq. 16, the joint probability of a label sequence y given an observation sequence x can be written as follows;

$$p(y|x,\theta) = \frac{1}{Z(x,\theta)} \cdot \exp\Big(\sum_{i=1}^{n}\sum_{f} \lambda_f t_f(y_{i-1}, y_i, x, i) + \sum_{i=1}^{n}\sum_{g} \mu_g s_g(y_i, x, i)\Big) \qquad (17)$$

As CRF is similar to HMM in their characteristics, it is easy to build a CRF model by defining a single feature for each label-observation pair (y_b, x) and label-label pair (y_a, y_b) according to the training data set as follow;

$$t_{y_a,y_b}(y_u, y_v, x) = \begin{cases} 1 & \text{if } y_u = y_a \text{ and } y_v = y_b \\ 0 & \text{otherwise} \end{cases} \qquad (18)$$

$$s_{y_b,x}(y_v, x_v) = \begin{cases} 1 & \text{if } y_v = y_b \text{ and } x_v = x \\ 0 & \text{otherwise} \end{cases} \qquad (19)$$

Based on the foregoing mentioned, the parameters $\mu_{y_b,x}$ and λ_{y_a,y_b} which corresponds to $s_{y_b,x}(y_v, x_v)$ and $t_{y_a,y_b}(y_u, y_v, x)$ features respectively are equivalent to the logarithms of the HMM observation and transition probabilities.

1) Learning Parameter for CRF: The maximum likelihood parameter estimation problem for CRFs which defines the probability distribution (Eq. 17) is the task of estimating the parameters $\theta = (\lambda_1, \lambda_2, ..., \lambda_{N_f}; \mu_1, \mu_2, ..., \mu_{N_g})$ from training data set $D = \{(x^{(j)}, y^{(j)})\}_{j=1}^{T_d}$. Here, $x^{(j)}$ is an observation sequence of training data set, $y^{(j)}$ represents the corresponding label sequence and T_d refers to the number of training sequences. The learning parameters of CRF is based on the maximum entropy. According to the principle of maximum entropy, it is considered a good measure for the variational problems as a finite training data. In addition, it has the ability to justify the probability distribution from

Open Science Index, Computer and Information Engineering Vol:7, No:11, 2013 publications.waset.org/17283/pdf

incomplete information. The maximization of log-likelihood that learns the parameter θ is computed by[2];

$$L(\theta) = \sum_{j=1}^{T_d} \log\, p(y^{(j)}|x^{(j)}, \theta) = \sum_{j=1}^{T_d} \Big(\sum_{i=1}^{n} F_\theta(y_{i-1}^{(j)}, y_i^{(j)}, x^{(j)}, i) - \log Z(x^{(j)}, \theta) \Big) \tag{20}$$

Up to now, there is no closed solution to Eq. 20. Instead, iterative techniques have been used to determine the best solution [3], [5]. Likelihood maximization is performed using a gradient ascent method with Broyden-Fletcher-Goldfarb-Shanno (BFGS) optimization technique with 300 iterations to converge [19];

$$\frac{\partial L(\theta)}{\partial \theta} = \sum_{j=1}^{T_d} \Big(\sum_{i=1}^{n} \frac{\partial F_\theta(y_{i-1}^{(j)}, y_i^{(j)}, x^{(j)}, i)}{\partial \theta} - \sum_{x} p(y|x^{(j)}) \sum_{i=1}^{n} \frac{\partial F_\theta(y_{i-1}, y_i, x^{(j)}, i)}{\partial \theta} \Big) \tag{21}$$

2) Inference CRF: To compute the probability $p(y|x, \theta)$ of label sequence y for the given new observation sequence x, a set of matrices is computed [4], [5]. To simplify some expressions, special starting y_0 and stopping y_{n+1} states are added. These states are dummy (i.e. observe no symbol and are passed without time delay). Suppose that $p(y|x, \theta)$ is given by Eq. 16. For each position i in the observation sequence, $M_i(x)$ is $|\mathcal{Y} \times \mathcal{Y}|$ matrix, which defined as follows;

$$M_i(y', y|x) = \exp\big(F_\theta(y', y, x, i)\big) \tag{22}$$

where $\mathcal{Y} = y_1, y_2, ..., y_l$ represents a set of labels of the training data set. l refers to the number of the labels, and y', y are the labels of $\mathcal{Y}$ at time i. Using this notation, the conditional probability of a label sequence y given the observation sequence x can be written as the product of the appropriate elements of the $n + 1$ matrices for that pair of sequences (Eq. 23);

$$p(y|x, \theta) = \frac{1}{Z(x, \theta)} \cdot \prod_{i}^{n+1} M_i(y_{i-1}, y_i|x) \tag{23}$$

Similarly, the normalization factor $Z(x, \theta)$ for observation sequence x is given by the $(starting, stopping)$ entry of the product of all $M_i(x)$ matrices;

$$Z(x, \theta) = \Big(\prod_{i=1}^{n+1} M_i(x) \Big)_{starting, stopping} \tag{24}$$

3) CRF with Hidden Variables: Other approaches including the hidden variables offer several advantages over previous CRF model. Although the CRF model the transition among gestures and overcome the weakness of directed graphical models which suffer from bias problem, it does not have the ability to learn the internal sub-structure of gesture sequences. Hidden Conditional Random Fields (HCRF) is the extension of CRF, which incorporate hidden state variables to deal well with gesture sub-structure [11], [12]. The main advantage of HCRF is to automatically model the local interconnection between labels (i.e. states) with hidden variables. However, it cannot model the dynamics among the states (Fig.1).

Latent-Dynamic Conditional Random Fields (LDCRF) is considered as one of the approaches, which combine the advantages of CRF and HCRF by using both extrinsic dynamics and intrinsic sub-structure [6]. The strategy of LDCRF is based on two main points. Firstly, they learn extrinsic dynamics by modeling the class labels. Secondly, they learn the intrinsic sub-structure of gesture sequence using intermediate hidden states. Thus, LDCRF models has the ability to overcome the main weaknesses of HCRF model (Fig.1). LDCRF model can be used to recognize the un-segmented sequences because they contain a class label per observation. Furthermore, LDCRF model can efficiently infer the gesture sequences during training and testing processes. HCRF model has only one label associated to each sequence while CRF and LDCRF have one label associated to each time sample in the sequence.

4) Data Format of CRF: CRF and LDCRF models are applied to un-segmented sequences while HCRF should be applied to pre-segmented sequences (only one label per sequence). The data and the label files are encoded using Comma Separated Values (CSV) format according to HCRF library. The CRF formulation is implemented by extending the software of the library of Hidden-state Conditional Random Field [20]. This library implements three models: CRF, HCRF and LDCRF with C++ and Matlab languages.

Each file contains multiple matrices or vectors encoding the feature values (data files) or label values (label files). A data file contains multiple matrices, one for each sequence. For each matrix, the first line always contains two numbers: the number of rows and the number of columns. The number of rows for each matrix represents the number of features. All the matrices should have the same number of features. The number of columns for a specific matrix represents the number of time samples in the sequence. Since HCRF model has only one label associated to each sequence while CRF and LDCRF have one label associated to each time sample in the sequence, the HCRF library supports two file format for labels. For HCRF model, the label file contains one integer per line, representing the label for the specific sequence. For CRF and LDCRF models, the label file is encoded as a data file with matrix headers specifying the number of rows and columns but in this case the matrices always have one row. This row should have the same length as the corresponding sequence in the data file, with one label for each time sample.

III. Generative Versus Discriminative Models

The difference between HMM and CRF is that HMM is the generative models and define a joint probability distribution to solve a conditional problem, thus focusing on modeling the observation to compute the conditional probability. Moreover, one HMM is constructed per label (i.e. each alphabet character or number) where HMM assumes that all the observations

[2]More details about the derivation of Eq. 20 can be found in [19]

Open Science Index, Computer and Information Engineering Vol:7, No:11, 2013 publications.waset.org/17283/pdf

are independent. CRF is undirected graphical model and is developed for labeling sequential data. The key features of CRF than HMM are represented in their conditional nature and the dependencies assumptions of their computations to ensure tractable inference. In addition, CRF overcomes the weakness of directed graphical models which suffer from the bias problem as in MEMM [18]. Furthermore, CRF combines the strength of MEMM and HMM where they have all characteristics of the directed graphical models as in HMM. In addition, each label in CRF is employed as exponential model as in MEMM to conditional probabilities of the next label for a given current label. Additionally, CRF uses a single model for all alphabets and numbers.

IV. Experimental Results

In our experimental results, the segmentation of the hand with complex background takes place using 3D depth map and color information over $YCbCr$ color space, which is more robust to the illumination variation and partial occlusion. Gaussian Mixture Model (GMM) was considered where a large database of skin and non-skin pixel is used to train it. Moreover, morphological operations were used as a pre-processing, and Mean-shift algorithm in conjunction with Kalman filter [1] is to track the hand to generate the hand motion trajectory. Combined features of location, orientation and velocity for hand gestures are extracted and then, k-means clustering is employed for HMM, CRF, HCRF and LDCRF codeword. Our experiments are carried out on isolated gestures according to two different classification techniques: generative model and discriminative models. The following sections discuss the analysis of HMM and CRF results in details.

A. Data Set

The alphabets and numbers are classified using HMMs, CRFs, HCRFs and LDCRFs by the motion trajectory of single hand. A database is developed containing 2160 video samples for gesture symbols taken from three subjects on a set of 26 alphabets and 10 numbers. In other words, each isolated gesture is based on 60 video sequences where 42 video samples for training and 18 video samples for testing (In total, our database contains 1512 video samples for training and 648 video samples for testing). The sample test data is entirely different from the training data and is tested on *Intel(R) Core(TM)2 Duo CPU 2.2GHz PC with 4 GB of RAM*. The input images are captured by Bumblebee stereo camera system which has 6 mm focal length at 15FPS with 240×320 pixels image resolution, and Matlab implementation. Bumblebee camera is used for acquisition of 2D images along with depth map. Therefore the databases are captured in IESK lab[3], Otto-von-Guericke-University Magdeburg, Germany.

B. Experimental Discussion

To handle isolated gesture, CRF, HCRF and LDCRF with different number of window sizes (W) ranging from 0 to 7 are applied and tested to decide the best in term of recognition results. A window size of zero means that the feature vector at the current frame is only used to construct the input feature while the window size of three means that the input feature vector at each frame consists of seven feature vectors which contain the current frame, three preceding frames and three future frames. In our application, the size of window is based on the complexity of each gesture as described in previous section. So, multiple experiments have been conducted with a variety of window size to empirically conclude the optimal outcome of the recognition system. Fig. 3 shows the recognition rate of CRF, HCRF and LDCRF according to different window sizes for training and testing data. The recognition of hand gesture path using LDCRF is higher than CRF and HCRF. In addition, the yield of training data is higher than testing data in the proposed method. Furthermore, the gesture recognition rate is initially improved as the window size increases but degrades as the window size further increases.

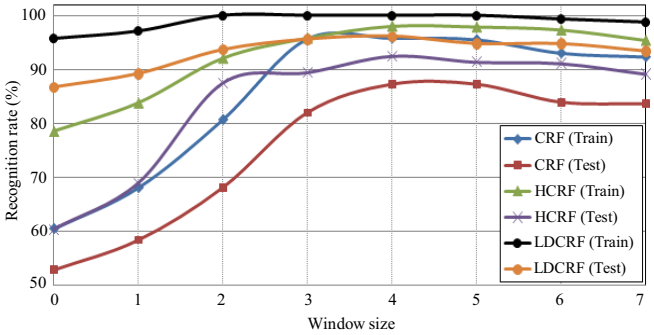


Fig. 3. Recognition accuracy with different window sizes (0-7) for CRFs, HCRFs and LDCRFs on training and testing data.

In HMM, we use a Left-Right Banded model based on Gaussian emission probabilities which have a full covariance matrix for each state. The HMM parameters (i.e. the emission probability and the state transition matrix) are learned from the same training data used by CRF. HMM is trained by Baum-welch algorithm while CRF is trained using Gradient ascent with the BFGS optimization technique. On a standard desktop PC, training process is more expensive for CRF, HCRF and LDCRF than HMM since the required time to model ranges from 20 Minutes to several hours and is based on observation window. On the contrary, the Inference (i.e. recognition) process is less costly and very fast for all models with Sequences of several frames (e.g. more than 80 frames in a sequence). The type of observed gesture is decided with HMM by Viterbi algorithm, frame by frame. As shown in Table I, the overall recognition rates (the average of the training and the testing of recognition result) of HCRF at window size equal to 0 is higher than CRF. Also, in that case, the overall recognition rate achieved by HMM is 96.91%. Furthermore, HMM is the best in terms of results than CRF, HCRF and LDCRF at $W = 0$.

Whereas at window size equal to 4, LDCRF recognition rate is higher than HMM according to the training and the testing data (Fig. 4). Our results show that the overall recognition rates

[3]http://www.iesk.ovgu.de/

TABLE I
RESULTS OF GESTURES RECOGNITION AT $W = 0$

Model type	Data set		Recognition result (%)		
	Training	Testing	Training	Testing	Overall
CRFs	1512	648	60.34	52.78	56.56
HCRFs	1512	648	78.55	60.34	69.45
LDCRFs	1512	648	95.68	86.73	91.21
HMMs	1512	648	99.07	94.75	96.91

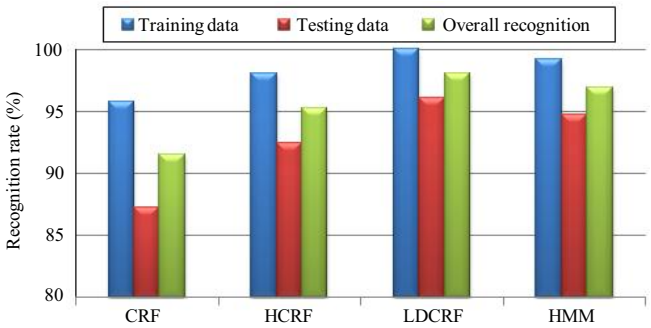


Fig. 4. Results of gestures recognition using CRFs, HCRFs, LDCRFs versus HMMs at window size = 4.

are 91.52%, 95.28%, 96.94% and 98.05% for CRF, HCRF, HMM and LDCRF, respectively. The recognition ratio is the number of correctly recognized gestures over the number of input gestures (Eq. 25).

$$Recognition\ ratio = \frac{\#\ recognized\ gestures}{\#\ test\ gestures} \times 100\% \tag{25}$$

The high recognition rate achieved by the proposed system is due to the following reasons; 1) As a benefit of depth information, a high segmentation accuracy of the hand is achieved. 2) A set of feature candidates that optimally discriminate among the input patterns is elected. 3) A carefully experimental based selection of initialization parameters for training process. 4) HMM, CRF, HCRF and LDCRF classification techniques have the ability to efficiently alleviate spatio-temporal variabilities.

V. CONCLUSION

Experiments were carried out on isolated gestures according to two different classification techniques: a generative models such as HMM and discriminative models like CRF, HCRF and LDCRF. For discriminative models, CRF, HCRF and LDCRF with different number of window sizes ranging from 0 to 7 were applied and tested to decide the best among them. In contrast to generative and discriminative models, HMM was the best in terms of results than CRF, HCRF and LDCRF at window size = 0. Whereas at window size equal to 4, LDCRF recognition results were higher than HMM according to the training and the testing data. Our results showed that, the overall recognition rates were 91.52%, 95.28%, 96.94% and 98.05% for CRF, HCRF, HMM and LDCRF, respectively. It is noted that the proposed system achieves high recognition rate due to a high segmentation accuracy of hand. In addition, a good election for the set of feature candidates that optimally discriminate among input patterns. Also, a careful experimental based selection is required for initialization parameters of training process. Above all, HMM, CRF, HCRF and LDCRF classification techniques have the ability to efficiently alleviate spatio-temporal variabilities.

REFERENCES

[1] M. Elmezain, A. Al-Hamadi and B. Michaelis, *Real-Time Capable System for Hand Gesture Recognition Using Hidden Markov Models in Stereo Color Image Sequences*, Journal of WSCG, Vol. 16, No. 1, pp. 65-72, 2008.
[2] M. Yang, N. Ahuja and M. Tabb, *Extraction of 2D Motion Trajectories and its Application to Hand Gesture Recognition*, EEE Trans. on PAMI, Vol. 24, No. 8, pp. 1061-1074, 2002.
[3] Yang, S. Sclaroff and S. Lee, *Sign Language Spotting with a Threshold Model Based on Conditional Random Fields*, IEEE Trans. on PAMI, Vol. 31(7), pp. 1264-1277, 2009.
[4] C. Sminchisescu, A. Kananujia and D. Metaxas, *Conditional Models for Contextual Human Motion Recognition*, Journal of CVIU, Vol.104, No. 2, pp. 210-220, 2006.
[5] J. Lafferty, A. McCallum and F. Pereira, *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling sequence Data*, Conf. on ICML, pp. 282-289, 2001.
[6] L. P. Morency, A. Quattoni and T. Darrell, *Latent-Dynamic Discriminative Models for Continuous Gesture Recognition*, IIEEE Conf. on CVPR, pp. 1-8, 2007.
[7] X. Deyou, *A Network Approach for Hand Gesture Recognition in Virtual Reality Driving Training System of SPG*, Conf. on ICPR, pp.519-522, 2006.
[8] M. Elmezain, A. Al-Hamadi, and B. Michaelis, *Spatio- Temporal Feature Extraction-Based Hand Gesture Recognition for Isolated American Sign Language and Arabic Numbers*, IEEE Symposium on ISPA, pp. 254-259, 2009.
[9] R. R. Lawrence, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceeding of the IEEE, Vol.77(2), pp. 257-286, 1989.
[10] K. Takahashi, S. Sexi and R. Oka, *Spotting Recognition of Human Gestures From Motion Images*, In Technical Report IE92-134, pp.9-16, 1992.
[11] A. Gunawardana, M. Mahajan, A. Acero and J. C .Platt, *Hidden Conditional Random Fields for Phone Classification*, Proceeding of European Conf. on Speech Communication and technology, pp. 1117-1120, 2005.
[12] A. Quattoni, S. Wang, L. P. Morency, M. Collins and T. Darrell, *Hidden Conditional Random Fields*,EEE Trans. on PAMI, Vol. 29, No.10, pp. 1848-1852, 2007.
[13] L. Rabiner, *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, Proceedings of the IEEE, Vol. 77, No. 2, pp. 257-286, 1989.
[14] X. D. Huang, Y. Ariki, and M. Jack, *Hidden Markov Models for Speech Recognition*, Edinburgh University Press, 1990.
[15] S. Goronzy, *HRobust Adaptation to Non-Native Accents in Automatic Speech Recognition*, Lecture Notes in Computer Sciences, Springer, ISBN-13: 978- 540003250, 2002.
[16] M. Elmezain, A. Al-Hamadi, and B. Michaelis, *Discriminative Models-Based Hand Gesture Recognition*, International Conference on Machine Vision, pp. 123-127, 2009.
[17] A. McCallum, D. Freitag, and F. Pereira, *Maximum Entropy Markov Models for Information Extraction and Segmentation*,International Conference on Machine Learning, pp. 591-598, 2000.
[18] J. Lafferty, A. McCallum, and F. Pereira, *Conditional Random Fields: Probabilistic Models for Segmenting and Labeling sequence Data*, International Conference on ICML, pp. 282-289, 2001.
[19] A. McCallum, *Efficiently Inducing Features of Conditional Random Fields*, Conf. on Uncertainty in AI, 2003.
[20] L. P. Morency, A. Quattoni, C. M. Christoudias, and S. Wang, *Hidden-state Conditional Random Field Library*, EVersion 1.3c, http://pt.sourceforge.jp/projects/sfnet hcrf/," 2008.

Open Science Index, Computer and Information Engineering Vol:7, No:11, 2013 publications.waset.org/17283/pdf