Proceedings of the IASTED International Conference
Robotics and Applications (RA 2010)
November 1 - 3, 2010 Cambridge, Massachusetts, USA

TRAJECTORY LEARNING BASED ON CONDITIONAL RANDOM FIELDS FOR ROBOT PROGRAMMING BY DEMONSTRATION

Aleksandar Vakanski [a,b], Farrokh Janabi-Sharifi [a], Iraj Mantegh [b], Andrew Irish [b]

[a] Department of Mechanical and Industrial Engineering, Ryerson University
350 Victoria Street, Toronto, Ont., M5B 2K3, Canada
aleksandar.vakanski@ryerson.ca, fsharifi@ryerson.ca

[b] National Research Council Canada (NRC) - Institute for Aerospace Research (IAR)
5145 Decelles Avenue, Montreal, Que., H3T 2B2, Canada
iraj.mantegh@cnrc-nrc.gc.ca, Andrew.Irish@cnrc-nrc.gc.ca

ABSTRACT

This work presents an approach for implementation of conditional random fields (CRF) in transferring motor skills to robots. As a discriminative probabilistic model, CRF models directly the conditional probability distribution over label sequences for given observation sequences. Hereby, CRF was employed for segmentation and labeling of a set of demonstrated trajectories observed by a tracking sensor. The key points obtained by CRF segmentation of the demonstrations were used for generating a generalized trajectory for the task reproduction. The approach was evaluated by simulations of two industrial manufacturing applications.

KEY WORDS

Robotics, Programming by Demonstration, Industrial Automation, Artificial Intelligence.

1. INTRODUCTION

One of the ultimate goals of the artificial intelligence field is creation of machines and robots with a high level of cognitive abilities. An important step toward this goal is inventing effective methods for machine learning and independent analytical reasoning. One such framework is robot programming by demonstration (PbD), where a robot acquires new skills from observation of the skills demonstrated by a human, or by another robot. Based on the observations, the learner robot creates an abstract representation of the task and generates a plan for reproduction of the demonstrated actions.

Probabilistic methods have been used in robot PbD for encoding the human motions, extracting the constraints of the demonstrated tasks, and for obtaining a generalized trajectory from a set of demonstrations. Hidden Markov Model (HMM) [1] has been employed for this purpose by a number of authors (cf. [2]–[5]). In a recent work, Irish et al. [5] applied HMM for both segmentation and temporal clustering of trajectory data observed from multiple demonstrations. Furthermore, they presented a new method based on Moving Window Principle Component Analysis (PCA) to extract feature data from the segmented trajectories that led into creation of the generalized robot trajectories based on the recognized features. Calinon *et al.* [6] used a Gaussian Mixture Model for spatio-temporal skill encoding, whereas generalization of the demonstrated trajectories was accomplished by using Gaussian Mixture Regression. Coates *et al.* [7] presented a method for PbD using Dynamic Bayesian Networks, based on learning the dynamic model of the demonstrations. Regression techniques were employed in [8], [9] for generalization of probabilistically modeled demonstrations.

In this work we investigate implementation of a new method for robot learning by using Conditional Random Field (CRF) [10] for (conditional) modeling of the demonstrated trajectories, and for extraction of the most relevant components for task reproduction.

CRF is a discriminative probabilistic approach for finding the conditional probability distribution of a sequence of hidden states (labels) $\mathbf{S}$ for a given sequence of observations $\mathbf{O}$, i.e., $p(\mathbf{S}|\mathbf{O})$. In essence, CRF is a discriminative model which is a probabilistic equivalent to HMM as a generative model (i.e., HMM defines the joint probability distribution of observed sequences and the hidden states sequences). Unlike HMMs which assume independence of the observations given the latent states, CRFs do not require the observations to be independent. This allows to involve complex features from the entire sequence of observations in calculating the probabilities of the hidden state variables.

One of application domains where CRFs have been most extensively used is the language processing. Examples include part-of-speech tagging [10], [11], shallow parsing [12], named-entity recognition [13], [14]. CRFs were reported to outperform HMMs for classifications tasks in these studies. Other areas of implementation include image segmentation [15], [16], gene prediction [17], activity recognition [18], [19], generation of objects' trajectories from video sequences [20], etc.

In the robot PbD environment, Kjellstrom *et al.* [21] presented an approach for grasp recognition from images using CRF. Martinez and Kragic [22] employed Support Vector Machines for activity recognition in robot PbD by

modeling the demonstrated tasks at a symbolic level, followed by using CRF for temporal classification of each sub-task into one of the several predefined action primitives classes. The comparison results of recognition rates between HMM and CRF indicated similar performance of both methods for short actions, and higher recognition rates by CRF for the case of prolonged continuous actions.

To the best of authors' knowledge, CRF has not been implemented before for acquisition of skills at the trajectory level in the area of robot PbD. The proposed method builds upon our previous work [23], which employed HMM for learning skills at a trajectory level in robot PbD. Both works are based on extracting the trajectories' key points, which refer to the relevant features for reproduction from the set of demonstrated trajectories [3], [24], [25]. This study employs CRF for identifying the sequence of hidden states for each demonstrated trajectory, based on the evidence from observed demonstrations. A cross-validation was employed, by training on the entire demonstrated set with one trajectory excluded, and performing inference on the remained trajectory. The trajectories key points were assigned at the transitions between the hidden states. Afterwards, the obtained key points were temporally shifted to a common time vector using the Dynamic Time Warping (DTW) [26]. A generalized trajectory was generated via cubic spline interpolation of the key points from all trajectories. The proposed approach was evaluated for simulated learning of industrial tasks of painting and peening.

The material of this paper is organized as follows. Section 2 provides preliminaries about the problem at hand and sets the terminology. Section 3 briefly introduces the theoretical background behind CRFs. Section 4 presents the proposed approach for implementing CRF for trajectories learning. The experiments and obtained results are given in Section 5. Section 6 discusses certain aspects of the presented work and section 7 summarizes the work.

2. PROBLEM DEFINITION

We consider a case where a skill is transferred to a robot based on multiple demonstrations of the same skill under similar conditions. A human demonstrator has the role of a teacher, by providing multiple examples of the skill. For the considered applications, the demonstrator uses a tool to accomplish the task goals, while the robot learner observes the demonstrations and records the tool's trajectories. Let denote the set of demonstrated trajectories with $\mathbf{O}^{(1)}, \mathbf{O}^{(2)}, ..., \mathbf{O}^{(N)}$, each of which represents a sequence of measurements data $\mathbf{O}^{(n)} = \left\{o_1^{(n)}, o_2^{(n)}, ..., o_{T_n}^{(n)}\right\}$. The sequence index is denoted by the superscripts $n = 1, 2, ..., N$, and each sequence can have a different length T_n. In a general case, each measurement consists of position and orientation of the tool $o_t^{(n)} = \left(x, y, z, \varphi, \vartheta, \psi\right)_t^{(n)}$ with respect to a predefined reference frame, where the subscript *t* denotes the time index of the measurement.

Based on a set of demonstrated trajectories, the goal is to generate a generalized trajectory to be executed by the robot for task reproduction. The generalization is based on extracting the relevant features for reproduction from the demonstrations. We relate these task features with the trajectories' key points, which represent the transitions between the different types of motions of the demonstrations. Therefore, it is first required to infer an unobserved (hidden) state $s_t^{(n)}$ for each observation feature $o_t^{(n)}$ from the trajectory $\mathbf{O}^{(n)}$ at time *t*. The hidden states are associated with certain types of performance (e.g., trajectory velocity or position), and the key points are assigned at the transitions between the hidden states. The efficiency of the trajectory segmentation with CRF can be assessed by the classification rate with regards to a set of initial labels for the trajectories. After the sequences of key points for the demonstrated trajectories are extracted, a generalized trajectory for task reproduction is obtained by applying a curve fitting technique.

3. CONDITIONAL RANDOM FIELD

CRF belongs to the family of undirected graphical models [27]. An undirected graph consists of a set of vertices (or nodes) connected by edges (or links) $G = (V, E)$, where the nodes represent random variables, and the (undirected) edges represent the conditional dependencies between the variables. A subset of nodes in the graph which contains a link between each pair of nodes is called clique. The cliques in the graph with the maximum number of conditionally dependent variables are the maximal cliques. The joint probability distribution of the graph variables is defined as a product of potential functions ψ_C,

$$p(\mathbf{x}) = \frac{1}{Z}\prod_C \psi_C\left(\mathbf{x}_C\right) \tag{1}$$

where each potential function corresponds to a maximal clique in the graph over a subset of variables $\mathbf{x}_C$. The quantity Z is called normalization factor (or partition function), and ensures that $p(\mathbf{x})$ in (1) represent a properly normalized distribution, i.e.,

$$Z = \sum_{\mathbf{x}}\prod_C \psi_C\left(\mathbf{x}_C\right). \tag{2}$$

In addition, in order to have proper values for the probabilities in (1), the potential functions $\psi_C\left(\mathbf{x}_C\right)$ must be real-valued and strictly positive. Consequently, the potential functions are usually expressed as exponentials of energy functions, i.e., $\psi_C\left(\mathbf{x}_C\right) = \exp\left\{E(\mathbf{x}_C)\right\}$. As a result, the joint distribution (1) is represented as a sum of the energies of maximal cliques of the graph

$$p(\mathbf{x}) = \frac{1}{Z}\exp\left(\sum_C E_C\left(\mathbf{x}_C\right)\right). \tag{3}$$

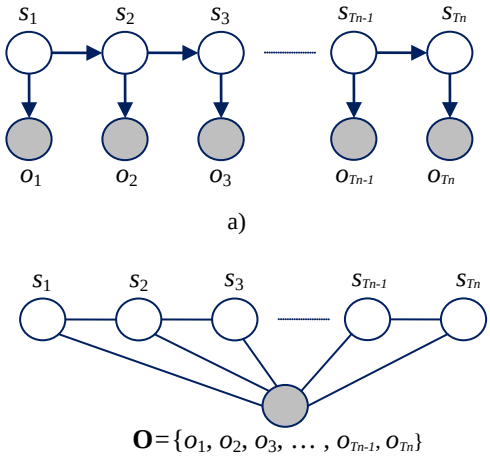

Fig. 1 Graphical models of: a) HMM, b) linear chain CRF. The shaded nodes depict the sequence of observed elements $\mathbf{O}=\{o_1, o_2, ..., o_{T_n}\}$, and the white nodes depict the hidden states sequence $\mathbf{S}=\{s_1, s_2, ..., s_{T_n}\}$.

For the applications with sequential data, the inputs represent ordered sequences (e.g., words in a text, labels of DNA sequence, trajectory measurements), and the corresponding structure of the graphical models is called linear chain (Fig. 1a shows an HMM graph as an example of a chain structure). A graphical model of a linear chain CRF is given in Fig.1b. In that case, the maximal cliques of the graph include the links between the pairs of adjacent state variables (s_t, s_{t-1}), and the links with the set of observations **O**. This is due to the assumption of first-order Markov property over state variables in the CRF graph structure, i.e., each state is conditionally independent of the other states given its neighbors states.

A *linear chain CRF* is defined as a conditional distribution $p(\mathbf{S}|\mathbf{O})$ of a sequence of unobserved states **S** given an observations sequence **O** [10], [11]:

$$p(\mathbf{S}|\mathbf{O}) = \frac{1}{Z}\exp\left\{\sum_l \mu_l\, f_l(s_t, s_{t-1}, \mathbf{O}, t) + \sum_m \eta_m\, g_m(s_t, \mathbf{O}, t)\right\}. \quad (4)$$

The set $\mathbf{\Lambda}=\{\mu_l, \eta_m\}$ denotes the parameters of the model. The functions $f_l(s_t, s_{t-1}, \mathbf{O}, t)$ correspond to transition feature functions of the states at times t and $t-1$ and the observation sequence **O**, whereas $g_m(s_t, \mathbf{O}, t)$ denote state feature functions of the observation sequence **O** and the state at time t. Often the notation is simplified by generalizing the notation of the above two feature functions into $\phi_k(s_t, s_{t-1}, \mathbf{O}, t)$. With this notation, ϕ_k can represents either a transition feature function or a state feature function, and the model (4) is written in the form

$$p(\mathbf{S}\,|\,\mathbf{O}) = \frac{1}{Z}\exp\left\{\sum_k \lambda_k\, \phi_k\,(s_t,\, s_{t-1}, \mathbf{O},\, t)\right\}. \quad (5)$$

As noted earlier, the normalization function Z is obtained by summing over all possible configurations on **S**, and enforces that the cumulative probabilities sum to one:

$$Z = \sum_{\mathbf{S}} \exp\left\{\sum_k \lambda_k\, \phi_k(s_t,\, s_{t-1}, \mathbf{O},\, t)\right\}. \quad (6)$$

The choice of feature functions will be discussed in Section 4. In the remaining of this section, we briefly address the problems of training and inference in CRFs.

For a given independent and identically distributed training set $\left\{\mathbf{O}^{(n)}, \mathbf{S}^{(n)}\right\}_{n=1}^{N}$ consisting of N observation sequences and the corresponding labelled sequences, the parameters of the model $\mathbf{\Lambda}$ are estimated by maximizing the conditional log-likelihood of the observations for the given labels

$$\ell(\mathbf{\Lambda}) = \sum_{n=1}^{N} \log p\left(\mathbf{S}^{(n)}\,\middle|\,\mathbf{O}^{(n)}\right) = \sum_{n=1}^{N}\left[\sum_{k=1}^{K} \lambda_k\, \phi_k\left(s_t^{(n)},\, s_{t-1}^{(n)},\, \mathbf{O}^{(n)},\, t\right) - \log Z\left(\mathbf{O}^{(n)}\right)\right]. \quad (7)$$

A noticeable property of the function $\ell(\mathbf{\Lambda})$ in (7) is that it represents a logarithm of sum of exponential components, and as a consequence it is concave. This property significantly facilitates the optimization problem and guarantees that any found local maximum is a global maximum.

Partial derivatives of the log-likelihood (7) with respect to parameters $\mathbf{\Lambda}$ are as follows

$$\frac{\partial \ell}{\partial \lambda_k} = \sum_{n=1}^{N} \phi_k\left(s_t^{(n)},\, s_{t-1}^{(n)},\, \mathbf{O}^{(n)},\, t\right) - \sum_{n=1}^{N}\sum_{\mathbf{S}} \phi_k\left(s_t^{(n)},\, s_{t-1}^{(n)},\, \mathbf{O}^{(n)},\, t\right) p\left(s_t^{(n)},\, s_{t-1}^{(n)}\,\middle|\,\mathbf{O}^{(n)},\, t\right). \quad (8)$$

Setting the partial derivatives in (8) to zero results in maximum entropy solution, i.e., the expectations of each feature function ϕ_k with respect to the empirical distribution of the training data and the expectations of the feature functions with respect to the model distribution are equalized.

Once the log-likelihood function (7) and its derivative (8) are obtained, estimation of the model parameters is solved by numerical optimization methods. Often quasy-Newton methods with approximation of the Hessian matrix are employed, such as limited-memory BFGS or conjugate gradient approach [11]. Overfitting of the model parameters can be avoided by regularization of the optimization function (7). In that case, the log-likelihood is penalized by adding an additional term proportional to the Euclidean (l_2) norm of the parameters vector $\sum \lambda_k^2$, or by adding an additional term proportional to the l_1 norm of the parameters $\sum |\lambda_k|$ [28].

Inference in linear chain CRFs is associated with solving two problems. The first one is finding the most likely sequence of states $\mathbf{S}'$ for a given sequence of observations $\mathbf{O}'$ and known model parameters. In addition, computation of the gradient (8) for parameters estimation during the training requires finding the pairwise marginal probabilities of the states

$p(s_t, s_{t-1} | \mathbf{O}, t)$. These two problems are efficiently solved with the Viterbi and forward-backward algorithm, respectively [11].

4. CRF FOR TRAJECTORY GENERALIZATION

Observed data in this work are the continuous trajectories captured during the demonstration phase. In general, observed features of CRFs can be either continuous or discrete in nature, whereas only discrete state variables are considered. First, we discuss formation of the feature functions, for both continuous and discrete observations.

Vail *et al.* [19] presented a CRF model for classification of activities based on observed continuous trajectories. The authors introduced the following continuous state feature functions

$$g_{i,u}(s_t, \mathbf{O}, t) = I(s_t = i)\, o_t(u), \quad (9)$$

for $i \in \{1, N_s\}$ (where N_s denotes the number of hidden states), and u denotes the dimensionality of the trajectories o_t. The notation $I(s_t = i)$ pertains to a binary indicator function, which equals to 1 when the state at time t is i, and 0 otherwise. To enhance the classification rate of the model, other continuous state feature functions were added to the model, such as velocities, squared positions, etc. The transition feature functions encode the transition scores from the state i at time $t-1$ to state j at time t, i.e.,

$$f_{i,j}(s_t, s_{t-1}, \mathbf{O}, t) = I(s_{t-1} = i)\, I(s_t = j), \quad (10)$$

for $i, j \in \{1, N_s\}$.

An important point reported in [19] was that the empirical sum of the features in (7) can be badly scaled when continuous features functions are used. This can cause slow convergence of the optimization algorithms, or in the worst case, the algorithm would not converge. As a remedy, the authors normalized the continuous observations to sequences with zero mean and variance one [19].

Most of CRF's applications in the literature deal with categorical input features (e.g., labeling words in a text), rather than continuous measurements. In that case, the observed features are mapped to a set of discrete symbols $o_t^{(n)} \in \{v_r\}_{r=1}^{R}$. The state feature functions are defined as binary indicators functions for each state-observation pairs

$$g_{i,r}(s_t, \mathbf{O}, t) = I(s_t = i)\, I(o_t = v_r), \quad (11)$$

for $i \in \{1, N_s\}$, and $r \in \{1, R\}$.

The transition feature functions are defined in an identical manner as in (10). The structure of CRFs allows additional observed features to be easily added to the model by generalizing the feature functions. For instance, the dependence of state transitions to the observed symbol can be modeled by adding additional feature functions $I(s_t = i)\, I(s_{t-1} = j) I(o_t = v_r)$.

Implementation of CRF in our work is based on discrete observation features, due to the scalability problems with the continuous features. For that purpose, the continuous trajectories were mapped to a discrete set of symbols by using the Linde-Buzo-Gray (LBG) algorithm [29]. The LBG quantization procedure is basically a variant of the *k*-means clustering technique [2], and it consists in grouping the input data into a pre-specified number of clusters. In our case, the preprocessing first involved normalization of each observation feature (e.g., *x*-position coordinates of the raw measurement data) to a sequence with zero mean and unity variance (similarly to the approach in [19]). Afterwards, the normalized data from the entire set of demonstrations (i.e., $\{x_t^{(n)}$ for $t \in (1, T_n)$, $n \in (1, N)\}$) were concatenated, and they were iteratively clustered with the LBG algorithm. After the clustering, each individual observation point was assigned to the closest centroid cluster. This procedure was repeated for all dimensions of the observed data, resulting in the set of discrete sequences $\mathbf{O}^{(1)}$, $\mathbf{O}^{(2)}$, ..., $\mathbf{O}^{(N)}$.

Training of the CRF model also requires providing the corresponding sequences of labels for each observation sequence. Hence, for initial labeling of the trajectories we employed again the technique of LBG clustering. This time for clustering we used a set of concatenated normalized positions and velocities vectors from all demonstrated trajectories $\{[x, y, z, \dot{x}, \dot{y}, \dot{z}]_t^{(n)}$ for $t \in (1, T_n), n \in (1, N)\}$. The clustering procedure associated each position-velocity vector with the closest centroid. The transitions between the clusters labels which occurred in the same temporal order in all demonstrations were adopted as key points. The detected initial key points yielded the labeled sequences $\mathbf{S}_{\text{init}}^{(1)}$, $\mathbf{S}_{\text{init}}^{(2)}$, ... , $\mathbf{S}_{\text{init}}^{(N)}$. This procedure assigns automatically the trajectories' key points, based on changes in position and velocity of the trajectories [23].

A common challenge associated with task modeling in robot PbD is the limited number of demonstrations available for estimation of the model parameters (since it may be frustrating for a demonstrator to perform many demonstrations, and in addition, the quality of the performance can decrease due to fatigue or other factors). Therefore, to extract the maximum information from limited training data, as well as to avoid testing on the training data, we adopted the leave-one-out cross-validation technique for training purposes. It consists of using a single observation sequence for validation, and using the remaining observation sequences for training. For instance, to label the observed sequence which corresponds to the third demonstration $\mathbf{O}^{(3)}$, a CRF model is trained on the set which consists of the observed sequences $\mathbf{O}^{(1)}$, $\mathbf{O}^{(2)}$, $\mathbf{O}^{(4)}$, ... , $\mathbf{O}^{(N)}$, and the corresponding state sequences $\mathbf{S}_{\text{init}}^{(1)}$, $\mathbf{S}_{\text{init}}^{(2)}$, $\mathbf{S}_{\text{init}}^{(4)}$, ... , $\mathbf{S}_{\text{init}}^{(N)}$ obtained from the initial selection of the candidate key points. The set of parameters $\mathbf{\Lambda}^{(3)}$ estimated during the training phase

using (7) and (8), are afterwards employed to infer the sequence of states $\mathbf{S}^{(3)}$ which maximizes the conditional probability $p(\mathbf{S}^{(3)} | \mathbf{O}^{(3)}, \Lambda^{(3)})$. This procedure was repeated for each observed sequence from the data set.

The inference problem of interest here consists of finding the most probable sequence of labels for an unlabeled observation sequence. The problem is typically solved by Viterbi decoding or maximal marginals [1]. Here we used the maximal marginals approach, by introducing additional state transitions constraints. Namely, the state sequences in the presented model are defined such that they start with the state 1 and are ordered consecutively in a left-right structure, meaning that state 1 can either continue with a self-transition or transition to state 2. Furthermore, some states might not appear in all sequences, e.g., a sequence can transition from state 5 directly to 7, without occurrence of state 6. By analogy with the right-left Bakis topology in HMMs, the contributions from the potential functions which correspond to transitions to past states (i.e., $\exp\left(\mu_{i,j}\, f_{i,j}(s_t, s_{t-1}, \mathbf{O}, t)\right)$ for $i < j,\ i, j \in \{1, N_s\}$) were minimized by setting low values for the parameters $\mu_{i,j}$, making these transitions less likely to occur. Additionally, another constraint minimized the possibility of transitioning to distant future states, by setting low values for the parameters $\mu_{i,j}$ for $j > i+2$. As a result, the potential functions for transitioning to more than 2 future states had approximately zero values. These constraints reflect the sequential ordering of the states in our model. Estimation of the label at time t was based on computing the maximal marginal probabilities of the graph edges, and were solved by the forward-backward algorithm [1]

$$p\left(s_t, s_{t-1} \middle| \mathbf{O}, t\right) \propto \alpha_{t-1}(s_{t-1})\, F_t\left(s_t, s_{t-1}, \mathbf{O}, t\right) \beta_t(s_t)\ , \quad (12)$$

where α and β denote the forward and backward variables which were calculated in the same recursive manner as with HMMs, and the functions F_t in (12) correspond to the transition feature functions in (4) [11]

$$F_t\left(s_t, s_{t-1}, \mathbf{O}, t\right) = \exp\left\{\sum_l \mu_l f_l\left(s_t, s_{t-1}, \mathbf{O}, t\right)\right\}, \quad (13)$$

for $l \in \{1, N_s^2\}$.

The classification rate comparing the initial sequence of states $\mathbf{S}_{\text{init}}^{(n)}$ and the sequence of states obtained by CRF labeling $\mathbf{S}^{(n)}$ was used as a measure of fit for each trajectory n,

$$CR^{(n)} = \frac{1}{T_n} \sum_{t=1}^{T_n} I\left(s_{t,\,\text{init}}^{(n)} = s_t^{(n)}\right). \quad (14)$$

For each sequence of hidden states, the key points were taken to be the transitions between the states. Since the recorded demonstrations differ in length, generating a generalized trajectory requires applying a technique to tackle the temporal variance. For that purpose, multidimensional DTW algorithm was employed [26]. The observed trajectory corresponding to the states sequence with the maximum classification rate was selected as a reference sequence $\mathbf{O}_{\text{ref}}$, against which all the other observation sequences were aligned. This process resulted in temporally warped sequences of key points, with the time stamps corresponding to the time vector of the reference sequence. Thus obtained key points from all trajectories were interpolated with a smoothing cubic spline, yielding a generalized trajectory. We introduced weighting coefficients for the importance of interpolation of the key points. The weighting coefficients were calculated based on the variance of each key point relative to the key points from the other trajectories with the same time stamps. Hence, the key points which pertained to highly variant parts of the demonstrations were subjected to greater smoothing.

5. EXPERIMENTS

Implementation of the presented concept in robot PbD environment was simulated for two types of manufacturing processes: painting and shot peening. The experimental setup involved capturing of tools' trajectories during the demonstrations by an optical marker-based tracker. The tracking system provided the tools' positions and orientations with respect to a global frame with a sampling frequency of 100 Hz. The setup shown in Fig. 2a was employed to simulate the trajectories of the tool for painting the target object.

For the first experiment, the painting task was demonstrated 14 times, and it consisted of first painting the contour of the object followed by painting the interior part of the panel. The demonstrations are shown in Figs. 2b and c. The trajectories length T_n ranged from 1393 to 2073 time frames. For this task, the orientation of the painting tool was approximately constant and normal to the panel, and therefore it was not considered as a relevant discriminative feature. Initially, the trajectories were automatically segmented with the LBG algorithm using 32 clusters of discrete vectors. A total of 33 transitions between the LBG's discrete symbols appeared in all trajectories, and these were adopted as the initial key points. We also added the first and last measurements of each trajectory as key points, resulting in 35 key points per trajectory. One sample trajectory with the identified key points is shown in Fig. 2b. We used the CrfChain toolbox [30] in MATLAB environment to implement our approach. The 3D positions and velocities were discretized into 32 symbols per sequence, making the total number of observation symbols R = 192. The percentage of the average classification rates of CRFs for different observation features are given in Table 1. The best classification rates were obtained when both velocities and positions were used as observation features. The generalized trajectory is shown with a red line in Fig. 2c. Note that even if some key points were wrongfully classified by the CRF, they would not have big impact on the generalization. This is due to the weighting scheme, which assigned low weights for fitting to the key points

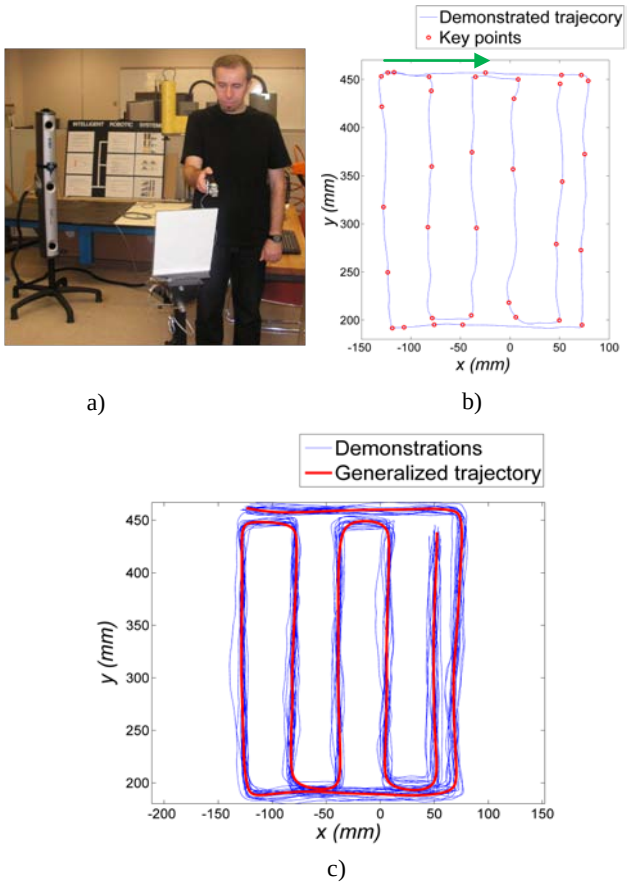

Fig. 2 a) Experimental setup for the painting experiment showing the optical tracker, the tool with attached markers and the object for painting. b) One of the demonstrated trajectories with the initially selected key points. The arrow indicates the direction of tool's motion. c) Demonstrated trajectories (blue lines) and the generalized trajectory (red line).

with high variability, relative to the key points from the other trajectories.

The proposed approach was compared to HMM, due to its wide usage for labeling and analysis of sequential data. To ensure that both methods were compared on an equal basis, a discrete multidimensional HMM was employed, with the same discrete observation sequences and label sequences which were used in the CRF approach. The multidimensional HMM reported in [31] was implemented. Differently from the one-dimensional HMM, the multidimensional HMM has one observation matrix for each dimension of the observed data. The model is built on an assumption that each dimension of the observed sequences is independent from the other dimensions. On the other hand, the state transition matrix has the same form as in the one-dimensional case. The training and inference problems are solved with small modifications of the one-dimensional HMM (see [31] for details). To maintain consistency with the adopted cross-validation testing for CRF, the HMM parameters were initialized for one of the labeled trajectories (e.g., $\mathbf{S}_{\text{init}}^{(3)}$, $\mathbf{O}^{(3)}$), followed by parameters estimation with the Baum-Welch algorithm for the rest of the trajectories (e.g., $\mathbf{O}^{(1)}$, $\mathbf{O}^{(2)}$, $\mathbf{O}^{(4)}$, ... , $\mathbf{O}^{(N)}$). The most probable sequence of hidden states for each observation trajectory and the given model were obtained by the Viterbi algorithm. This procedure was repeated for each of the observed trajectories. The mean classification rates from the multidimensional HMM are reported in Table 1. For the 3 cases of considered observation features, HMM had lower number of correct labels than CRF. The results are consistent with the findings in other studies in the literature [10], [11], [19], [22], which report of higher classification accuracy of CRF when compared to HMM.

Table 1. Means and standard deviations of the classification rates obtained by CRF and HMM for the painting task.

	CRF	HMM
1. Position	81.17 (± 9.2)	77.69 (± 1.96)
2. Velocity	76.86 (± 3.27)	74.75 (± 1.99)
3. Position, velocity	89.11 (± 3.13)	86.46 (± 1.20)

The second task for evaluation of the proposed approach pertained to material peening. Peening is a process of impacting material's surface with small spherical elements called shots. This process produces a compressive residual stress layer on material's surface, which increases the resistance to cracks caused by material fatigue or stress corrosion. Demonstration of the peening task was performed 7 times, with the number of measurements T_n varying between 2438 and 2655. Conversely to the painting experiment, where the demonstrations were simulated in the school lab, this task's demonstrations were captured for a real process of peening in an industrial environment. The trajectories represented waiving motions over a curved surface (Fig. 3). Tool's positions, velocities and orientations were examined as observation features. The number of clusters for initial key points selection with the LBG algorithm was set to 4. This choice was appropriate for the simple waving pattern of the task. In total 41 key points per trajectory were identified, making the number of hidden states N_s equal to 40. One sample trajectory with the initially selected key points is shown in Fig. 3a. Discretization of the trajectories was performed with the number of pre-specified clusters for LBG equal to 16, i.e., there were 16 discrete observation symbols for each dimension of the observed sequences. The classification rates of trajectories segmentation with CRF and discrete multidimensional HMM are given in Table 2. From the provided results it can be concluded that the velocity features are the most suitable for CRF segmentation of this example. The reason is that the end points in the trajectories (see Fig. 3a) correspond to lowest velocities of the trajectories, whereas the middle points correspond to highest velocities of the trajectories. Hence, these features are invariant for the entire set, and are informative for classification purposes. With regards to the tool's orientation, only the angles around the axis

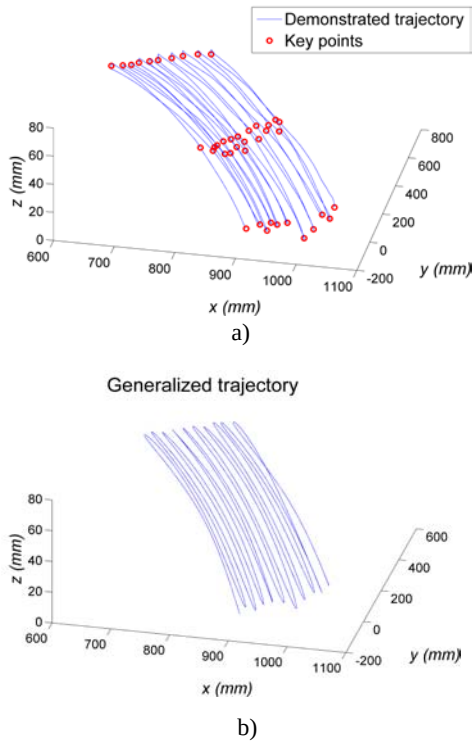

Fig. 3 a) Plot of a sample trajectory for the peening experiment and a set of initially selected key points. b) Generalized trajectory for the peening experiment.

normal to the surface was taken into consideration, since the orientations around the other two axes are almost constant and do not convey important information for classification purposes. The results indicated that the orientation information did not improve the classification rate. In addition, CRF classification with the orientation as only observation feature failed. The generalized trajectory for the case of position and velocity observation features is shown in Fig. 3b. (We opted not to show the demonstrated trajectories in Fig. 3b, since the demonstrations were dissimilar, and it would be impossible to differentiate between the individual demonstrations). For all 4 types of observed features reported in Table 2, HMM generated higher errors in predicting the labels for observed data. Adding the tools' orientation degraded significantly the HMM performance, whereas for positions and velocities features the classification accuracy was improved, but lower when compared to the CRF success rates.

6. DISCUSSION

Implementation of CRF for trajectory generalization in robot PbD involves several challenges.

One is the choice of observed features for classification. The other areas of CRF application, such as part of speech tagging or image segmentation, usually provide a rich set of local features for encapsulating the information within the input data. In the case of trajectories, the main features one can rely on are the positions, orientations and velocities. In [19] CRF was implemented for the tagging game with three robots, with a goal to label the state of each robot during the tagging game. The authors used additional types of features, as velocities threshold, the distance between the robots, the directions between the robots etc. From the presented results in Section 5 one can note that for the peening task the velocity features were sufficient for segmentation of the trajectories. On the other hand, for the painting task the distributions of the velocities were more uniform, and thus less useful for segmentation. However, when combined with the position features, the velocities improved the performance of CRF. To conclude, depending on the application, it is necessary to identify all relevant features which will contribute towards increased classification rate of the algorithm.

The proposed approach of generalization based on key points requires efficient initial identification of candidate key points. The employed LBG technique for automated detection of the key points entails tuning of the number of clusters for trajectories with different levels of complexity. One possible alternative to facilitate this problem is to take advantage of the expertise of the human operator and let him/her select the appropriate choice for the number of clusters. This way, it can be assured that all the relevant features of the trajectories are taken into account. Furthermore, another limitation of the proposed approach is its computational expensiveness, since the adopted cross-validation imposes the training of the model parameters to be performed for each demonstration.

Table 2. Means and standard deviations of the classification rates obtained by CRF and HMM for the peening task.

	CRF	HMM
1. Position	79.06 (± 4.33)	74.99 (± 3.12)
2. Velocity	94.3 (± 1.79)	80.97 (± 1.79)
3. Position, velocity	94.7 (± 1.55)	89.98 (± 1.81)
4. Position, velocity, orientation	93.49 (± 4.37)	64.50 (± 42.62)

7. CONCLUSION

In this work the discriminative character of CRF has been used for trajectory segmentation in a PbD environment. CRF was used for probabilistic representation of demonstrations, leading to spatio-temporal labeling of the observed features across the demonstrated set. An interpolation technique was employed to obtain a generalized trajectory for reproduction of the demonstrated skills.

The approach was employed to simulate transfer of trajectories to a robot for tasks of painting and peening. The results showed that CRF was able to correctly classify about 90 % of the trajectories points. It was concluded that increasing the number of observed features usually improves the CRF performance. However, the most important aspect is the selection of features which convey rich discriminative information for the specific

task. The comparisons with HMM indicated on higher classification rates of CRF for the considered observation features.

ACKNOWLEDGEMENTS

This work is partially supported by the New Initiative Funding program, at National Research Council Canada's Institute for Aerospace Research (NRC-IAR), and Ontario Partnership for Innovation and Commercialization (OPIC). The work is also supported through a collaborative research and development grant from the Natural Sciences and Engineering Research Council of Canada (NSERC) CRDPJ – 350266 – 07.

The shot peening experiment was performed at NRC-IAR facilities.

REFERENCES

[1] L. Rabiner, A tutorial on Hidden Markov Models and selected applications in speech recognition, *Proc. of the IEEE, 77*(2), 1989, 257–286.

[2] S. K. Tso, and K. P. Liu, Demonstrated trajectory selection by Hidden Markov Model, *Proc. of the Int. Conf. on Robotics and Automation*, Albuquerque, NM, 1997, 2713–2718.

[3] S. Calinon, and A. Billard, Stochastic gesture production and recognition model for a humanoid robot, *Proc. of the 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sendai, Japan, 2004, 2769–2774.

[4] J. Aleotti, and S. Caselli, Robust trajectory learning and approximation for robot programming by demonstration, *Robotics and Autonomous Systems, 54*, 2006, 409–413.

[5] A. Irish, I. Mantegh, and F. Janabi-Sharifi, A PbD approach for learning pseudo-periodic robot trajectories over curved surfaces, *Proc. of the IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, Montreal, Canada, 2010, 1425-1432.

[6] S. Calinon, F. Guenter, and A. Billard, On learning, representing and generalizing a task in a humanoid robot, *IEEE Transaction on Systems, Man and Cybernetics, Part B, 37*, 2007 286–298.

[7] A. Coates, P. Abbeel, and A. Y. Ng, Learning for control from multiple demonstrations, *Proc. of the 25 Int. Conf. on Machine Learning*, Helsinki, Finland, 2008, 141–151.

[8] C. G. Atkenson, A. W. Moore, and S. Schaal, Locally weighted learning for control, *Artificial Intelligence Review, 11*, 1997, 75–113.

[9] S. Vijayakumar, and S. Schaal, Locally weighted projection regression: An O(n) algorithm for incremental real time learning in high dimensional space, *Proc. of Seventeenth Int. Conf. on Machine Learning*, San Francisco, CA, 2000, 1079–1086.

[10] J. Lafferty, A. McCallum, and F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, *Proc. of the Int. Conf. on Machine Learning*, 2001, 282–289.

[11] C. Sutton, A. McCallum, *An Introduction to conditional random fields for relational learning* (in *Introduction to Statistical Relational Learning*, eds. L. Getoor, and B. Taskar, MIT Press, 2006).

[12] F. Sha, and F. Pereira, Shallow parsing with conditional random fields, *Proc. of Human Language Technology Conf. and North American Chapter of the Association for Computational Linguistics*, 2003, 213–220.

[13] R. McDonald, and F. Pereira, Identifying gene and protein mentions in text using conditional random fields, *BMC Bioinformatics, 6(*Suppl 1.S6), 2005.

[14] R. Klinger, L. I. Furlong, C. M. Friedrich, H. T. Mevissen, J. Fluck, F. Sanz, and M. Hofmann-Apitius, Identifying gene specific variations in biomedical text, *Journal of Bioinformatics and Computational Biology, 5*(6), 2007, 1277–1296.

[15] X. He, R. S. Zemel, and M. A. Carreira-Perpinan, Multiscale conditional random fields for image labeling, *Proc. of the 2004 IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, 2004, 695–702.

[16] A. Quattoni, M. Collins, and T. Darrell, Conditional random fields for object recognition (in *Advances in Neural Information Processing Systems*, Cambridge, MA: MIT Press, 2005, 1097–1104).

[17] D. DeCaprio, J. P. Vinson, M. D. Pearson, P. Montgomery, M. Doherty, and J. E. Galagan, Conrad: Gene prediction using conditional random fields, *Genome Research, 17*, 2007, 1389–1396.

[18] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, Conditional models for contextual human motion recognition, *Proc. of the 10th IEEE Int. Conf. on Computer Vision*, Beijing, China, 2005, 1808–1815.

[19] D. L. Vail, M. M. Veloso, and J. D. Lafferty, Conditional random fields for activity recognition, *Int. Conf. on Autonomous Agents and Multi-agent Systems (AAMAS)*, Honolulu, USA, 2007, 235–243.

[20] D. A. Ross, S. Osindero, and R. S. Zemel, Combining discriminative features to infer complex trajectories, *Proc. of the 23rd Int. Conf. on Machine Learning*, Pittsburgh, PA, 2006, 761–768.

[21] H. Kjellstrom, J. Romero, D. Martinez, and D. Kragic, Simultaneous visual recognition of manipulation actions and manipulated objects, in *Lecture Notes in Computer Science, Computer Vision – ECCV*, 5303, 2008, 336–349.

[22] D. Martinez, and D. Kragic, Modeling and recognition of actions through motor primitives, *IEEE Int. Conf. on Robotics and Automation*, Pasadena, USA, 2008, 1704–1709.

[23] A. Vakanski, I. Mantegh, A. Irish, and F. Janabi-Sharifi, Trajectory learning for robot programming by demonstration using Hidden Markov Model and Temporal Dynamic Warping, *Robotics and Autonomous Systems*, submitted for publication.

[24] T. Inamura, H. Tanie, and Y. Nakamura, Keyframe compression and decompression for time series data based on the continuous hidden Markov model, *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, NV, 2003, 1487–1492.

[25] T. Asfour, F. Gyarfas, P. Azad, and R. Dillmann, Imitation learning of dual-arm manipulation tasks in a humanoid robot, *Int. Journal of Humanoid Robotics, 5(*2), 2008, 183–202.

[26] H. Sakoe, and S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, *IEEE Trans. on Acoustics, Speech and Signal Processing, 26*(1), 1978, 43–49.

[27] C. M. Bishop, *Pattern Recognition and Machine Learning* (New York, USA: Springer, 2006, 383–418).

[28] D. L. Vail, J. D. Lafferty, and M. M. Veloso, Feature selection in conditional random fields for activity recognition, *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, San Diego, USA, 2007, 3379–3384.

[29] Y. Linde, A. Buzo, and R. M. Gray, An algorithm for vector quantizer design, *IEEE Trans. on Communications, 28*(1), 1980, 84–95.

[30] M. Schmidt, K. Swersky, and K. Murphy, Conditional Random Field Toolbox for MATLAB, [online] http://www.cs.ubc.ca/ ~murphyk/ Software/CRF/ crf.html, 2010.

[31] J. Yang, Y. Xu, C. S. Chen, Hidden Markov Model approach to skill learning and its application to telerobotics, *IEEE Trans. on Robotics and Automation, 10*(5), 1994, 621–631.