Robotic Learning of Manipulation Tasks from Visual Perception Using a Kinect Sensor

Aleksandar Vakanski, Farrokh Janabi-Sharifi, and Iraj Mantegh

***Abstract*—The paper addresses the problem of transferring new skills to robots from observation of human demonstrated skill examples. An approach is presented for retrieving trajectories of an object, being manipulated during the demonstrations, from Kinect-provided measurements. The problem of object tracking across the image frames is solved by using weighted dynamic template matching with normalized cross-correlation. Such approach takes advantage of the simultaneous image and depth measurements by the Kinect device in leveraging the pattern localization and pose estimation. Demonstrated trajectories are stochastically encoded with hidden Markov model, and the obtained model is exploited for generation of a generalized trajectory for task reproduction. The developed methodology is experimentally validated in a real-world task learning scenario.**

***Index Terms*—Robotics, programming by demonstration, visual learning, pose estimation.**

I. Introduction

The ever-increasing demands for novel robotic applications across different domains, reinforced with the recent progress in the fields of machine learning and artificial intelligence, have led to substantial research in the area of robotic learning systems [1]. A particular class of these robotic systems employs observational learning for acquiring knowledge from demonstrations of tasks executed by another agent [2], [3]. These systems enable quick development of robotic programs in an intuitive manner, and as such have great potential not only for reducing the cost of development of industrial robotic applications, but also for development of service robotic applications. The observation-based robotic systems are most often associated with the robot programming by demonstration (PbD) paradigm [4].

A typical observational learning procedure consists of the following steps: (i) *Task perception*: where the robotic system observes one or multiple task demonstrations and records certain attributes related to the demonstrated motions; (ii) *Task analysis and planning*: the acquired data is processed with an objective of generating a policy for reproduction of the demonstrated task, and (iii) *Task reproduction*: the generated policy is translated into a robot language program and it is deployed on the robotic platform for execution. The step of perception of the demonstrated actions and the states of the world is accomplished by employing different sensing modalities, e.g., electro -magnetic sensors [5], inertial sensors [4], optical marker-based systems [6], vision cameras [7], etc. The focus of this work is on robot learning by using external form of task perception [8], i.e., the sensors are placed externally with respect to the demonstrating agent, as opposed to the perception modes that employ sensing devices mounted directly on the demonstrator's body. The external perceptibility systems relate to the use of vision cameras for observation of demonstrated tasks. For instance, a single vision camera [9], stereo cameras [10], or multiple cameras [11] can be utilized.

On one hand, tasks perception using vision cameras is very challenging, and requires solving difficult problems, such as: detection and identification of objects and/or agents in acquired images; tracking the objects; and, estimation of the spatial pose (position and orientation) of scene objects from two-dimensional images. Consequently, this sensing modality has often been avoided in the robot PbD methodology. However, the steady progress in the field of image processing has lately produced several efficient, fast and fairly robust approaches for object detection and tracking (e.g., adaptive discriminative trackers [12], [13], particle filter tracking [14], TLD framework [15], etc. On the other hand, even if the problem of object tracking during the task perception is performed perfectly, retrieving full six-dimensional poses or trajectories of the scene objects from video images requires additional task knowledge, such as geometric models of the objects or the environment.

The release of Kinect sensor in 2010 have caused momentous advances in the domain of computer vision, where the provision of both visual (RGB) and depth (D) information of the environment enabled a new spectrum of possibilities and applications [16]–[18]. Regarding the robotic observational learning, the abilities of the Kinect sensor for on-line tracking of human motions have been quickly embraced by the research community and utilized in a body of works [19]–[21]. These works employ a skeletal model of human bodies (provided by open source Kinect libraries) for extracting task knowledge in the form of joint trajectories of the demonstrator's limbs, and transferring the task knowledge to a robot for reproduction. However, Kinect libraries for objects tracking are not currently available, and therefore most of the robot learning works using Kinect are focused on imitation of human gestures.

The article studies the problem of robotic learning of tasks that involve manipulation of an object (e.g., a tool or a work piece) for accomplishing the task goals. A single Kinect

Manuscript received October 19, 2013; revised December 26, 2013. This work was supported by the Scientists and Engineers in Business grant, as a joint initiative between Ryerson University and the Federal Economic Development Agency for Southern Ontario.

Aleksandar Vakanski and Farrokh Janabi-Sharifi are with the Department of Mechanical and Industrial Engineering at Ryerson University, Toronto, M5B 2K3, Canada (phone: +1-416-979-5000 x 7089; fax: +1-416-979-5265; e-mail: aleksandar.vakanski@ryerson.ca, fsharifi@ryerson.ca).

Iraj Mantegh is with the National Research Council Canada (NRC) - Aerospace Portfolio, Montreal, H3T 2B2, Canada (e-mail: iraj.mantegh@cnrc-nrc.gc.ca).

sensor is utilized for perception of the demonstrations. Such sensing mode allows unobtrusive object manipulation since it does not require attaching sensors for capturing the demonstrated motions. Differently from the task perception using vision cameras, the Kinect sensor allows extraction of objects' trajectories without any prior knowledge about their geometry. The presented work introduces a novel method for task perception, which employs weighted template matching for detection and tracking of a manipulated object during the task demonstrations. The template weights are assigned on the basis of the range measurements for the corresponding object. The orientation of the object is also estimated using the depth channel information from the Kinect sensor. The task analysis and planning involves task modeling at the trajectory level of abstraction using hidden Markov model (HMM) [22], and generating a trajectory for task reproduction via smoothing spline regression. Experimental validation of the proposed method for observational learning is performed with a CRS-A255 robot.

The rest of the paper is organized as follows: Section II introduces a few basics facts about the Kinect sensor. Section III presents the proposed approach for object tracking and detection for robotic learning. Section IV describes the pose estimation from Kinect measurements. The task modeling and planning is described in Section V. The experimental evaluation is provided in Section VI, and the last section briefly summarizes the paper.

II. Kinect Sensor

The Kinect sensor has a color camera and an infrared camera for acquiring image and range data simultaneously. The maximum rate of data acquisition is 30 frames per second. The default resolution of the RGB sensor is 640×480, whereas the maximum resolution is 1280×960 pixels at a rate of 12 frames per second. Similarly, for the infrared camera, the default resolution is 640×480 at 30 frames per second. The device has also an infrared laser emitter, an array of 4 microphones, and a motor for tilting the unit.

The range data is generated using a structured light pattern that is projected on the environment by the infrared laser projector. Based on the correlation between the emitted and captured light patterns, the distances between the scene points and the infrared camera are calculated.

The Kinect sensor was initially designed as a means of natural user interface in gaming environments for Microsoft's Xbox console. Its wide popularity among the researchers, hobbyist, and the industry, motivated Microsoft to offer it as a stand-alone unit for Windows operating systems, and more importantly, to release a software development kit (SDK). The SDK provides libraries for access to the raw RGB and depth streams, skeletal tracking data, audio noise suppression, integration with the Windows speech recognition API, etc. This incited developers to build a number of new applications for the Kinect sensor.

III. Object Tracking with Kinect

The first step of observational learning pertains to task perception, where an agent demonstrates a task in front of a robot learner, while the robot observes the demonstrations and records the motions and changes in the environment. It is assumed here that the agent is a human demonstrator, and the task involves manipulation of an object. Without loss in generality, the object can represent a tool for performing an industrial task (e.g., painting, welding), an object for accomplishing a service task (e.g., setting dishes on a table), a work piece for an assembly operation, etc.

The task demonstrations are captured solely by a Kinect sensor that is strategically located in the scene, so that during the task demonstrations the objects of interest are visible, i.e., within the field-of-view and free of occlusions.

This section deals with the problem of tracking the object across the sequence of acquired data streams from the Kinect sensor. The presented work here assumes that the learning system does not have any prior information about the manipulated object. For initialization of the tracking procedure, the demonstrator is asked to identify the object of interest in the first image. The learning system should afterwards possess abilities for autonomous detection of the object in the sequence of consecutive images.

The proposed approach employs the method of template matching using normalized cross-correlation (NCC) [23] for localizing a given pattern in an image.

Let I denotes an image of size $m_u \times m_v$, and let the intensity of the pixel with coordinates (u,v) is denoted $I(u,v)$, for $u \in \{1,2,\ldots,m_u\}$, $v \in \{1,2,\ldots,m_v\}$. The template T represents an image patch of size $t_u \times t_v$ (Fig. 1(a), (b)). The NCC at the pixel (p,q) is calculated as

$$r(u,v) = \frac{\sum_p \sum_q \left(I(u+p,v+q) - \bar{I}(u+p,v+q) \right) \sum_p \sum_q \left(T(p,q) - \bar{T} \right)}{\sqrt{\sum_p \sum_q \left(I(u+p,v+q) - \bar{I}(u+p,v+q) \right)^2 \sum_p \sum_q \left(T(p,q) - \bar{T} \right)^2}} \quad (1)$$

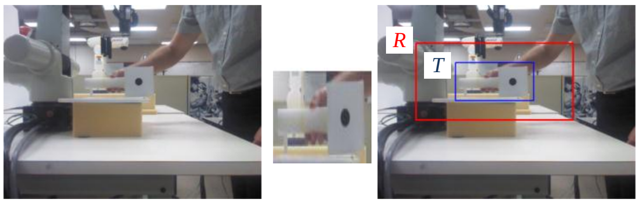
where the summations are performed for $p \in \{1,2,\ldots,t_u\}$, $q \in \{1,2,\ldots,t_v\}$. The notation $\bar{T}$ pertains to the average value of the mask, whereas $\bar{I}$ denotes the average value of the image patch corresponding to the template. The best match for the pattern in the image is at the position with the maximum correlation, i.e.,

$$\left(u_{\text{pos}}, v_{\text{pos}} \right) = \max_{u,v} r(u,v). \quad (2)$$

The normalization of the correlation coefficient in (1), achieved by subtracting the patches means and dividing by the patches standard deviations, enhances the robustness of the matching procedure to changes in brightness and contrast.

However, the above described procedure has several drawbacks, among which are its computational expensiveness and sensitivity to the pattern appearance.

The computational time for performing the template matching can be reduced by performing the procedure only within a region of interest in the image, based on prior information for the estimated location of the pattern. Taking into consideration that in the studied case the images are acquired 30 times per seconds, it is assumed that the object of interest is located in close vicinity of its position in the previous image. Therefore, if $\left(u_{\text{pos}}^{k}, v_{\text{pos}}^{k}\right)$ are the coordinates of the pattern in the image I^{k}, where k is used for indexing the images in the video stream, for finding the best template match in the image I^{k+1}, the NCC in (1) is calculated for an image patch R with size $\theta t_u \times \theta t_v$ that is centered at the coordinate $\left(u_{\text{pos}}^{k+1}, v_{\text{pos}}^{k+1}\right)$. After the maximum correlation between the region R and the template T is found, the location of the origin of the pattern with respect to the image I^{k+1} is retrieved (Fig. 1c). The value of the parameter θ, related to the size of the region of interest, was set equal to 2 in the presented work. Its value is application specific, and it depends on the speed of motion of the target object.



(a) (b) (c)

Fig. 1. (a) An image acquired with Kinect sensor; (b) Selected template that contains the target object; (c) The boundaries of the template *T* and the region of interest *R* are represented in the image *I* with blue and red rectangles, respectively.

As mentioned earlier, the template matching algorithm has strong dependence on the appearance of the pattern in the image, and it can fail when the scale or orientation of the object of interest change. This problem is handled here by applying dynamic templates [24], i.e., the image patch with the highest correlation coefficient in the current image T^{k}, is employed as a template for the next image I^{k+1}.

One shortcoming of dynamically adapted templates is the drift phenomenon. Namely, since the template patch beside the object of interest contains also pixels that belong to the background (Fig. 1b), the cross-correlation calculation also encompasses the background pixels. Switching the template patches for each consecutive image can cause the object of interest to slowly drift within the template. As the drift errors accumulate, over time the object can be lost partially, or even lost completely in some cases. This phenomenon is more pronounced for scenes with cluttered background, and also when the pixels intensities between the object and the background are similar. In order to tackle the drift problem, this work introduced weighted template matching [23], by using the depth information provided by the Kinect sensor. Consequently, the depth data is utilized to filter out the background pixels from the template, so that only the pixels that correspond to the object of interest are employed for the template matching. To initialize the procedure, the demonstrator selects the target object by clicking in the first image, i.e., I^{1}. Then a mask image is created over the template T^{1}, so that

$$W^{1}\left(p,q\right)=\begin{cases}1, & \text{if } z-\Delta \le D^{1}\left(p,q\right) \le z+\nabla \\ 0, & \text{otherwise}\end{cases} \qquad (3)$$

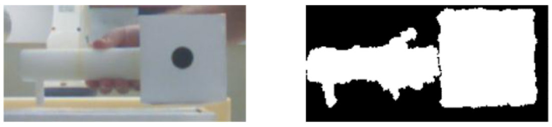

(a) (b)

Fig. 2. (a) A template image; (b) Weights mask, calculated based on the depth data for the template image.

For $p \in \{1,2,\ldots,t_u\}$, $q \in \{1,2,\ldots,t_v\}$, where W^{1} is a matrix of weight coefficients, D^{1} is the matrix of depth values corresponding to the template T^{1}, z is the depth of the selected pixel by the demonstrator, and Δ and ∇ denote lower and upper range of distances, respectively. For illustration, Fig. 2 shows the template image patch T^{1} and the resulting weights image W^{1}.

For the kth image, the weighted normalized cross-correlation is calculated as

$$r(u,v)=\frac{\sum_{p}\sum_{q}W^{k-1}(p,q)\left(R^{k}(u+p,v+q)-\bar{R}^{k}(x+p,y+q)\right)}{\sqrt{\sum_{p}\sum_{q}W^{k-1}(p,q)\left(R^{k}(u+p,v+q)-\bar{R}^{k}(x+p,y+q)\right)^{2}}} \cdot \frac{\sum_{p}\sum_{q}W^{k-1}(p,q)\left(T^{k-1}(p,q)-\bar{T}^{k-1}\right)}{\sqrt{\sum_{p}\sum_{q}W^{k-1}(p,q)\left(T^{k-1}(p,q)-\bar{T}^{k-1}\right)^{2}}}. \qquad (4)$$

Note that the weights matrix W^{k-1} and the template T^{k-1} from the previous frame are used for calculating the maximum correlation in the image R^{k}, and consequently, I^{k}.

The current weights W^{k} are updated at each frame using (3), based on the depth values D^{k}, yielding

$$W^{k}\left(p,q\right)=\begin{cases}1, & \text{if } z-\Delta \le D^{k}\left(p,q\right) \le z+\nabla \\ 0, & \text{otherwise}\end{cases}. \qquad (5)$$

The variable z is also updated at each iteration. For simplicity, an artificial feature is added on the object in the form of a dark circle, and the depth value z is assigned to the pixel that corresponds to the centroid of the circle.

IV. Pose Estimation with Kinect

The pose estimation from images (i.e., calculating the translation and rotation of a frame attached to an object) can be achieved in several different ways. In the case of using a pair of stereo cameras, the pose is estimated by the triangulation technique. In cases when a monocular vision system is used for image acquisition, the pose estimation problem is often solved using homography transformation [25]. Other approaches involve fitting a 3D model of the object to image features [26], reconstructing the relative transformation of an object's pose between consecutive frames based on features correspondence [27], etc. For estimation of the homography matrix, the corresponding image coordinates of at least 4 coplanar points, or 8 non-coplanar points, are required. If the Cartesian distances between the points are known, the full pose can be recovered from a single image; otherwise, a partial pose up to an unknown scalar can be estimated from images. The other approaches [26], [27] also require either a 3D model of the object or an initial estimate of the pose.

The pose estimation problem from Kinect-provided measurements is alleviated, due to the access to both visual and range information of the scene. First, the pose estimation does not require knowledge of the objects geometry, or the scene geometry. Second, the pose estimation does not require extraction of the projections of several object features (at least 4) in the images, and subsequently finding the features correspondences across the image frames. Third, it does not require pose estimation at the initial states of the objects in the scene.

Let assume a coordinate frame *O-xyz* is attached to the object. The distance of the frame origin to the Kinect sensor (*z* coordinate) is read directly from the depth data. The other two coordinates of the frame origin with respect to the camera are easily calculated using the perspective projection equation

$$x = z\frac{\left(u_f - u_0\right)s_u}{f}, \quad y = z\frac{\left(v_f - v_0\right)s_v}{f}, \tag{6}$$

where u_f and v_f are pixel coordinates of the frame origin, f denotes the focal length of the camera, s_u and s_v are the horizontal and vertical pixel sizes, respectively, whereas u_0 and v_0 are used to denote the horizontal and vertical pixel coordinates for the principal point of the image sensor, respectively.

The three coordinates form the Cartesian position of the object frame relative to the camera frame, i.e., $P_o^c = \left[x \quad y \quad z\right]^T$.

The orientation of the object can also be retrieved from the available RBGD information. For that purpose, first, the surface normal of the object at the frame origin is calculated as a cross product of two perpendicular vectors that lie in the plane of the object:

$$\begin{aligned}\mathbf{o}_z = &\left(\mathbf{w}\left(u_f + \varsigma, v_f\right) - \mathbf{w}\left(u_f, v_f\right)\right) \\ &\times\left(\mathbf{w}\left(u_f, v_f - \varsigma\right) - \mathbf{w}\left(u_f, v_f\right)\right)\end{aligned}, \tag{7}$$

where $\mathbf{w}(u,v) = \left[x(u,v) \quad y(u,v) \quad z(u,v)\right]^T$ is a vector in the Cartesian space with coordinates $x(u,v)$ and $y(u,v)$ calculated using (6), $z(u,v)$ is the depth value in D of the corresponding pixel, and ς is an adopted value that represents the distance in pixels in calculating the vectors. The obtained surface normal vector is afterwards normalized to a unit length vector: $\mathbf{o}'_z = \mathbf{o}_z / \left\|\mathbf{o}_z\right\|_2$.

The rotation of the object around the plane of the image sensor can also be extracted from the RGBD data. In the considered case, the orientation of the template weights with respect to the horizontal axis is calculated for this purpose.

Afterwards, the vector is calculated from

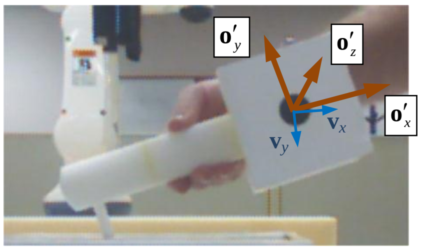


Fig. 3. Orientation vectors for the object. The surface normal vector $\mathbf{o}'_z$ is obtained as a normalized cross product of the vectors $\mathbf{v}_x$ and $\mathbf{v}_y$ (based on a slight violation of the notation in (7)). The vector $\mathbf{o}'_x$ is calculated based on (8), using the weights' axis orientation. The vector $\mathbf{o}'_y$ completes an orthonormal frame with $\mathbf{o}'_z$ and $\mathbf{o}'_x$.

$$\mathbf{o}_x = \mathbf{w}\left(u + \varsigma, v + \varsigma\tan\alpha\right) - \mathbf{w}\left(u, v\right), \tag{8}$$

where α is the angle between the principal axis of the object and the horizontal axis, and ς is an user-selected value for the pixels distance in the vector calculation. The obtained vector is also normalized to a unit length vector, i.e., $\mathbf{o}'_x = \mathbf{o}_x / \left\|\mathbf{o}_x\right\|_2$.

The vector $\mathbf{o}'_y$ is calculated to form a triplet of orthonormal vectors, as a cross product of $\mathbf{o}'_z$ and $\mathbf{o}'_x$.

The vectors are arranged in an orthogonal rotation matrix $R_o^c = \left[\mathbf{o}'_x \quad \mathbf{o}'_y \quad \mathbf{o}'_z\right]$, that gives the rotation of the object in the camera frame (See Fig. 3).

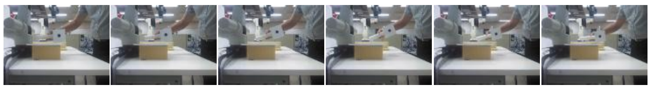
V. Task Modeling, Planning and Reproduction

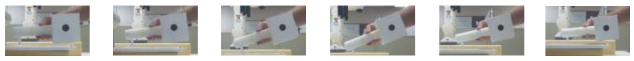
The trajectories of the object from multiple demonstrations of the task under similar conditions are used as input data for generating a policy for task reproduction. For this purpose, the approach we developed in [6] has been employed. It is based on modeling the task demonstrations with HMM by using trajectories key points as relevant task features. The observed variables for the HMM are 12-dimensional vectors, consisting of the poses (6-dimensional) and the corresponding velocities for the captured trajectories. Consequently, the key points are assigned to the trajectories coordinates with significant change in position or velocity. A Bakis left-right topology is employed for the HMM configuration. The Baum-Welch algorithm is used for training the model parameters, whereas the most likely sequence of hidden states is calculated based on the Viterbi algorithm. The trajectories key points are assigned at the transitions between the hidden states. The output key points from the HMM are afterwards temporally aligned along a common time vector using the dynamic temporal warping (DTW) technique [28]. The final step performs smoothing spline regression through the resulting key points for generating a trajectory for task reproduction. The generalized trajectory is translated to a robot executable program and deployed on the robotic platform for task execution.

VI. Experiments

This section is dedicated to experimental validation of the presented robot learning approach.

The task consists of manipulating an object along a desired Cartesian trajectory. It is demonstrated 5 times by a human demonstrator. A Kinect sensor is employed for capturing the demonstrated motions. The acquired data streams are stored in a computer's memory for processing. A sequence of images for one of the demonstrations is displayed in Fig. 4a. The demonstrator indicates the location of the object in the first image of each demonstration using a rectangular selection tool. The demonstrator also clicks once within the circular dot on the object in the first image, to initialize the adaptive template matching process through assignment of weights for the image pattern. A sequence of extracted templates is shown in Fig. 4b, whereas the corresponding weights masks for the templates are presented in Fig. 4c.


(a)


(b)

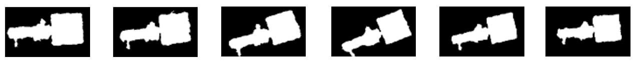
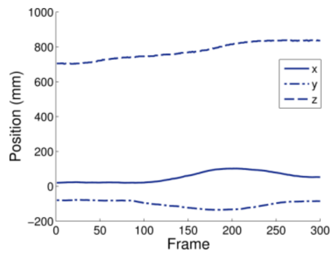
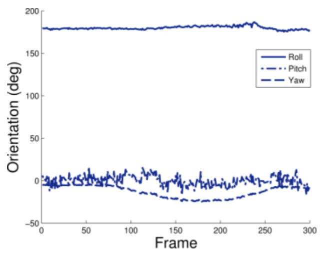

(c)

Fig. 4. (a) Sample images of the task demonstration acquired with Kinect sensor at frames *k* = 32, 95, 142, 189, 237 and 284; (b) Sequence of templates for the images in (a); (c) Corresponding weights masks.



(a)

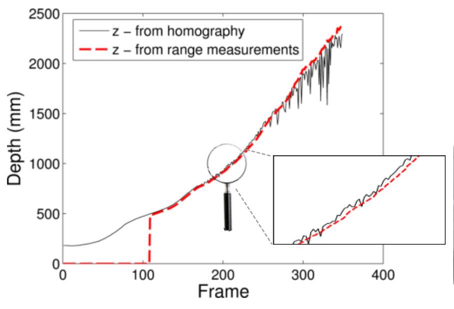


(b)

Fig. 5. (a) Position, and (b) Orientation, of the manipulated object with respect to the camera frame.

The estimation of the pose of the object is based on the described procedure in Section IV. The origin of the object frame is attached to the centroid of the circular feature. The measured position trajectory for one sample demonstration is shown in Fig. 5a. The orientation of the object expressed in Euler's roll-pitch-yaw angles is displayed in Fig. 5b.

One important problem of the Kinect device that is worth mentioning is the inconsistency of the depth data. In particular, the depth maps do not provide measurements for a number of points (which is very pronounced around the objects edges), as well as the measurement noise produce depth fluctuations and incoherent reading among the neighboring pixels. To improve the depth accuracy of the measurements, several filtering techniques has been applied, e.g., bilateral filter [16], spatio-temporal median filter [29], joint-bilateral filter [30]. These approaches rely on the intensity values of the corresponding pixels, as well as on the past temporal information for the depth maps in recovering the missing depth measurements. However, in the considered application, the object of interest is changing its position across the sequence of image frames and therefore the temporal filtering may not improve the depth accuracy. In addition, the most relevant depth information for the considered application relates to the target object. Therefore, a Gaussian filter was employed for the depth data in this work. To smooth the orientation vectors $\mathbf{o}'_x$, $\mathbf{o}'_y$ and $\mathbf{o}'_z$, the calculations in (7) and (8) are averaged for 5 values of the parameter ς, i.e., $\varsigma \in \{2,5,8,14,20\}$.



(a)

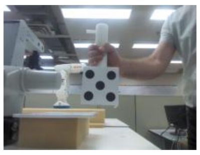
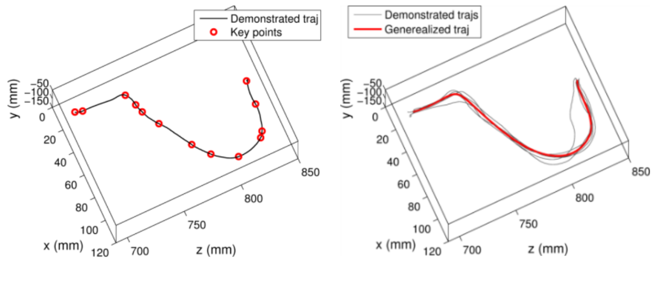

(b)

Fig. 6. (a) Depth measurements provided by the Kinect sensor (dashed line) and corresponding depth values calculated via homography transformation; (b) The 4 outer circular features on the object are employed for calculating the homogrpahy matrix.

Note also that the frame origin can be attached to any feature of the object. Examples are the four corners on the rectangular base of the object, or the top point of the nozzle. These features would be easier to detect and identify across the image frames, based on the significant changes of the neighborhood intensities values. On the other side, the depth measurements are noisy and inconsistent around the edges, thus an interior feature that lies in a locally planar region is preferred for the presented pose estimation.

For comparison, the depth readings from the Kinect sensor are compared to the depth values calculated with homography transformation (Fig. 6a). An object with 4 circular planar features, shown in Fig. 6b, is employed for calculating the depths using homography. The distances between the circular features were measured and used for the calculations. The depth measurements from the Kinect sensor are displayed with dashed (red) line in Fig. 6a, alongside the depths calculated using homography displayed with solid line. One can conclude that the Kinect readings contain less fluctuation, which is especially pronounced for the distance of the object from the camera greater than 1.5 meters. On the other hand, the direct measurements were not available for distances less than 480 millimeters.

The next step entails probabilistic encoding of the task with HMM. A discrete form of HMM is employed for modeling the extracted positions and orientations of the object, by mapping the continuous sequences of observations into a codebook of 128 discrete values. A variant of the Linde-Buzo-Gray algorithm [31] is utilized for the vector quantization. For initialization of the HMM, the minimum distortion trajectory is calculated, and based on the number of regions between the key points of the minimum distortion trajectory, the number of hidden states of the HMM was set equal to 13. Key points are assigned at the transitions between the hidden states of the model. One of the demonstrated trajectories with the assigned key points is shown in Fig. 7a. Temporal normalization of the time indexes for the resulting key points is performed via the multi-dimensional DTW algorithm. The generalized trajectory for task reproduction is obtained by interpolation through the key points from all trajectories using cubic splines. For the Cartesian position, the generalized position trajectory is displayed on Fig 7b, superimposed with the demonstrated trajectories.



(a) (b)

Fig. 7. (a) A demonstrated trajectory with assigned key points by using HMM; (b) The set of demonstrated trajectories (black lines) and the generalized task trajectory (red line) obtained by the described approach.

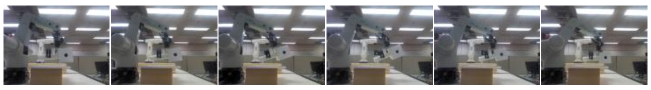


Fig. 8. A sequence of images from the task reproduction by CRS A255 robot.

The generalized trajectory is afterwards transferred to a CRS A255 robot for task reproduction. The A255 robot has 5 degrees-of-freedom, and therefore, only the roll and yaw angles are taken into account for the task execution. A sequence of images of the task execution by the CRS robot is displayed in Fig. 8.

VII. Summary

The work presents an approach for robotic learning based on task abstraction at a trajectory level. A Kinect sensor is employed here for visual observations of multiple task demonstrations performed by a human task expert. The work focuses on the task perception component of the trajectory learning problem, i.e., object detection and tracking, and pose estimation.

The novelty of the presented methodology is in exploiting the combined color and range structure of the Kinect measurements in solving these problems. A dynamic template matching is used for detecting the location of a manipulated object in the video streams. Weight masks are introduced for filtering the object pixels from the background, based on the depth measurements of the scene. The pose estimation from Kinect data is greatly simplified, and does not require a priory knowledge of the object geometry. The only provided information is the initial selection of a single appearance of the object of interest using a rectangle and a single click on the object (to help in differentiating between the object and the background). The presented method performs well in cluttered environments, and it is robust to color/texture fluctuation. Additionally, the object tracking does not require extraction of a set of features and finding their correspondences across the image frames.

The extracted object trajectories are probabilistically modeled and used to calculate a generalized task model, which is transferred to a robot learner for execution.

References

[1] G. Biggs and B. MacDonald, "A survey of robot programming systems," in *Proc. Australasian Conf. Robotics Automation*, Brisbane, Australia, 2003, pp. 1–10.

[2] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot programming by demonstration," in *Handbook of Robotics*, B. Siciliano, and K. Oussama, Eds. Berlin, Germany: Springer-Verlag, 2008, ch. 59.

[3] S. Schaal, A. Ijspeert, and A. Billard, "Computational approaches to motor learning by imitation," *Philosophical Trans. Royal Society of London. Biological Sciences*, vol. 358, no. 1431, pp. 537–547, 2003.

[4] S. Calinon, *Robot Programming by Demonstration: A Probabilistic Approach*, Boca Raton, USA: EPFL/CRC Press, 2009.

[5] R. Dillmann, "Teaching and learning of robot tasks via observation of human performance," *Robotics and Autonomous Systems*, vol. 47, no. 2–3, pp. 109–116, 2004.

[6] A. Vakanski, I. Mantegh, A. Irish, and F. Janabi-Sharifi, "Trajectory learning for robot programming by demonstration using hidden Markov model and dynamic time warping," *IEEE Trans. Systems, Man, Cybernetics - Part B: Cybernetics*, vol. 44, no. 4, pp. 1039–1052, 2012.

[7] H. Kjellstrom, J. Romero, D. Martinez, and D. Kragic, "Simultaneous visual recognition of manipulation actions and manipulated objects," *European Conf. Computer Vision, Part II*, LNCS 5303, pp. 336–349, 2008.

[8] B. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[9] M. Ogino, H. Toichi, Y. Yoshikawa, and M. Asada, "Interaction rule learning with a human partner based on an imitation faculty with a simple visuo-motor mapping," *Robotics and Autonomous Systems*, vol. 56, no. 5, pp. 414–418, 2006.

[10] M. Asada, Y. Yoshikawa, and K. Hosoda, "Learning by observation without three dimensional reconstruction," in *Proc. 6th Intl. Conf. Intelli. Autonomous Systems*, Venice, Italy, 2000, pp. 555–560.

[11] S. Ekvall, D. Aarno, and D. Kragic, "Task learning using graphical programming and human demonstrations," in *Proc. IEEE Int. Symp. Robot Human Interactive Communication*, Hatfield, United Kingdom, 2006, pp. 398–403.

[12] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised on-line boosting for robust tracking," in *Proc. European Conf. Computer Vision, Lecture Notes in Computer Science*, vol. 5302, Marseille, France, 2008, pp. 234–247.

[13] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Miami, USA, 2009, pp. 983–990.

[14] M. D. Breitenstein, F. Reichlin, B. Leibe, E. Koller-Meier, and L.V. Gool, "Robust tracking-by-detection using a detector confidence particle filter," in *Proc. IEEE Intl. Conf. Computer Vision*, Kyoto, Japan, 2009, pp.1515–1522.

[15] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking – Learning – Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 6, no. 1, pp. 1409–1422, 2012.

[16] R. Newcombe, S. Izadi, O. Hilliges et *al.*, "Kinect fusion: Real-time dense surface mapping and tracking," in *IEEE Intl. Symp. Mixed and Augmented Reality*, Bazel, Switzerland, 2011, pp. 127–136.

[17] R.A. Clark, Y.-H. Pua, K. Fortin *et al.*, "Validity of the Microsoft Kinect for assessment of postural control," *Gait & Posture*, vol. 36, no. 3, pp. 372–377, 2012.

[18] R. Held, A. Gupta, B. Curless, and M. Agrawala , "3D puppetry: a Kinect-based interface for 3D animation," in *Proc. Symp. User Interface Software and Technology*, Cambridge, USA, 2012, pp. 423–434.

[19] V.V. Nguyen and J.-H. Lee, "Full-body imitation of human motions with Kinect and heterogeneous kinematic structure of humanoid robot," in *Proc. IEEE Intl. Symp. System Integration*, Fukuoka, Japan, 2012, pp. 93–98.

[20] F. Zuher and R. Romero, "Recognition of human motions for imitation and control of a humanoid robot," in *Proc. Latin American Robotic Symp.*, Ceara, Brazil, 2012, pp. 190–195.

[21] J. Lambrecht, M. Kleinsorge, and J. Kruger, "Markerless gesture-based motion control and programming of industrial robots," in *Proc. IEEE*

Conf. Emerging Technologies & Factory Automation, Toulouse, France, 2011, pp. 1–4.

[22] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," in *Proc. IEEE*, vol. 77, no. 2, 1989, pp. 257–286.

[23] B.D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Intl. Joint Conf. Artificial Intelligence*, Vancouver, Canada, 1981, pp. 674–679.

[24] J. Shi, and C. Tomasi, "Good features to track," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Seattle, USA, 1994, pp. 593–600.

[25] O. Faugeras, *Three-dimensional Computer Vision: A Geometric Viewpoint*, Cambridge, USA: MIT Press, 1993.

[26] P. David, D. DeMenthon, R. Duraiswami, and H. Samet, "SoftPOSIT: Simultaneous pose and correspondence determination," *Int. Journal Computer Vision*, vol. 59, no. 3, pp. 259–284, 2004.

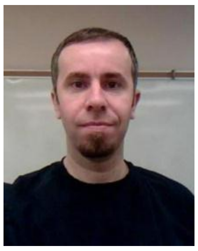
[27] M. A. Fischler and R. C. Bolles, "Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications Association for Computing Machinery*, vol. 24, no. 6, pp. 381–395, 1981.

[28] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-26, no. 1, pp. 43–49, 1978.

[29] S. Matyunin, D. Vatolin, Y. Berdnikov, and M. Smirnov, "Temporal filtering for depth maps generated by Kinect depth camera," in *Proc. 3DTV Conf.: The True Vision - Capture, Transmission and Display of 3D Video*, Antalya, Turkey, 2011, pp. 1–4.

[30] M. Camplani and L. Salgado, "Efficient spatio-temporal hole filling strategy for Kinect depth maps," in *Proc. Intl. Conf. 3D Image Processing and Applications*, San Francisco, USA, 2012, pp. 1–10.

[31] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Communications*, vol. COM-28, no. 1, pp. 84–95, 1980.



Aleksandar Vakanski is a postdoctoral fellow with the Department of Mechanical and Industrial Engineering at Ryerson University, Toronto. He also received his Ph.D. degree from Ryerson University in 2013. He obtained his B.Eng. and M.A.Sc. degrees in mechanical engineering from UKIM University, Macedonia in 1998 and 2003, respectively. His research interests include cognitive robotics, visual servoing, artificial intelligence, and control systems. In 2009 he worked as a visiting student at the National Research Council Canada – Institute for Aerospace Research (NRC-IAR) in Montreal.

Farrokh Janabi-Sharifi is a professor of mechanical-industrial engineering and the director of Robotics, Mechatronics and Manufacturing Automation Laboratory (RMAL) at Ryerson University, Toronto, Canada. He received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Canada in 1995. He was NSERC postdoctoral fellow and instructor in the Center for Intelligent Machines and Department of Electrical-Computer Engineering of McGill University (1995-1997). His research interests span over optomechatronic systems with the focus on image-guided control and planning of robots. He has been a visiting professor in KAIST, Taejon, Korea; IRISA-INRIA, Rennes, France; and TUM, Munich, Germany. He has also been Organizer and/or Co-organizer of several international conferences on optomechatronic systems control. He was General Chair of 2010 International Symposium on Optomechatronic Technologies, Toronto, Canada. He currently serves as the associate editor of several journals including *International Journal of Optomechatronics.*

Iraj Mantegh is a senior research officer and project leader at the National Research Council Canada. He joined NRC's Institute of Aerospace Research in 2003, and pursued his industrial research interests in the areas of manufacturing process control, robot task planning and learning, and mobile manipulators. He obtained his Ph.D in 1998 from the University of Toronto and later received an MBA from Dessautel's School of Management. He is an adjunct professor at Ryerson University and affiliate of several other Canadian universities**.**