Title	Motion Planning from Demonstrations and Polynomial Optimization for Visual Servoing Applications
Author(s)	Shen, T; Radmard, S; Chan, A; Croft, E; Chesi, G
Citation	IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3-7 November 2013. In I E E E International Conference on Intelligent Robots and Systems Proceedings, 2013, p. 578-583, articel no. 6696409
Issued Date	2013
URL	http://hdl.handle.net/10722/199362
Rights	I E E E International Conference on Intelligent Robots and Systems Proceedings. Copyright © I E E E.

2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Motion Planning from Demonstrations and Polynomial Optimization for Visual Servoing Applications

Tiantian Shen[1], Sina Radmard[2], Ambrose Chan[2], Elizabeth A. Croft[2] and Graziano Chesi[1]

Abstract— **Vision feedback control techniques are desirable for a wide range of robotics applications due to their robustness to image noise and modeling errors. However in the case of a robot-mounted camera, they encounter difficulties when the camera traverses large displacements. This scenario necessitates continuous visual target feedback during the robot motion, while simultaneously considering the robot's self- and external-constraints. Herein, we propose to combine workspace (Cartesian space) path-planning with robot teach-by-demonstration to address the visibility constraint, joint limits and "whole arm" collision avoidance for vision-based control of a robot manipulator. User demonstration data generates safe regions for robot motion with respect to joint limits and potential "whole arm" collisions. Our algorithm uses these safe regions to generate new feasible trajectories under a visibility constraint that achieves the desired view of the target (e.g., a pre-grasping location) in new, undemonstrated locations. Experiments with a 7-DOF articulated arm validate the proposed method.**

I. INTRODUCTION

Eye-in-hand visual servoing (VS) systems incorporate a vision sensor mounted directly on the robot end-effector for the task of steering the end-effector from an initial location to a desired one identified by image features provided in advance. Established VS methods include position-based visual servoing (PBVS) [1] and image-based visual servoing (IBVS) [2]. Combinations of IBVS and PBVS have also been explored, for example, 2 1/2-D visual servoing [3], partition of the degrees of freedom [4] and switching controllers [5]. Other approaches include: navigation functions [6], path-planning techniques [7], etc. Surveys of the work in this area can be found in [8]–[10].

IBVS approaches have been popular due to their robustness to image noise and modeling errors. In these approaches, the visibility constraint is an essential requirement to avoid servo failure, while robot's joint limits and "whole-arm" collisions must also be avoided. Most motion planning methods (e.g, [11], [12]) manage constraints in joint space and Cartesian space without taking into account visual feedback. Mezouar and Chaumette [7] developed a potential field based path-planning strategy for robust image-based control in the sense of fulfilling on-line FOV constraints and joint limits. In [13], polynomial parameterization of the workspace model facilitated avoidance of joint, FOV and end-effector collision constraints; however, "whole arm" collisions were not considered and a detailed knowledge of the environment including the obstacles was required a priori. To avoid exhaustive modeling and consider "whole arm" collisions, a teach-by-demonstration approach was presented in [14] for constrained manipulator visual servoing (CMVS). This approach servos the robot end-effector to an untaught target location, considering "whole-arm" collisions, robot's joint limits and self-occlusion; however, the visibility constraint could be violated and result in failure of VS task, upon which the controller fell back to "blind" joint space control.

This paper proposes a constrained VS scheme to achieve the convergence of the image trajectory to a desired view of a repositioned target (i.e., a pre-grasping location). These constraints, which encompass "whole arm" collisions, joint limits and visibility constraint, are simultaneously realized by combining robot teach-by-demonstration samples and path-planning techniques in the robot's workspace. Robot teach-by-demonstration helps to define safe regions for the robot motion without the need of expensive mapping (especially in case of having a single eye-in-hand vision sensor). This approach is very appealing in cluttered environments, where the chance of collision and occlusion is high. A trained user guides the robot through a few motion demonstrations towards a general target location while avoiding joint limits and obstacles (without strictly considering visibility constraints at this stage). These demonstrations define safe regions via an average trajectory and its time-dependent covariance matrices in joint space and later in workspace. The final camera pose that achieves the reference view is first estimated. Between the initial and the estimated final camera poses, a set of control points that meet all of the constraints are selected in safe regions where the largest covariance values are detected. These control points are connected by a complete camera trajectory that is modeled and optimized under visibility constraint by polynomials with C^2 continuity. We validate this approach by steering a redundant manipulator along the corresponding joint trajectories that are obtained through a weighted transition from the planned camera trajectory.

The paper is organized as follows. Section II describes the robot teach-by-demonstration method and Section III presents an optimization method to plan feasible trajectories. We present experimental and robustness results for our algorithm in Section IV, and then conclude in Section V.

II. ROBOT TEACH-BY-DEMONSTRATION

A reference image of the target is taken as a prerequisite for robot teach-by-demonstration. Next, the user moves the

[1]T. Shen, G. Chesi are with the Department of Electrical and Electronic Engineering, University of Hong Kong, Hong Kong tiantianshen@gmail.com

[2]S. Radmard, A. Chan, E. A. Croft are with the Department of Mechanical Engineering, University of British Columbia, Canada s.radmard85@gmail.com

robot arm towards different target locations several times, so that we can extract statistical information about the demonstrated joint trajectories. Variations among the demonstrated joint trajectories approximate how closely the robot should track a given trajectory. Specifically, when the robot is in close proximity to obstacles, we expect this set of joint trajectories to have little variation. When the workspace is relatively free of obstacles, we expect the demonstrated trajectories to result in larger variations. Further details are reported in [14].

A. Canonical Time Warping (CTW)

We assume that the trajectory variations follow a Gaussian distribution in time and space. We wish to extract a robust trajectory from a set of demonstrations. In addition, we wish to quantify any variation that may exist between the trajectories. To remove the effect of temporal variations, we use CTW to solve for the best temporal alignment between two trajectories (via dynamic programming [14]) while adhering to temporal precedence and continuity constraints.

B. Gaussian Mixture Model (GMM)

We represent the robot's workspace using a multivariate GMM of M-components with dimensionality $N+1$ (for a robot with N-degrees of freedom and the time index t): $\mathscr{P}(\mathbf{q}(t)) = \sum_{m=1}^{M} \pi_m \mathscr{N}(\mathbf{q}(t); \mu_m, \Sigma_m)$ where π_m is the prior probability on the Gaussian component m, $\mathscr{N}(\mathbf{q}(t), \mu_m, \Sigma_m)$ is the $(N+1)$-dimensional Gaussian density of component m, and μ_m and Σ_m are the mean and covariance matrix, respectively. These parameters are estimated using the Expectation Maximization (EM) algorithm. We separate μ_m and Σ_m into their spatial and temporal constituents:

$$\mu_m = [\mu_m^t, \mu_m^{\mathbf{q}}], \quad \Sigma_m = \begin{bmatrix} \Sigma_m^t & \Sigma_m^{t\mathbf{q}} \\ \Sigma_m^{\mathbf{q}t} & \Sigma_m^{\mathbf{q}} \end{bmatrix}. \tag{1}$$

C. Gaussian Mixture Regression (GMR)

We perform GMR along the time index to reconstruct the average joint trajectory $\bar{\mathbf{q}}(t)$ and its time-dependent covariance matrix $\Sigma^{\mathbf{q}}(t)$:

$$\bar{\mathbf{q}}(t) = \sum_{m=1}^{M} \beta_m(t) \bar{\mathbf{q}}_m(t), \quad \Sigma^{\mathbf{q}}(t) = \sum_{m=1}^{M} \beta_m(t) \Sigma_m^{\mathbf{q}} \tag{2}$$

where

$$\beta_m(t) = \frac{\pi_m \mathscr{N}(t, \mu_m^t, \Sigma_m^t)}{\sum_{j=1}^{M} \pi_j \mathscr{N}(t, \mu_j^t, \Sigma_j^t)}, \tag{3}$$

$$\bar{\mathbf{q}}_m(t) = \mu_m^{\mathbf{q}} + \Sigma_m^{\mathbf{q}t} (\Sigma_m^t)^{-1} (t - \mu_m^t). \tag{4}$$

The average joint trajectory and its time-dependent weighting matrix $\mathbf{W}(t) = (\Sigma^{\mathbf{q}}(t))^{-1}$, which gives a measure of the relative importance of each joint, are sampled p times and denoted as $\bar{\mathbf{q}}_i$ and $\mathbf{W}_i$, $i = 1, \ldots, p$. The cost function,

$$(\mathbf{q}_i - \bar{\mathbf{q}}_i)^\top \mathbf{W}_i (\mathbf{q}_i - \bar{\mathbf{q}}_i), \tag{5}$$

evaluates the covariance-weighted distance between the average joint trajectory $\bar{\mathbf{q}}_i$ and its candidate trajectory $\mathbf{q}_i$.

III. POLYNOMIAL OPTIMIZATION

A feasible conservative domain for the camera translation is found according to the cost function (5) defined in joint space. We first estimate the new untaught target location, and then the final camera pose that gives the desired view of the target. Subsequently, we plan a feasible trajectory between the initial and final camera poses with the help of a set of control points in the feasible domain.

A. Feasible Domain in Workspace

An average camera trajectory is obtained by applying forward kinematics [15] to the average joint trajectories yielding a camera trajectory, $\bar{\mathbf{c}}_i = [\bar{\mathbf{d}}_i^\top, \bar{\mathbf{a}}_i^\top]^\top$, where $\bar{\mathbf{d}}_i \in \mathscr{R}^3$ ($\mathscr{R}$ is the real number set) is camera translation and $\bar{\mathbf{a}}_i \in \mathscr{R}^4$ is quaternions of camera rotation. We rewrite the cost function in joint space (5) as: $(\mathbf{q}_i - \bar{\mathbf{q}}_i)^\top \mathbf{J}^\top \mathbf{J}^{\dagger\top} \mathbf{W}_i \mathbf{J}^\dagger \mathbf{J} (\mathbf{q}_i - \bar{\mathbf{q}}_i)$, where $\mathbf{J}^\dagger$ is the pseudoinverse of the manipulator Jacobian $\mathbf{J}$. When the distance between $\mathbf{q}_i$ and $\bar{\mathbf{q}}_i$ is very small, the cost function is transformed into $(\mathbf{c}_i - \bar{\mathbf{c}}_i)^\top \mathbf{M}_i (\mathbf{c}_i - \bar{\mathbf{c}}_i)$, where $\mathbf{c}_i$ is candidate servoing trajectory of $\bar{\mathbf{c}}_i$ and $\mathbf{M}_i = \mathbf{J}^{\dagger\top} \mathbf{W}_i \mathbf{J}^\dagger$, according to velocity kinematics. Here we focus on the camera translation (mainly dependant on the first three robot joints) in order to avoid collisions; therefore we use the weighting matrix $\mathbf{M}_{\mathbf{d}i} = \mathbf{J}_{\mathbf{v}}^{\dagger\top} \mathbf{W}_i \mathbf{J}_{\mathbf{v}}^\dagger$, where $\mathbf{J}_{\mathbf{v}} \in \mathscr{R}^{3\times n}$ is the upper part of the manipulator Jacobian, to measure the relative importance of the x, y and z-coordinates in camera translation. The minimum eigenvalue of $\mathbf{M}_{\mathbf{d}i}$, denoted as σ_i, serves as a conservative covariance for these three coordinates:

$$\bar{\mathbf{d}}_i - \sigma_i \mathbf{1}_3 < \mathbf{d}_i < \bar{\mathbf{d}}_i + \sigma_i \mathbf{1}_3. \tag{6}$$

Here, $\mathbf{1}_n$ is an $n \times 1$ vector with all unity elements. The above inequality provides a feasible conservative domain over which to adjust the camera location, $\mathbf{d}_i$, while maintaining collision avoidance between the robot arm and obstacles as specified by equations (1)-(4).

B. New Target Position and Final Camera Pose

The initial view of the target in a new untaught location is taken at the beginning of the servoing task. We estimate the target position and relative camera pose [16] based on the initial and desired views of the target. Estimation results are denoted as $\mathbf{R}$ (relative rotation), $\mathbf{d}$ (relative translation) and $\mathbf{h}_j$, $j = 1, \ldots, n$ (target coordinates in the robot base frame, n is the number of points combined as a target). The result of $\mathbf{h}_j$, $j = 1, \ldots, n$ helps to obtain the final camera pose $\mathbf{g}^*$:

$$\begin{aligned} \mathbf{g}^* = \min_{\mathbf{c}} & \sum_{j=1}^{n} \|\mathbf{m}_j(\mathbf{c}) - \mathbf{K}^{-1} \mathbf{p}_j^*\|^2, \\ & s.t. \quad \bar{\mathbf{d}}_p - \sigma_p \mathbf{1}_3 < \mathbf{d} < \bar{\mathbf{d}}_p + \sigma_p \mathbf{1}_3. \end{aligned} \tag{7}$$

The above optimization minimizes the 2D reprojection error on the normalized image plane, where the variable $\mathbf{c}$ is initialized as the last sampled data, $\bar{\mathbf{c}}_p$, on the average camera trajectory, $\mathbf{m}_j(\mathbf{c})$ is the projection of $\mathbf{h}_j$ at camera pose $\mathbf{c}$, $\mathbf{p}_j^*$ is the reference pixel coordinates of $\mathbf{h}_j$, and $\mathbf{K} \in \mathscr{R}^{3\times 3}$ is the camera intrinsic parameters matrix.

C. Control points

A number of control points are selected at locations with large covariance within the feasible domain to facilitate path-planning between the initial and final camera poses. We use $\mathbf{g}_l = [\mathbf{d}_l^\top, \mathbf{a}_l^\top]^\top$, $l = 1, ..., k$ to represent these control points, where the image projection of $\mathbf{h}_j$ is formed as:

$$\begin{aligned} [X_j(\mathbf{g}_l), Y_j(\mathbf{g}_l)]^\top &= [\mathbf{I}_{2\times 2}, \mathbf{0}_2] \frac{\mathbf{K}\mathbf{P}_{lj}}{\mathbf{e}_3^\top \mathbf{P}_{lj}}, \\ \mathbf{P}_{lj} &= [\mathbf{I}_{3\times 3}, \mathbf{0}_3] \begin{pmatrix} \mathbf{R}(\mathbf{a}_l), \mathbf{d}_l \\ \mathbf{0}_3^\top, 1 \end{pmatrix}^{-1} \mathbf{h}_j. \end{aligned} \tag{8}$$

Here, $\mathbf{I}_{n\times n}$ is the $n \times n$ identity matrix, $\mathbf{0}_n$ is the $n \times 1$ null vector, $\mathbf{e}_i$ is the i-th column of 3×3 identity matrix. The pixel coordinates $[X_j(\mathbf{g}_l), Y_j(\mathbf{g}_l)]^\top$ are in the camera FOV if

$$\begin{cases} \varepsilon < X_j(\mathbf{g}_l) < X_{max} - \varepsilon, \\ \varepsilon < Y_j(\mathbf{g}_l) < Y_{max} - \varepsilon, \end{cases} \tag{9}$$

where ε is a positive tolerance in pixels, X_{max} and Y_{max} are the image size in pixels. Initial values of these control points, $\bar{\mathbf{g}}_l = [\bar{\mathbf{d}}_l^\top, \bar{\mathbf{a}}_l^\top]^\top$, $l = 1, ..., k$, are assigned to be the sampling data that correspond to the largest covariance values along the average camera trajectory. From $\bar{\mathbf{g}}_l$, we obtain $\mathbf{g}_l$ as follows:

$$\begin{aligned} if \quad & f(\bar{\mathbf{g}}_l) > 0, \quad \mathbf{g}_l = \bar{\mathbf{g}}_l; \\ otherwise, \quad & \mathbf{g}_l = \arg\min_{\tilde{\mathbf{g}}_l} (-f(\tilde{\mathbf{g}}_l)) \\ & s.t. \quad \bar{\mathbf{d}}_l - \sigma_l \mathbf{1}_3 < \tilde{\mathbf{d}}_l < \bar{\mathbf{d}}_l + \sigma_l \mathbf{1}_3 \\ & \text{and} \quad \|\tilde{\mathbf{a}}_l\| = 1, \end{aligned} \tag{10}$$

where

$$f(\tilde{\mathbf{g}}_l) = \min_{j=1,...,n} \left\{ \min \begin{pmatrix} X_j(\tilde{\mathbf{g}}_l) - \varepsilon \\ Y_j(\tilde{\mathbf{g}}_l) - \varepsilon \\ X_{max} - \varepsilon - X_j(\tilde{\mathbf{g}}_l) \\ Y_{min} - \varepsilon - Y_j(\tilde{\mathbf{g}}_l) \end{pmatrix} \right\}. \tag{11}$$

The updated camera poses in $\mathbf{g}_l$, $l = 1, ..., k$ lie in the safe domain defined by $\bar{\mathbf{d}}_l$ and σ_l, and satisfy the camera FOV limits. In the following section, we use general polynomials with C^2 continuity to connect the initial and final camera poses, $\mathbf{g}^\circ$ and $\mathbf{g}^*$ with these control points as: $\mathbf{G} = \{\mathbf{g}^\circ, \mathbf{g}_1, ...\mathbf{g}_l, ...\mathbf{g}_k, \mathbf{g}^*\}$.

D. General Polynomial

For every segment between two successive camera poses in $\mathbf{G}$, the camera trajectory, $\mathbf{c}$, is modeled by a multi-dimensional polynomial parameterized by $w \in [0, 1]$:

$$\mathbf{c} = \begin{cases} \mathbf{d}(w) = \mathbf{U} \cdot [w^{\delta_u}, \ldots, w, 1]^\top, \\ \mathbf{a}(w) = \mathbf{V} \cdot [w^{\delta_v}, \ldots, w, 1]^\top. \end{cases} \tag{12}$$

Here, $\mathbf{U} \in \mathscr{R}^{3\times(\delta_u+1)}$ and $\mathbf{V} \in \mathscr{R}^{4\times(\delta_v+1)}$ are polynomial coefficients where δ_u and δ_v denote degrees that are not less than five. The parametrization in (12) maintains C^2 continuity by setting the following constraints:

$$\mathbf{U} = \left[\tilde{\mathbf{U}}, \acute{\mathbf{a}}_u, \acute{\mathbf{v}}_u, \acute{\mathbf{p}_u}\right], \tag{13}$$

$$\mathbf{V} = \left[\tilde{\mathbf{V}}, \acute{\mathbf{a}}_v, \acute{\mathbf{v}}_v, \acute{\mathbf{p}}_v\right], \tag{14}$$

$$\tilde{\mathbf{U}} \cdot \mathbf{1}_{(\delta_u-2)} = \grave{\mathbf{p}}_u - \acute{\mathbf{p}}_u - \acute{\mathbf{v}}_u - \acute{\mathbf{a}}_u, \tag{15}$$

$$\tilde{\mathbf{U}} \cdot [\delta_u, \delta_u - 1, \ldots, 3]^\top = \grave{\mathbf{v}}_u - \acute{\mathbf{v}}_u - 2\acute{\mathbf{a}}_u, \tag{16}$$

$$\tilde{\mathbf{U}} \cdot [\delta_u(\delta_u - 1), \ldots, 3 \times 2]^\top = \grave{\mathbf{a}}_u - 2\acute{\mathbf{a}}_u, \tag{17}$$

$$\tilde{\mathbf{V}} \cdot \mathbf{1}_{(\delta_v-2)} = \grave{\mathbf{p}}_v - \acute{\mathbf{p}}_v - \acute{\mathbf{v}}_v - \acute{\mathbf{a}}_v, \tag{18}$$

$$\tilde{\mathbf{V}} \cdot [\delta_v, \delta_v - 1, \ldots, 3]^\top = \grave{\mathbf{v}}_v - \acute{\mathbf{v}}_v - 2\acute{\mathbf{a}}_v, \tag{19}$$

$$\tilde{\mathbf{V}} \cdot [\delta_v(\delta_v - 1), \ldots, 3 \times 2]^\top = \grave{\mathbf{a}}_v - 2\acute{\mathbf{a}}_v, \tag{20}$$

where $\{\acute{\mathbf{p}}_u, \acute{\mathbf{v}}_u, \acute{\mathbf{a}_u}\}$ are the camera position, velocity and acceleration values (PVA) at the beginning of a segment, and $\{\grave{\mathbf{p}}_u, \grave{\mathbf{v}}_u, \grave{\mathbf{a}}_u\}$ are those at the end of the segment. Similarly, $\{\acute{\mathbf{p}}_v, \acute{\mathbf{v}}_v, \acute{\mathbf{a}_v}\}$ and $\{\grave{\mathbf{p}}_v, \grave{\mathbf{v}}_v, \grave{\mathbf{a}}_v\}$ are PVAs of rotation quaternion at the beginning and the end of a segment, respectively. Position values are extracted from camera poses at the control points $\mathbf{G}$. Velocities at these control points are computed from the average camera trajectory. For example, the velocity in x-direction at $\mathbf{g}_l$ is computed by:

$$v_x = \alpha(dx_l/dt), \tag{21}$$

where α is a positive scalar and dx_l/dt is computed from two sample points that are adjacent to $\bar{\mathbf{g}}_l$ on the average camera trajectory. All control point accelerations are set to zero to preserve continuity. Under the restrictions in (13)-(20), the variables $\tilde{\mathbf{U}} \in \mathscr{R}^{3\times(\delta_u-2)}$ and $\tilde{\mathbf{V}} \in \mathscr{R}^{4\times(\delta_v-2)}$ are first initialized as follows:

$$\begin{aligned} &\tilde{\mathbf{U}}^I = \min_{\tilde{\mathbf{U}}} \sum_{i=1}^{a-1} \|\mathbf{d}(w_i) - \mathbf{d}(w_{i+1})\|^2, \\ &s.t. \quad \text{(13)-(20)} \quad \text{and} \quad (\mathbf{d}(w_i) - \bar{\mathbf{d}}_i)^\top \mathbf{M}_{\mathbf{d}i} (\mathbf{d}(w_i) - \bar{\mathbf{d}}_i) < \eta, \end{aligned} \tag{22}$$

$$\tilde{\mathbf{V}}^I = \min_{\tilde{\mathbf{V}}} \sum_{i=1}^{a-1} \|\mathbf{a}(w_i) - \mathbf{a}(w_{i+1})\|^2, \quad s.t. \quad \text{(13)-(20)}, \tag{23}$$

where a is the number of sample points in a segment including the beginning and end points, and $0 \le w_i < w_{i+1} \le 1$. Due to the parametrization in (12), the target coordinates in the current camera frame can also be represented by polynomials in $w \in [0, 1]$ and their polynomial coefficients are functions of the variables $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$. Specifically,

$$[x_j(w), y_j(w), z_j(w)]^\top = [\mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z]^\top \cdot [w^{2\delta_v+\delta_u}, \ldots, w, 1]^\top,$$

where $\mathbf{p}_x$, $\mathbf{p}_y$ and $\mathbf{p}_z$ are computed as:

$$\begin{aligned} \mathbf{p}_x =& \mathbf{o}_2 * [2(\mathbf{v}_2 * \mathbf{v}_3 + \mathbf{v}_4 * \mathbf{v}_1) + \mathbf{o}_3 * [2(\mathbf{v}_3 * \mathbf{v}_1 + \mathbf{v}_4 * \mathbf{v}_2)] \\ &+ \mathbf{o}_1 * (\mathbf{v}_1 * \mathbf{v}_1 + \mathbf{v}_2 * \mathbf{v}_2 - \mathbf{v}_3 * \mathbf{v}_3 - \mathbf{v}_4 * \mathbf{v}_4), \\ \mathbf{p}_y =& \mathbf{o}_1 * [2(\mathbf{v}_2 * \mathbf{v}_3 - \mathbf{v}_4 * \mathbf{v}_1) + \mathbf{o}_3 * [2(\mathbf{v}_3 * \mathbf{v}_4 + \mathbf{v}_2 * \mathbf{v}_1)] \\ &+ \mathbf{o}_2 * (\mathbf{v}_1 * \mathbf{v}_1 - \mathbf{v}_2 * \mathbf{v}_2 + \mathbf{v}_3 * \mathbf{v}_3 - \mathbf{v}_4 * \mathbf{v}_4), \\ \mathbf{p}_z =& \mathbf{o}_1 * [2(\mathbf{v}_3 * \mathbf{v}_1 + \mathbf{v}_4 * \mathbf{v}_2) + \mathbf{o}_2 * [2(\mathbf{v}_3 * \mathbf{v}_4 - \mathbf{v}_2 * \mathbf{v}_1)] \\ &+ \mathbf{o}_3 * (\mathbf{v}_1 * \mathbf{v}_1 - \mathbf{v}_2 * \mathbf{v}_2 - \mathbf{v}_3 * \mathbf{v}_3 + \mathbf{v}_4 * \mathbf{v}_4), \end{aligned}$$

where $\mathbf{v}_i$ is the i-th row in $\mathbf{V}$ and $\mathbf{o}_i$ is the i-th row of $\mathbf{O} = [\mathbf{0}_{3\times\delta_u}, \mathbf{h}_j] - \mathbf{U}$ with $\mathbf{0}_{3\times\delta_u}$ to be an $3 \times \delta_u$ zero matrix. The

image projections of points are required to remain in the camera FOV:

$$\begin{cases} z_j(w) > 0, \\ \varepsilon < f\dfrac{x_j(w)}{z_j(w)} + \dfrac{X_{max}}{2} < X_{max} - \varepsilon, \\ \varepsilon < f\dfrac{y_j(w)}{z_j(w)} + \dfrac{Y_{max}}{2} < Y_{max} - \varepsilon. \end{cases} \tag{24}$$

Here, $z_j(w)$ is the depth of the j-th point, f is the camera focal length. The constraints in (24) are transformed into the inequalities $s_l(j,w) > 0$, $l = 1,...5$, $w \in (0,1)$, where,

$$\begin{cases} s_1(j,w) = z_j(w), \\ s_2(j,w) = [(X_{max}/2) - \varepsilon]z_j(w) - fx_j(w), \\ s_3(j,w) = [(X_{max}/2) - \varepsilon]z_j(w) + fx_j(w), \\ s_4(j,w) = [(Y_{max}/2) - \varepsilon]z_j(w) - fy_j(w), \\ s_5(j,w) = [(Y_{max}/2) - \varepsilon]z_j(w) + fy_j(w). \end{cases} \tag{25}$$

All of the above functions are also polynomials in w. From $s_l(j,w)$ we define:

$$s = \min_{l=1,...,5} \left\{ \min_{j=1,...,n} \left(\min_{w\in(0,1)} [s_1(j,w)] \right) \right\}. \tag{26}$$

The value of s is expected to be positive. When the initial values of $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ result in non positive values of s, we update values of $\tilde{\mathbf{U}}$ and $\tilde{\mathbf{V}}$ by the following optimization:

$$\begin{aligned} &\text{if} \quad s \le 0, \quad \{\tilde{\mathbf{U}}^*, \tilde{\mathbf{V}}^*\} = \arg\min_{\tilde{\mathbf{U}},\tilde{\mathbf{V}}} (-s) \quad s.t. \quad (13)\text{-}(20), \\ &\text{otherwise}, \quad \tilde{\mathbf{U}}^* = \tilde{\mathbf{U}}^I, \quad \tilde{\mathbf{V}}^* = \tilde{\mathbf{V}}^I. \end{aligned} \tag{27}$$

It is notable that the constraint in (19) is relaxed here allowing the quaternion velocities to vary at the end of a segment. These velocities are computed with the updated value of $\tilde{\mathbf{V}}$ as constraint for the next segment. The values of $\tilde{\mathbf{U}}^*$ and $\tilde{\mathbf{V}}^*$ for all of the segments constitute the feasible camera trajectory.

E. Weighted Transition

For highly articulated (redundant) robotic arms, it is necessary to ensure that the joint space trajectory satisfies the "whole arm" collision constraints since inverse kinematics can provide infinite solutions. This is achieved through transforming the planned camera trajectory into joint space by the minimization:

$$\min_{\mathbf{q}_i} \left(\|\mathbf{c}_i(\tilde{\mathbf{U}}^*, \tilde{\mathbf{V}}^*) - \mathbf{c}(\mathbf{q}_i)\|^2 + (\mathbf{q}_i - \bar{\mathbf{q}}_i)^\top \mathbf{Q}^\top \mathbf{W}_i \mathbf{Q} (\mathbf{q}_i - \bar{\mathbf{q}}_i) \right), \tag{28}$$

where

$$\mathbf{Q} = \begin{pmatrix} \mathbf{I}_{3\times3} & \mathbf{0}_{3\times4} \\ \mathbf{0}_{4\times3} & \mathbf{0}_{4\times4} \end{pmatrix}. \tag{29}$$

Here, $\mathbf{c}_i(\tilde{\mathbf{U}}^*, \tilde{\mathbf{V}}^*)$ represents sample points on the planned camera trajectory, $\mathbf{c}(\mathbf{q}_i)$ is the camera pose associated with joint configuration $\mathbf{q}_i$. In the objective function in (28), the Cartesian space term deals with the end effector pose planning (satisfying the FOV constraint), while the joint space term provides elbow configuration planning (satisfying "whole arm" collision constraint). The initial value of $\mathbf{q}_i$ is assigned to be $\bar{\mathbf{q}}_i$ that represents the average joint trajectory.

IV. RESULTS

A. Experimental Set-up

We implement our proposed algorithm on a 7-DoF articulated arm (Barrett Technology WAM). A CCD camera is mounted on the arm to servo to a target composed of 12 points. A rectangular plate is placed in the robot's workspace representing an obstacle/potential occlusion.

B. Experimental Procedure

The experimental procedure for a specific scenario is as follows: 1) the robotic arm is taught feasible trajectories by the user to achieve a desired view of the target that varies in every demonstration. 2) An average joint trajectory and related covariance matrices are extracted from the recorded joint trajectories (as stated in Section II). 3) The average joint trajectory and related covariance matrices are transformed into workspace coordinates (Section III-A). 4) The target is placed in a new location and an initial view of the target is captured to estimate the target position and final camera pose (Section III-B). 5) A number of control points are inserted in between the initial and final camera poses (Section III-C). 6) A camera trajectory is modeled and planned (applying method stated in Section III-D) with the help of information acquired from demonstrations and control points. Planned image trajectories are restricted within an image size of 1024×768 pixels. 7) Finally, joint trajectories are obtained by weighted transformation from the planned camera trajectory for this specific redundant manipulator.

C. Experimental Results

In the experimental scenario, the robot initial configuration is far from any joint limit, and the relative target location seems tractable. As a result, the user finds it trivial to demonstrate a servoing task. In this example, the user teaches the robot seven times to achieve a reference image of the target at different locations each time, as shown in Fig. 1. The

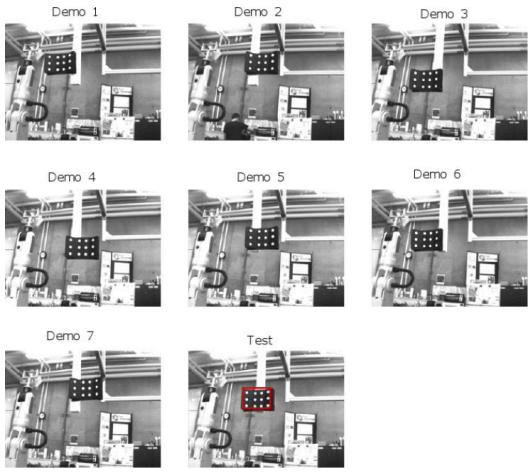

Fig. 1. Initial views of the target at various locations. The red rectangle demonstrates the moving region of the target center.

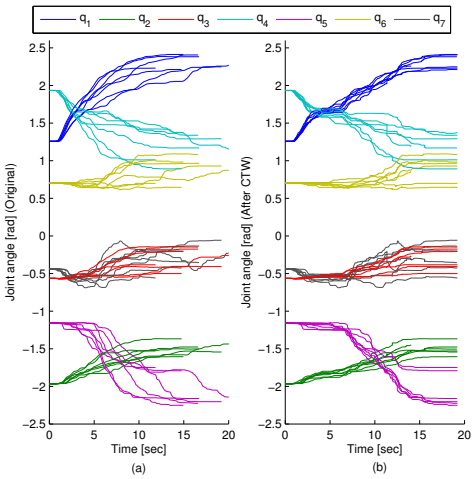

Fig. 2. User-demonstrated joint-space trajectories. (a) Original. (b) After temporal alignment with CTW.

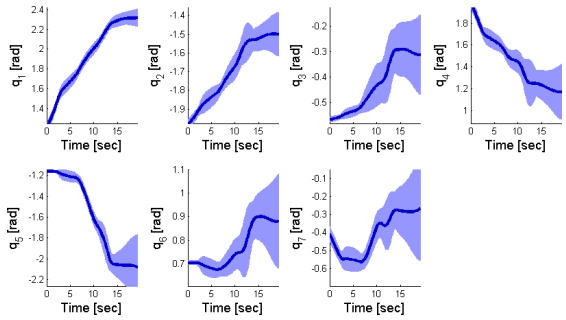

Fig. 3. Joint-space average trajectory and its time-dependent variance obtained from GMR.

training data (joint trajectories) of all of the demonstrations are provided in Fig. 2 (a)-(b), before and after their temporal alignment. Fig. 3 shows the average joint space trajectories and their time-dependent variance over the training data obtained from GMR. Initial and desired target views at its new/untaught location are presented in Fig. 4 (a). Apart from the initial and final points, seven control points are considered and the tolerance value for the FOV limits in (11) is selected as $\varepsilon = 10$ pixels. Fig. 5 (a) shows these initial control points along the average trajectory ($\bar{\mathbf{g}}_l$ in Section III-C). The optimized control points obtained by (10), $\mathbf{g}_l$ are also shown as new control points. Velocities at these control points are computed using (21) with $\alpha = 2$. We used $\delta_u = 5$ and $\delta_v = 6$ polynomial parametrization of camera translation and rotation quaternion in (12) and (13). Initialization of polynomial coefficients in (22) is constrained by parameter $\eta = 10^{-3}$. Fig. 5 (a) shows the planned path.

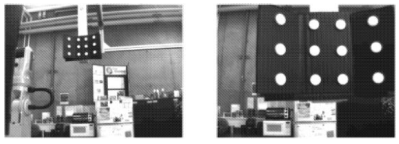

Fig. 4. The initial (left) and desired (right) target views at its new/untaught position.

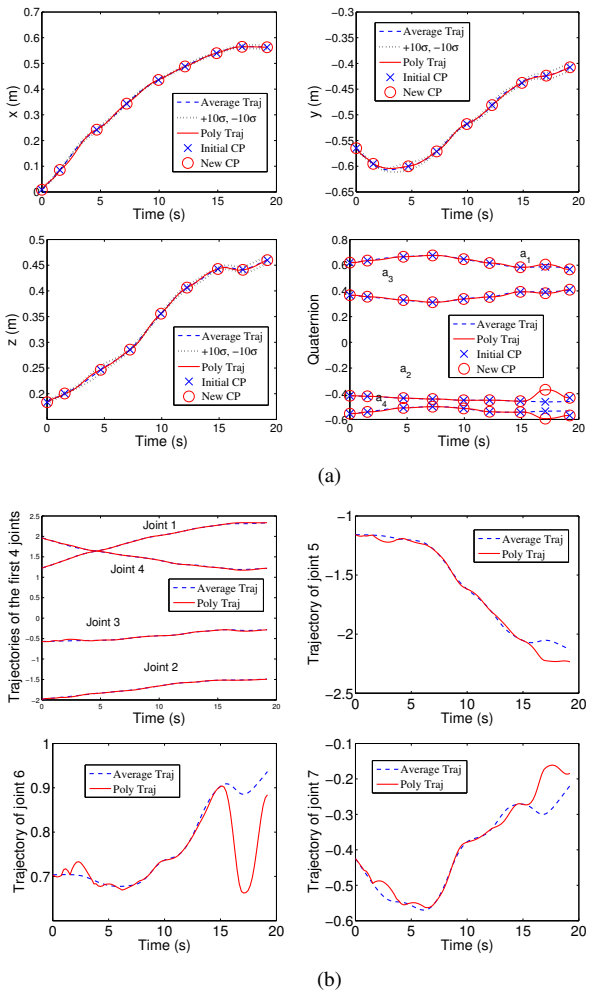

Fig. 5. (a) Planned camera trajectories expressed in the robot base frame. (b) Corresponding joint trajectories obtained by weighted transformation.

For this redundant manipulator, a weighted transformation is used to obtain seven joint trajectories. In Fig. 5 (b), the average joint trajectories are represented by dots and planned joint trajectories by lines. If the robot follows the average camera trajectory, the image trajectory of the target crosses the image boundary (FOV violation), as shown by in Fig 6 (a). Image trajectories extracted from the real-time video stream of the mounted camera following the planned polynomial path are displayed in Fig. 6 (b). As captured, the target image trajectories stay within the FOV boundaries. Fig. 6 (c) shows the initial robot configuration, while Fig. 6 (d) shows the final robot configuration after completion of visual servoing task.

To evaluate the robustness of the proposed approach, 10000 new target locations were randomly generated in simulation with their centres bounded by a rectangle (see Fig. 1). This rectangle encompasses the centres of all of the target locations used during teach-by-demonstration with approximate dimensions $l \times h \approx 17 \times 13$cm, see Fig. 7. The proposed algorithm was run for each of the 10000 target locations, and failures were reported in any of the following circumstances: 1) the initial view of the re-located target is

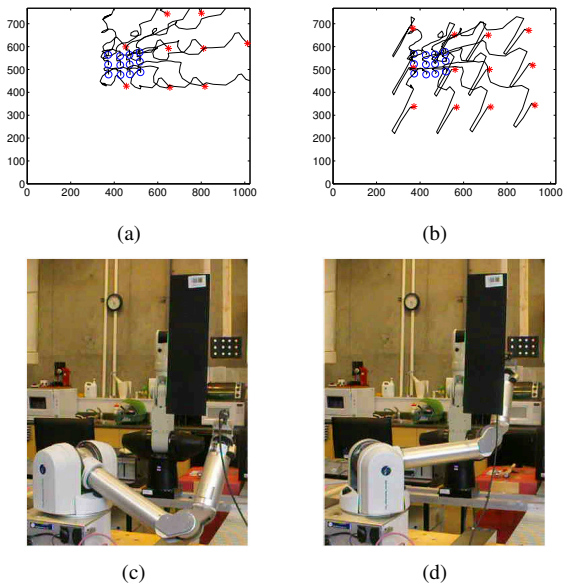

Fig. 6. (a) Image trajectory of the target when the robot follows the average joint trajectory. (b) Extracted image trajectory from experimental video stream following polynomial path. (c) Initial robot configuration. (d) Final robot configuration after completion of visual servoing task.

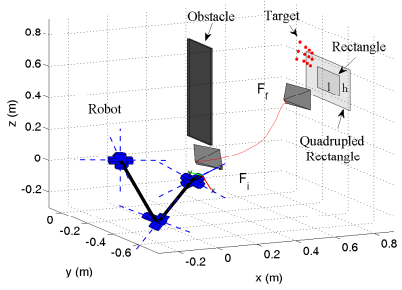

Fig. 7. Assumed rectangle in which target locations are randomly generated in simulation. Dimensions $l \times h \approx 17 \times 13$cm are determined by target locations used for teach-by-demonstration. (F_i and F_f refer to the initial and final camera frames)

unavailable with $f(\mathbf{g}^\circ) \leq 0$ in (11); 2) target view is lost at the estimated final camera pose with $f(\mathbf{g}^*) \leq 0$ in (11); 3) target is out of camera FOV at any inserted control point with $f(\mathbf{g}_l) \leq 0$ in (10); 4) FOV violation happens between any two successive camera poses in $\mathbf{G}$ with $s \leq 0$ in (27). Next, we varied the dimensions of the bounding rectangle (about the centre of the target area) to embrace different sets 10000 randomly generated target locations. The results are presented in Table I. The algorithm can successfully find a collision free visually unimpaired path in over 97% of the trials when the bounding rectangle is the original (demonstration) target area and over 91% of trials even when the target area is extended well beyond the taught area.

V. CONCLUSIONS

This paper provides a complementary approach that combines robot teach-by-demonstration and path-planning techniques to ensure the convergence of image trajectories to a desired view of a relocated target. This method allows the robot to travel in safe regions defined in joint space, while keeping a relocated target in the camera FOV. Future work will focus on on-line servoing and improved formulation of the feasible domain in workspace.

TABLE I

PERCENTAGE OF ALGORITHM FAILURES FOR RANDOMLY GENERATED TARGETS OVER VARIOUS BOUNDED RANGES

	% Failure for 10000 trials
One Quarter of Target Area	1.14%
Half of Target Area	1.27%
Original Target Area (17 × 13 cm)	2.96%
Doubled Target Area	7.33%
Quadrupled Target Area	8.18%

VI. ACKNOWLEDGMENTS

The authors thank the Associate Editor and the reviewers for their useful comments. This work is supported in part by Grant HKU711213E and by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

[1] C. Taylor and J. Ostrowski, "Robust vision-based pose control," in *ICRA*, San Francisco, CA, 2000, pp. 2734–2740.
[2] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator control with image-based visual servo," in *ICRA*, San Francisco, CA, 1991, pp. 2267–2272.
[3] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 d visual servoing," *IEEE Trans. on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, April 1999.
[4] P. Y. Oh and P. K. Allen, "Visual servoing by partitioning degrees of freedom," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 1, pp. 1–17, 2001.
[5] G. Chesi, K. Hashimoto, D. Prattichizzo, and A. Vicino, "Keeping features in the field of view in eye-in-hand visual servoing: A switching approach," *IEEE Trans. on Robotics*, vol. 20, no. 5, pp. 908–914, 2004.
[6] N. Cowan, J. Weingarten, and D. Koditschek, "Visual servoing via navigation functions," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 4, pp. 521–533, 2002.
[7] Y. Mezouar and F. Chaumette, "Path planning for robust image-based control," *IEEE Trans. on Robotics and Automation*, vol. 18, no. 4, pp. 534–549, 2002.
[8] F. Chaumette and S. Hutchinson, "Visual servo control, part I: Basic approaches," *IEEE Robotics and Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.
[9] ——, "Visual servo control, part II: Advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.
[10] G. Chesi and K. Hashimoto, Eds., *Visual Servoing via Advanced Numerical Methods*. Springer, 2010.
[11] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, "Manipulation planning on constraint manifolds," in *ICRA*, Kobe, Japan, 2009, pp. 1383–1390.
[12] A. Dragan, N. Ratliff, and S. Srinivasa, "Manipulation planning with goal sets using constrained trajectory optimization," in *ICRA*, Shanghai, China, May 2011, pp. 4582–4588.
[13] G. Chesi and Y. S. Hung, "Global path-planning for constrained and optimal visual servoing," *IEEE Trans. on Robotics*, vol. 23, no. 5, pp. 1050–1060, 2007.
[14] A. Chan, E. A. Croft, and J. J. Little, "Constrained manipulator visual servoing (cmvs): Rapid robot programming in cluttered workspaces," in *IROS*, San Francisco, California, 2011, pp. 2825–2830.
[15] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. John Wiley and Sons, 2006.
[16] R. I. Hartley, "In defense of the eight-point algorithm," *IEEE Trans. on Pattern Recognition and Machine Intelligence*, vol. 19, no. 6, pp. 580–593, June 1997.