2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Trajectory Specification via Sparse Waypoints for Eye-In-Hand Robots Requiring Continuous Target Visibility

Ambrose Chan, Elizabeth A. Croft, and James J. Little

Abstract—**This paper presents several methods of managing field of view constraints of an eye-in-hand system for vision-based pose control with limited controller input. Herein, the possible inverse kinematic solutions for a desired relative camera pose are evaluated to determine whether the interpolated trajectories satisfy field of view constraints for the target of interest. If no immediately feasible trajectory exists, additional waypoints are specified to guide the robot towards its goal while maintaining visibility. The insertion of an additional visible and feasible waypoint divides the problem into two sub-problems of the same form, but of lesser difficulty by reducing the robot's interpolation distance. *Virtual* image-based visual servoing (IBVS) is used to generate an ideal image trajectory to guide the selection of waypoints. A damped least-squares inverse kinematics solution is implemented to handle robot singularities. The methods are simulated for a CRS-A465 robot with a Sony XC-HR70 camera.**

I. Introduction

In many industrial and biomedical applications, a single camera is mounted on a robot manipulator arm in order to use visual feedback to control the pose of the robot end-effector with respect to a target object. The goal of this pre-positioning task may be, for example, to set up for the grasping of an object, to assemble two mating parts, to precisely insert a needle into a patient, or to visually inspect an object from a particular viewpoint. In contrast to pan-tilt cameras, which only have two degrees of freedom (DoF), the translation of an eye-in-hand camera allows the sensor to avoid occluding objects and to obtain images of the target from multiple viewpoints, resulting in increased robustness to out-of-plane position and orientation uncertainties in the localization of the target. However, using an eye-in-hand robot for relative end-effector control is not without its own challenges. The frequent changes in the pyramid shaped view-space of the camera during robot motion makes it difficult to guarantee that the target will be continuously visible during the entire robot motion. This is undesirable, since many visual algorithms rely on continuous target visibility for frame-to-frame tracking over small changes in the scene, in order to speed up computations and to establish robustness against image noise. Given an estimate of the 3D location of a target object and the appropriate camera-robot model, it is possible to determine which robot configurations keep the target in the field of view [15]. It is much more difficult, however, to find a non-conservative analytical condition to guarantee that the target remains within the camera's field of view *during* the interpolated robot motion between two points spanning large distances. Unlike pan-tilt cameras, the relationships between the image space and robot joint space are highly non-linear for eye-in-hand robots. Careful consideration must be given to the selection of the trajectory waypoints to ensure that the object remains within the camera's field of view not only at each waypoint, but also throughout the interpolated trajectories.

II. Vision-Based Pose control

Recent progress in the field of visual servoing has extended the ability to generate robot motion online based on continuous visual inputs from eye-in-hand cameras. Proponents of image-based visual servoing (IBVS), position-based visual servoing (PBVS), and visual servoing using hybrid 2D-3D features have all devised methods to deal with the limited and changing field of view of eye-in-hand systems. Off-line path planning is typically used with IBVS in order to generate a set of physically valid intermediate images corresponding to quasi-optimal 3D camera viewpoints. In [1][2][6], a straight-line or geodesic camera path is initially assumed and a repulsive potential is defined around the camera border to push the target back in the field of view. The required modification to the camera path is found by computing the image Jacobian of the violating image point, and IBVS is used to track the resulting interpolated image trajectory. In PBVS methods [7][9], the image trajectories are not controlled explicitly so that it may be possible for the target to violate image boundary constraints over large camera motions. To address the lack of coordination between camera rotation and translation, the *virtual* position of the target frame is adjusted in [3], while allowing both translation and rotation errors to decrease exponentially as in PBVS. Finally, in 2½D visual servoing [4][5], a diagonal gains matrix is used online to attenuate camera rotational velocities when the target is detected to be near the periphery of the camera, allowing in-plane translational control to bring the target

Manuscript received September 14, 2001. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada.

Ambrose Chan is with the Mechanical Engineering Department, University of British Columbia, Vancouver, CA (phone: 604-822-3147; e-mail: achan82@ mech.ubc.ca)

Elizabeth A. Croft is with the Mechanical Engineering Department, University of British Columbia, Vancouver, CA (phone: 604-822-6614; e-mail: ecroft@mech.ubc.ca)

James J. Little is with the Computer Science Department, University of British Columbia, Vancouver, CA (phone: 604-822-4830; e-mail: little@ cs.ubc.ca)

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

back towards the center of the image. It is critical for these eye-in-hand visual servoing systems to keep the target in view in order to prevent unstable control and servo failure.

All of the above methods for handling field of view constraints require the robot controller to accept input in the form of Cartesian velocities or Cartesian positions at a high rate. Unfortunately, the external communication interfaces of typical industrial controllers do not have the level of flexibility necessary to accommodate velocity-based control or high-rate position control. The communication barrier between the vision software and the robot controller, often limits the use of visual servoing in industrial applications. The majority of industrial robots still rely on look-then-move techniques for vision-based end-effector pose control, since their controllers typically only allow for point-to-point commands via sparse waypoints and endpoints. The precise robot trajectories between waypoints are determined by the robot controller through standard interpolation schemes.

For many of these industrial applications, it is still important to keep the target object in the field of view during the interpolated trajectory, in order to maximize the available visual information for tracking purposes, for object pose calculations and updates, for appearance-based model building, and for inspection and fault monitoring. This paper addresses the problem of ensuring that the target stays within the field of view of the camera throughout the entire interpolated robot trajectory by specifying a sparse set of waypoints that guide the robot towards its goal.

III. Pose Calculations And Modeling

Let $\boldsymbol{F_o}$ be the canonical frame attached to the target object. The eye-in-hand robot starts at an initial configuration $\boldsymbol{q}_{initial}$ and provides an initial view of the target object with a pose, represented by $\boldsymbol{F_o}$, that may vary from instance to instance. In this work, however, the target object remains stationary with respect to the robot base frame $\boldsymbol{F_r}$ during all robot motion. The goal is to position the camera $\boldsymbol{F_c}$ at a desired pose with respect to the target object $\boldsymbol{F_o}$. This desired pose is specified via an offline teach-by-showing method. Namely, the controller is provided with a single reference image of the target object obtained at the desired camera pose. The pose of the object in the camera frame in the initial and desired images, respectively $({}^c\boldsymbol{H_o})_{initial}$ and $({}^c\boldsymbol{H_o})_{desired}$, are calculated using a calibrated camera and a 3D model of the target [8]. There exists a fixed geometric relationship between the end-effector frame $\boldsymbol{F_e}$ and the camera frame $\boldsymbol{F_c}$, described by the homogeneous transformation ${}^e\boldsymbol{H_c}$.. Therefore, the target object is within the field of view of the camera at the start and at the end of the robot motion.

The pose of the object with respect to the robot frame ${}^r\boldsymbol{H_o}$ can be computed by the following:

$$ {}^r\mathbf{H}_o = \left({}^r\mathbf{H}_e(\mathbf{q})\right)_{initial} {}^e\mathbf{H}_c \left({}^c\mathbf{H}_o\right)_{initial} \quad (1)$$

$$ {}^r\mathbf{H}_o = \left({}^r\mathbf{H}_e(\mathbf{q})\right)_{desired} {}^e\mathbf{H}_c \left({}^c\mathbf{H}_o\right)_{desired} \quad (2)$$

where ${}^r\boldsymbol{H_e}$ is the homogeneous transformation mapping the robot base frame $\boldsymbol{F_r}$ to the end-effector frame $\boldsymbol{F_e}$. This relationship is a function of the well-known forward kinematics of the industrial robot. The initial transformation is found through the measured joint angles,

$$ \left({}^r\mathbf{H}_e\right)_{initial} = f_{robot}\left(\mathbf{q}_{inital}\right) \cdot \quad (3)$$

It is necessary to determine the set of joint configurations $\boldsymbol{Q_{desired}}$ which correctly position the camera relative to the target object. Typical industrial robots have closed-form analytic solutions to the inverse kinematics of its end-effector:

$$ \begin{aligned} \mathbf{Q}_{desired} &= \left\{\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_p\right\}_{desired} \\ &= f_{robot}^{-1}\left({}^r\mathbf{H}_e\right)_{desired} \end{aligned} \quad (4)$$

where $({}^r\boldsymbol{H_e})_{desired}$ is computed through the following expression:

$$ \left({}^r\mathbf{H}_e\right)_{desired} = \left({}^r\mathbf{H}_e\right)_{initial} {}^e\mathbf{H}_c \left({}^c\mathbf{H}_o\right)_{initial} \left({}^c\mathbf{H}_o\right)_{desired}^{-1} \left({}^e\mathbf{H}_c\right)^{-1} \quad (5)$$

A diagram of the relationship between coordinate frames is shown in Fig. 1. Note that, for robots with multiple degrees of freedom in revolute joints, there may be multiple inverse kinematic solutions that satisfy the desired relative camera pose. The resulting interpolated camera trajectories are different for each of these joint configurations.

IV. Inverse Kinematics with Visual constraints

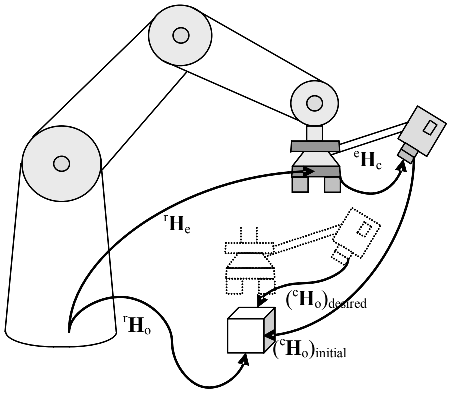
A. Description of Approach

It is well known that, due to the robot's mechanical joint limits, not all mathematical solutions of the inverse kinematic equations will correspond to physically realizable configurations. Once a solution to the mathematical equations is identified, it must be further checked to determine whether it satisfies all constraints on the ranges of possible joint motions. An extension of this requirement, for eye-in-hand robots considered herein, is to check whether the interpolated trajectory corresponding to each inverse kinematic solution satisfies the visibility constraints of the camera for a given target object.

B. Visibility Model

Continuous visibility constraints require that the target object remain within the field of view of the camera CCD array throughout the entire robot trajectory. Let the target object consist of a number of visual features. The image coordinates $(\boldsymbol{u},\boldsymbol{v})$ of the k visual features are functions of the joint coordinates of the robot, where:

$$ \mathbf{u} = \left[u_1(\mathbf{q}) \quad \cdots \quad u_k(\mathbf{q})\right]^T, \ \mathbf{v} = \left[v_1(\mathbf{q}) \quad \cdots \quad v_k(\mathbf{q})\right]^T \cdot \quad (6)$$



$^{e}\mathbf{H}_{c}$
$^{r}\mathbf{H}_{e}$
$(^{c}\mathbf{H}_{o})_{desired}$
$^{r}\mathbf{H}_{o}$
$(^{c}\mathbf{H}_{o})_{initial}$

Fig. 1. Geometric relations between various coordinate frames. The target object remains stationary during the robot's motion.

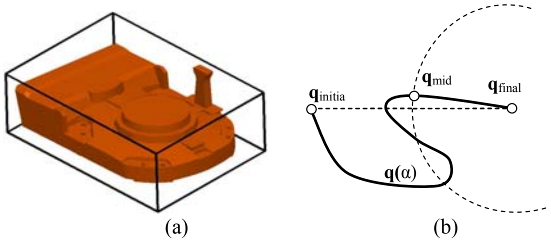


Fig. 2. (a) An oriented bounding box surrounding all the visual features on a typical industrial object. The union of several smaller bounding boxes can also be used to refine the target visibility tests. (b) Waypoint selection based on backwards search from the final configuration along the IBVS path until the half-error condition is met.

The problem of solving for a suitable inverse kinematic solution can be formulated as follows, with α as the path variable:

$$\begin{aligned} &\underset{\mathbf{q}_{desired}\in\mathbf{Q}_{desired}}{\arg\min} \qquad \left\|\mathbf{q}_{desired}-\mathbf{q}_{initial}\right\| \\ s.t. \quad & u_{\min} \le \mathbf{u}\left(\mathbf{q}\left(\alpha\right)\right) \le u_{\max}, \forall\alpha \\ & v_{\min} \le \mathbf{v}\left(\mathbf{q}\left(\alpha\right)\right) \le v_{\max}, \forall\alpha \end{aligned} \quad . \tag{7}$$

where u_{min}, u_{max}, v_{min}, v_{max} are specified by the size of the camera CCD array. Rather than using the array extrema, a thin forbidden region is created near the image border to increase robustness against modeling errors.

If the trajectory of *at least* one feasible inverse kinematic solution satisfies the field of view constraints, then there is no need to specify additional waypoints to guide the robot. If *more than* one inverse kinematics solution satisfies the field of view constraints, then the one with the minimum norm is chosen to minimize travel distance in joint space. For given upper bounds on joint velocities (typically set by robot manufacturers), this solution minimizes the required travel time for time-sensitive applications.

C. Evaluation Methods for Target Visibility

Using the forward kinematics of the robot and a model of the camera, feature points of the target can be projected onto the camera plane as the eye-in-hand camera traverses its 3-D trajectory in simulation. To speed up computations and to accommodate various parametric edge-based or region-based object models, the states of the specific visual features are not simulated. Rather, an oriented bounding box is defined with respect to the target object frame $\boldsymbol{F_o}$ such that it encloses the target object and all its relevant visual features, as shown in Fig. 2 (a). For more complex objects, a union of bounding boxes can be used, assuming that even with self-occlusion, some complete subsets of visual features can be observed from all points of view around the object. The visual and 3D location of eight *virtual* feature points defining the vertices of the bounding box are tracked to bound the visual location of the target object. That is, all *real* features are guaranteed to remain within the field of view of the camera throughout its trajectory if all eight *virtual* feature points are continuously visible.

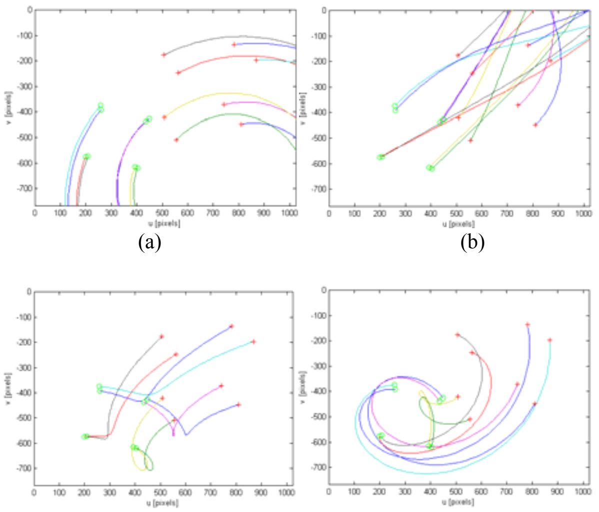
There exists a set of typical interpolation methods that robot manufacturers use for trajectory generation between specified waypoints. These methods include: quintic polynomials, hermite splines, linear-segment with parabolic blend (LSPB), Cartesian-space linear interpolation, simultaneous joint-space linear interpolation, sequential joint-space linear interpolation (i.e. joint-by-joint movement), etc. [10][11][12]. With knowledge of the interpolation scheme, it is possible to simulate both the camera 3D trajectory and the visual trajectory of the target object at the frame rate of the camera in order to determine whether the target remains in the field of view during the entire robot motion. By using a simplified target model, this test can be easily completed online, just prior to the selection of the inverse kinematic solution and the execution of the task.

If no feasible trajectory exists among the inverse kinematic solutions, then additional waypoints must be specified to guide the robot towards its goal while maintaining visibility, as described in Section V. The insertion of an additional visible and feasible waypoint divides the problem into two sub-problems of the same form but of lesser difficulty, since the robot's interpolation distance is reduced.

D. Experimental Results

A vision-guided positioning task is simulated using a CRS-A465 robot and Sony XC-HR70 camera. The target object is enclosed by a 20cm x 20cm x 20cm virtual bounding box made up of eight features, which need to remain inside the field of view during the robot motion. The camera starts from an initial overhead view, and is required to approach the target object while performing significant in-plane and out-of-plane rotations. Of the eight kinematic solutions available for the CRS-A465 to achieve the desired camera pose, four of them violated the mechanical joint limits of the robot. The image trajectories corresponding to the remaining four solutions are shown in Fig. 3. Between waypoints, the robot trajectories are generated using joint space linear interpolation. Note that only two of the four solutions give satisfactory visual trajectories. The one with the minimum travel distance in joint space, shown in Fig. 3 (c) is chosen as the preferred trajectory for executing the positioning task.

In industrial robotics, a common way to specify waypoints is to physically lead the robot through the desired motion with a teach pendant and record the joint configurations. When a robot is taught to observe a target object at two different joint configurations, an aspect that is often overlooked by the engineer is that multiple solutions exist. As demonstrated above, often an alternative joint configuration gives the exact same camera pose, but also provides increased visibility along the trajectory. In some cases, it is possible to obtain continuous target visibility just by considering a different inverse kinematic solution.



(a) (b)

(c) (d)

Fig. 3. Image trajectories of the target's feature points, corresponding to each of the four physically valid inverse kinematic solutions for the desired camera pose. The trajectory between the start and the end joint configurations are generated using linear interpolation in joint-space.

V. Waypoint Generation

A. Theoretical Motivation

When the trajectories to all inverse kinematic solutions of the robot violate the continuous target visibility constraints, it is necessary to insert a number of intermediate waypoints to satisfy these constraints while still moving the robot towards its goal. In terms of meeting the field of view constraints, the ideal image trajectory for a given visual feature is one that travels in a straight-line from its location in the initial image to its location in the desired image. If such trajectory could be realized, then the feature is guaranteed to remain inside the camera's convex field of view as long as it is visible in the initial image and in the desired image.

B. Simulated Visual Servoing

IBVS is a method that is known to generate near straight-line image trajectories. As described before, many industrial controllers are not capable of accepting the continuous velocity or high-rate position input that is required by IBVS. However, its open-loop form can still be used to guide the selection of a set of sparse waypoints.

As long as the target is within the field of view, it is not necessary to confine the image trajectory to follow a straight-line path at all times. Specifically, it is not necessary for the camera to track the IBVS path, which can be quite convoluted in joint space and in Cartesian space. Also, the timing of the IBVS trajectory must be addressed, since it is sub-optimal over large distances. The velocity of the camera decreases exponentially with the image error, generating large limit-exceeding velocities at the start and negligible motions towards the end. For time-sensitive applications, it is desirable to use a timing that is more consistent, such as joint-space interpolation. The key is to use IBVS as a guide to select a sparse set of waypoints, while allowing the robot to travel freely in between using its chosen method of joint interpolation.

We propose to use IBVS in simulation to generate joint space trajectories to guide in our selection of waypoints. IBVS is *virtually* applied to the eight feature points defining the vertices of the oriented bounding box, where the error to be controlled is the difference between their current (*u,v*) and their desired (u^*,v^*) image locations:

$$\mathbf{e} = \left[\left(u_1^* - u_1\right) \quad \left(v_1^* - v_1\right) \quad \cdots \quad \left(u_8^* - u_8\right) \quad \left(v_8^* - v_8\right) \right]^T \quad (8)$$

A proportional control law is used to drive image coordinates exponentially towards their desired locations,

$$\dot{\mathbf{e}} = -\lambda \mathbf{e} \quad (9)$$

where λ is the convergence rate. The exact value of λ is not important, since we are only interested in the path that is generated from IBVS, not the associated timing law. The required joint and camera motion in order to achieve such a trajectory can be found:

$$\dot{\mathbf{q}} = -\lambda \cdot \mathbf{J}^{+}(\mathbf{q}) \cdot \mathbf{L}^{+}(\mathbf{x},\mathbf{y},\mathbf{Z}) \cdot \mathbf{e} \quad (10)$$

where ***J(q)*** is the robot Jacobian corresponding to the eye-in-hand configuration, and ***L*** is the image Jacobian of the eight feature points. ***L*** is a stacked matrix of the following form [14],

$$\mathbf{L} = \left[\mathbf{L}_1^T \quad \cdots \quad \mathbf{L}_9^T \right]^T \quad (11)$$

where

$$\mathbf{L}_i\left(x_i,y_i,Z_i\right)=\begin{bmatrix} \frac{-1}{Z_i} & 0 & \frac{-x_i}{Z_i} & x_iy_i & -\left(1+y_i^2\right) & y_i \\ 0 & \frac{-1}{Z_i} & \frac{-y_i}{Z_i} & 1+y_i^2 & -x_iy_i & -x_i \end{bmatrix} \quad (12)$$

The distance Z of a feature point to the image plane is obtained from the model of the object. The normalized coordinates (x,y) are calculated from (u,v) using the camera calibration relationship.

C. *Image Local Minima and Robot Singularities*

By comparing the *virtual* camera pose with the final camera pose (and *virtual* image with the actual observed image), it is possible to determine whether the IBVS solution has converged after a set number of iterations. One can safely determine whether a local minimum is encountered, or whether the robot has exceeded its joint limits during IBVS servoing prior to committing the robot to the path. It is found that the *virtual* visual servoing often fails to converge if the trajectory passes near robot singularities, since very high joint velocities are generated which bring the trajectory (both in image space and in joint space) away from the true solution in the presence of numerical errors. This results in an invalid camera path, bringing the target out of the field of view and leading to servo failure. A damped least-squares inverse kinematics solution [13] is implemented to deal with robot singularities.

D. *Sparse Waypoint Selection*

The probability of the target object leaving the field of view of the camera for any joint interpolation scheme increases with the magnitude of the joint motion. If an exit is detected in the model, a natural way is to impose a waypoint on the IBVS path at the 'midpoint' between the two specified configurations.

Let the IBVS path be parameterized by $\boldsymbol{q} = f(\alpha)$ with $\alpha = 0$ at the start configuration and $\alpha = 1$ at the final configuration. The problem of waypoint selection can be formulated as:

$$\begin{aligned} &\underset{\mathbf{q}}{\arg\max} && \alpha = f^{-1}(\mathbf{q}) \\ &s.t && \left\|f(\alpha)-f(0)\right\| = \tfrac{1}{2}\left\|f(1)-f(0)\right\| \end{aligned} \quad (13)$$

A backwards search is initiated from the end configuration, selecting the first point along the IBVS path where the joint-space error is half of that between the start and end configuration. If the joint-space path is convoluted, the waypoint is asymmetrically biased towards to the end configuration rather than the start configuration. A conceptual illustration of this is shown in Fig. 2 (b). Due to the linear approximation nature of the proportional control law in IBVS, points near the end configuration are often more reliable and more likely to lie along a 'critical' path than points at the start of the servoing path. Other methods of waypoint selection, such as choosing a point that minimizes the norm of the Jacobian, to reduce camera pose errors during switchover between the two segments, can also be used as a selection criterion. This method is applied recursively between waypoints until the visibility criterion is met.

E. *Experimental Results*

The same vision-guided positioning task is simulated with the CRS-A465 robot and the Sony XC-HR70 camera. However, the focal length of the lens is increased and the field of view is reduced by approximately 33%. When using linear interpolation in joint-space for trajectory generation, all of the inverse kinematic solutions fail to maintain continuous visibility on the target object (Fig. 4). Virtual IBVS was used to generate candidate waypoints, and its corresponding image trajectory is shown in Fig. 5. This particular IBVS trajectory passes near a robot singularity, although it was determined to converge using a damped least-squares inverse near singularity regions. Using the proposed method, it is found that it usually takes a small number of waypoints (1 or 2) to bring the target object back in view. The image trajectory generated from the sparse waypoint selection method with joint-space interpolation is shown in Fig. 6.

F. *Further Discussion*

In the presence of uncertainties in the camera parameters, the robustness of this method can be extended by introducing a rectangular forbidden region for feature points observed around the image border. The reduced field of view creates additional, but not necessarily critical, waypoints to sub-divide the interpolation problem, and forces the camera to take smaller steps. The advantage of the method of sparse waypoint selection over simulated IBVS is that the path position data can be reduced by several orders of magnitude (from over a thousand points to three points in the example shown). This is particularly important for engineering applications where the communication rate to the robot controller is limited and real-time visual servo is not possible.

VI. Conclusion

This paper presented several methods of trajectory generation for an eye-in-hand robot with camera field of view constraints. Where real-time communication between the vision system and the robot controller is not possible, a simplified target and camera-robot model is used to predict the visibility of the target object online. The selection of sparse waypoints from a virtually simulated IBVS trajectory ensures that the object stays within the field of view, while significantly reducing the joint position data that must be communicated to the robot controller.

An obvious trade-off exists between the model-based approaches presented here and 'reactive' methods, such as visual servoing. When the rate of positional update is low, it is essential to have accurate models for trajectory prediction over large distances. In our collaboration with our industrial partners, it has been demonstrated that it is possible to achieve sufficiently accurate object models built from visual data, as well as good camera and robot calibration, in typically manufacturing environments. The communication barrier between the vision software and the robot controller that exists in manufacturing are also present in space robotics and tele-operations, where the user provides sparse (perhaps delayed) vision-based position input, but still expects continuous visual information from the target object.

For future work, the smoothness of the trajectory at each waypoint for a given interpolation type should be evaluated to ensure that camera velocities do not change abruptly between interpolated segments. In highly cluttered workspaces, occlusions by other objects and the robot arm itself should be modeled as well.

REFERENCES

[1] Y. Mezouar and F. Chaumette, "Avoiding self-occlusions and preserving visibility by path planning in the image," in *Robotics and Autonomous Systems*, vol. 41, pp. 77–87, 2002.

[2] Y. Mezouar and F. Chaumette, "Path Planning for Robust Image-Based Control," in *IEEE Transactions on Robotics and Automation*, vol. 18, no. 4, pp. 534–549, Aug. 2002.

[3] B. Thuilot, P. Martinet, L. Cordesses and J. Gallice, "Position based visual servoing: keeping the object in the field of vision," in *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, pp. 1624–1629, May 2002.

[4] E. Malis, F. Chaumette and S. Boudet, "2-1/2-D Visual Servoing," in *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, Apr. 1999.

[5] E. Malis and F. Chaumette, "Theoretical Improvements in the Stability Analysis of a New Class of Model-Free Visual Servoing Methods," in *IEEE Transactions on Robotics and Automation*, vol. 18, no. 2, pp. 176-186, Apr. 2002.

[6] L. Deng, F. Janabi-Sharifi and W. J. Wilson, " Hybrid Motion Control and Planning Strategies for Visual Servoing," in *IEEE Transactions on Industrial Electronics*, vol. 52, no. 4, pp. 1024–1040, Aug. 2005.

[7] W.J. Wilson, C. C. Williams Hulls and G. S. Bells, "Relative End-Effector Control Using Cartesian Position Based Visual Servoing," in *IEEE Transactions on Robotics and Automation*, vol. 12, no. 5, pp.684–696, Oct. 1996.

[8] D. F. Dementhon and L. S. Davis, "Model-Based Object Pose in 25 Lines of Code," in *International Journal of Computer Vision*, vol. 15, pp. 123–141, 1995.

[9] V. Lippiello, B. Siciliano and L. Villani, "Position-Based Visual Servoing in Industrial Multirobot Cells Using a Hybrid Camera Configuration," in *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 73–86, Feb. 2007.

[10] L. Sciavicco and B. Siciliano, *Modeling and Control of Robot Manipulators, 2nd Ed.*, New York, NY: McGraw Hill, Springer-Verlag, 2000.

[11] J. J. Craig, *Introduction to Robotics: Mechanics and Controls, 2nd Ed.*, Addison Wesley, 1989.

[12] M. W. Spong, S. Hutchinson and M. Vidyasagar, *Robot Modeling and Control*, John Wiley and Sons, 2006.

[13] S. Chiaverini and B. Siciliano, "Review of the Damped Least-Squares Inverse Kinematics with Experiments on an Industrial Robot Manipulator," in *IEEE Transactions on Control Systems Technology*, vol. 2, no. 2, pp.123–134, 1994.

[14] S. Hutchinson, G. D. Hager and P. I. Corke, "A Tutorial on Visual Servo Control," in IEEE Transactions on Robotics and Automation, vol. 12, no. 5, pp. 651-670, Oct. 1996.

[15] S. Abrams, P. K. Allen and K. A. Tarabanis, "Dynamic Sensor Planning," presented at the International Conference on Intelligent Autonomous Systems, Pittsburgh, PA, Feb. 1993, pp. 203-215.

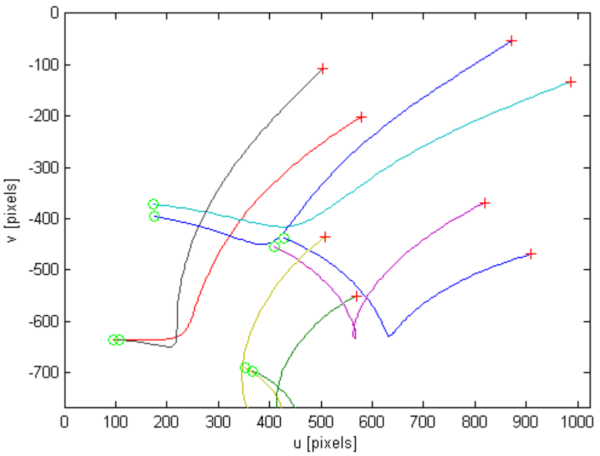


Fig. 4. Image trajectory of the target's feature points generated using the joint-space trajectory from section IV. The same trajectory fails to maintain continuous visibility on all feature points when focal length is increase and the field of view in reduced (as shown).

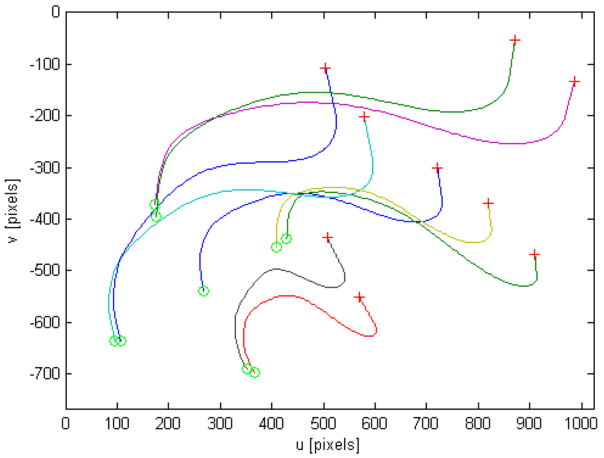


Fig. 5. Image trajectory of the target's feature points where the robot trajectory is generated using IBVS.

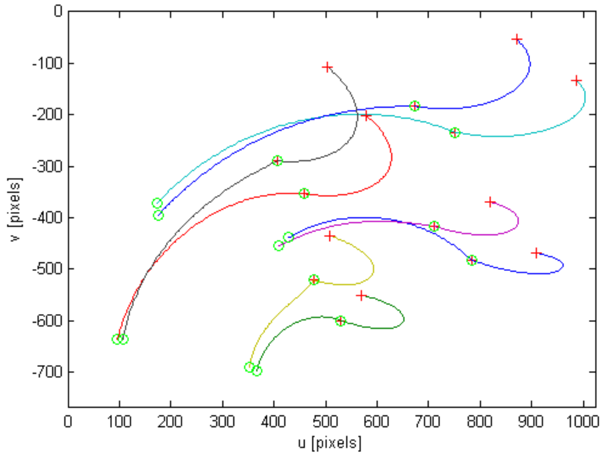


Fig. 6. Image trajectory of the target's feature points where the robot trajectory is generated using the proposed method: a sparse set of visible waypoints guided by IBVS. The trajectory between waypoints is produced using linear interpolation in joint-space.