Proceedings of the 2001 IEEE
International Conference on Robotics & Automation
Seoul, Korea - May 21-26, 2001

Visual Servoing: Path Interpolation by Homography Decomposition

Justin A. Borgstadt and Nicola J. Ferrier*
Department of Mechanical Engineering
University of Wisconsin-Madison
borgstad@robios6.me.wisc.edu ferrier@robios6.me.wisc.edu

Abstract

In order to successfully perform visual servoing of robot manipulators the control algorithm must account for the physical limitations of the manipulator. These constraints define the robot workspace boundary. Any visual servoing control algorithm must avoid this boundary if stability of motion is to be ensured. In this paper, a method is developed by which a desired object path can be interpolated between two arbitrary object poses based on image feature extraction. This path is defined to be a continuous change in object pose between the initial and final desired poses which avoids the workspace boundary. The development of this method involves the parameterization of the 2-D displacement transformation or homography. By decomposing this homography a set of object path poses can be interpolated in either the image plane or reconstructed in the 3-D workspace. Implementation of a visual servoing procedure confirms the validity of the interpolated path with respect to workspace boundary avoidance.

Keywords: *visual servo, path, homography*

1 Introduction

Visual servoing is the process by which machine vision is used in feedback control to manipulate the robot end-effector to a desired pose. Systems of a "look-then-move" architecture can be grouped into two general control type categories: position-based control and image-based control [9]. Position-based control combines image feature information, the task space manipulator jacobian, and a camera calibration to estimate the 3-D pose of the end-effector. Feedback control is computed to reduce errors between the current pose of the end-effector and the desired target pose. While clearly separating the vision and robot control issues position-based control is calibration sensitive and does not guarantee image features will stay in the field of view [2]. In image-based control the error function is computed in the image directly. Camera calibration errors are eliminated but the image jacobian used to define the control law is non-linear and requires depth estimation [9]. Additionally, the task space jacobian is still needed to accurately manipulate the end-effector.

Any type of motion control is ultimately determined by the available workspace of the robot. The robot model not only describes the physical 3-D link parameters but also describes the joint types, joint placement, and physical limitations of the actuators. The workspace boundary is broadly defined as the limit to which the end-effector can be manipulated with the robot fully stretched out [7]. For all joint positions corresponding to the workspace boundary the task space jacobian is singular. These configurations can be termed workspace boundary singularities [7]. Additionally, a loci of robot configurations lies within the work-space boundary where again the task space jacobian is singular and are defined as interior workspace singularities [7]. Any motion command attempting to move the end-effector must account for these singularity configurations to ensure stability and convergent motion.

1.1 Previous Work

Image-based visual servoing methods have been proposed which rely on estimating the image jacobian at each iteration of the control law from measured image features and their respective depth estimations [2, 5, 9]. This type of linear control commands the path of each image feature to be a straight line from the current image pose to the desired image pose. For large displacements particularly involving large rotations this approach can command undesirable motion of the end-effector beyond the workspace boundary.

Several methods have been proposed to avoid workspace boundary singularities. One image-based approach partitions the image jacobian into compo-

*Person to whom correspondence should be addressed. This research supported in part by NSF IRI-9703352.

nents containing purely image feature parameters and components containing depth parameters [6]. This method is designed to effectively control large motions of the end-effector about the optical axis. Other hybrid methods combine aspects of both position based and image based control [4, 10, 15]. These methods decompose the 2-D displacement transformation or homography computed from displaced co-planar image features into rotational and translational components. These components are then used to construct an error function of translations and rotations defined in 3-D or a combination of 2-D and 3-D. For these methods the image feature paths are not obvious and there is no guarantee the image features will stay in the camera field of view during the motion. In [12] a method is proposed where the image feature point paths are confined to the image space by implementing a potential function which "repels" image features away from the edges of the image. Another possible solution is presented in [3] where the end-effector trajectory follows a straight line in the task space. Methods have also been presented which avoid any motion of the robot in the vicinity of its joint limits [11] and kinematic singularity positions [13] through functions based on knowledge of the robot jacobian at each pose. In [14] an end-effector representation is used in conjunction with parameterized projective translation and rotation transformations. A projective jacobian is proposed to allow both the error function and the commanded joint space motion of the manipulator to be computed entirely in projective space.

1.2 Goals

In this paper, an End-point Closed Loop (ECL) architecture [9] is implemented where a single fixed camera is used to view a rigid object attached to the end-effector. The main goal is to develop a method by which a "desired viable object path" can be interpolated between two arbitrary image poses, where "desired viable object path" refers to a path along which the object can be manipulated where the change in object pose is continuous and workspace boundary singularities are avoided. A desired path can be uniquely defined by choosing a particular decomposition of the 2-D homography or collineation mapping the projective displacement of the object features between the initial and final image poses. Using a known object model the interpolation of this desired path can then be represented in the task space by a 3-D reconstruction *or* mapped directly to the image space. A visual-servoing procedure is used to validate the path interpolation method as a means to stabilize motion control with respect to workspace boundary singularities.

2 Image Displacements

Interpolating a viable object path for a given object displacement requires knowledge of the initial and final poses as well as how the object is to be displaced. A homography is a mapping from 2-D projective space to 2-D projective space, which is used here to define the 2-D displacement transformation between two object poses in the image. A specific form of the homography is derived and decomposed to interpolate a unique path. This specific form is obtained by directly mapping a 3-D displacement transformation to the image plane using a pinhole camera model.

2.1 Object Representation

Points are chosen here to represent a set of recognizable and distinguishable object features. An object pose can be represented by a matrix consisting of the object feature point vectors as follows:

$$P_k = \begin{bmatrix} M_{0,k} & M_{1,k} & M_{2,k} & \dots & M_{j,k} \end{bmatrix} \quad (1)$$

where the point $M_{j,k}$ is the *jth* object feature point in the *kth* object pose described in the task space. The projection of this object pose into the 2-D image pixel frame $\{p\}$ can be found using a generalized projection transformation as follows

$$\tilde{p}_k = \tilde{H}\tilde{P}\ {}^c\mathrm{T}_h P_k \quad (2)$$

Here $\tilde{P}$ represents the 3x4 normalized perspective projection matrix and $\tilde{H}$ is the intrinsic camera parameter matrix [8]. The 4x4 matrix ${}^c\mathrm{T}_h$ represents the transformation from a designated home frame $\{h\}$ to the camera frame $\{c\}$. This frame can be chosen to have the following form

$${}^c\mathrm{T}_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & {}^cZ_h \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

where ${}^cZ_h$ is the optical axis distance between $\{c\}$ and $\{h\}$. The pose P_0, referred to as the *virtual home position* of the object, is used to denote the pose at which the relationship between all object features is known *a priori* in $\{h\}$.

The true image pose of the object in the pixel frame p_k is given by

$$p_k = \begin{bmatrix} m_{0,k} & m_{1,k} & \dots & m_{j,k} \end{bmatrix} \quad (4)$$

where $m_{j,k}$ is the normalized projection of object feature point $M_{j,k}$.

Eq. 2 can be used to compute the virtual home position pose p_0 providing a reference position from which to describe all image displacements. In this paper P_0 is chosen to be object centered in $\{h\}$.

2.2 Homographies: Image Mapping

The displacement of an object from P_0 to an arbitrary pose P_k can be represented by

$$P_k = \mathrm{T}_{k0} P_0 \tag{5}$$

where T_{k0} is a 4x4 homogeneous matrix of the form

$$\mathrm{T}_{k0} = \begin{bmatrix} \mathrm{R}_{k0} & \tau_{k0} \\ 0 & 1 \end{bmatrix} \tag{6}$$

The 3x3 matrix R_{k0} describes the change in orientation of the object about the 3 coordinate axes of $\{h\}$ where $\mathrm{R}_{k0} \in SO(3)$. This matrix is chosen to be parameterized by three Euler angles α_k, β_k, and γ_k defined to be positive counterclockwise about the positive ${}^hX$, ${}^hY$, and ${}^hZ$ axes of $\{h\}$ respectively and is given by $\mathrm{R}_{k0} =$

$$\begin{bmatrix} c\beta_k\, c\gamma_k & -c\beta_k\, s\gamma_k & s\gamma_k \\ c\alpha_k\, s\gamma_k + s\alpha_k\, s\beta_k\, c\gamma_k & c\alpha_k\, c\gamma_k - s\alpha_k\, s\beta_k\, s\gamma_k & -s\alpha_k\, c\beta_k \\ s\alpha_k\, s\gamma_k - c\alpha_k\, s\beta_k\, c\gamma_k & s\alpha_k\, c\gamma_k + c\alpha_k\, s\beta_k\, s\gamma_k & c\alpha_k\, c\beta_k \end{bmatrix} \tag{7}$$

Taken in the order given these angles uniquely describe an arbitrary change in orientation of the object with respect to P_0 [16]. The 3x1 column vector τ_{k0} represents the translational changes in displacement in the ${}^hX$, ${}^hY$, and ${}^hZ$ directions.

A specific transformation describing the displacement from p_0 to an arbitrary pose p_k in the image can be derived by using Eq. 2 to solve for P_k and P_0. Substitution of these poses back into Eq. 5 yields the following

$$\tilde{p}_k = \tilde{H} \mathrm{H}_{k0} \tilde{H}^{-1} \tilde{p}_0 \tag{8}$$

The 3x3 matrix H_{k0} is defined as the homography in the image centered frame $\{i\}$ and is given by

$$\mathrm{H}_{k0} = \tilde{P}\, {}^c\mathrm{T}_h \mathrm{T}_{k0}\, {}^c\mathrm{T}_h^{-1} \tilde{P}^{-1} \tag{9}$$

Since $\tilde{P}$ and ${}^c\mathrm{T}_h$ are constants the homography becomes dependent only on the 6 independent parameters of T_{k0}. Eq. 9 is merely symbolic since the perspective projection matrix $\tilde{P}$ is a 3x4 matrix and has no unique inverse. Therefore, a different approach must be used in order to parameterize H_{k0}.

2.3 Homography Parameterization

The transformation between two object poses in the image plane with *unit focal length* can be now be represented as

$$\tilde{p}_k = \mathrm{H}_{k0} \tilde{p}_0 \tag{10}$$

where $\tilde{p}_k$ and $\tilde{p}_0$ are the $\{i\}$ coordinates of the object. Using Eq. 5 the *jth* object feature point can be written for two object poses P_0 and P_k explicitly as

$$M_{j,0} = \begin{bmatrix} X_{j,0} \\ Y_{j,0} \\ Z_{j,0} \\ 1 \end{bmatrix} \qquad M_{j,k} = \begin{bmatrix} X_{j,k} \\ Y_{j,k} \\ Z_{j,k} \\ 1 \end{bmatrix} \tag{11}$$

By combining Eqs. 5, and 10 the coordinates of $M_{j,0}$ and $M_{j,k}$ in $\{i\}$ can be found as follows

$$\tilde{m}_{j,0} = \tilde{P}\, M_{j,0} \quad \tilde{m}_{j,k} = \tilde{P}\, \mathrm{T}_{k0}\, M_{j,0} \tag{12}$$

By substituting Eq. 12 into Eq. 10 three equations can be written explicitly by matrix multiplication. Setting the coefficients of the variables $X_{j,0}$ and $Y_{j,0}$ equal to each other a *particular* solution for H_{k0} can be found as follows

$$(\mathrm{H}_{k0})_j = \begin{bmatrix} R_{k0}(1,1) & R_{k0}(1,2) & \frac{R_{k0}(1,3)\, Z_{j,0} + X_{k0}}{Z_{j,0} + {}^cZ_h} \\ R_{k0}(2,1) & R_{k0}(2,2) & \frac{R_{k0}(2,3)\, Z_{j,0} + Y_{k0}}{Z_{j,0} + {}^cZ_h} \\ R_{k0}(3,1) & R_{k0}(3,2) & \frac{R_{k0}(3,3)\, Z_{j,0} + Z_{k0} + {}^cZ_h}{Z_{j,0} + {}^cZ_h} \end{bmatrix} \tag{13}$$

The three elements of the last column of $(\mathrm{H}_{k0})_j$ include the term $Z_{j,0}$ which is the ${}^hZ$-coordinate of the *jth* object feature point. As indicated by the subscript j, the general form of the homography is object feature point dependent. Therefore, if a 3-D object is described using object feature points which are chosen such that they are not coplanar in $\{h\}$, the homography H_{k0} between two object image poses becomes the set of image feature point homographies as follows

$$\mathrm{H}_{k0} = \{\ (\mathrm{H}_{k0})_0,\ (\mathrm{H}_{k0})_1,\ \ldots\ (\mathrm{H}_{k0})_n\ \} \tag{14}$$

Here n is the total number of object features which are measurable between two object image poses.

If the object can be represented by a set of coplanar object feature points and the plane described by these points is $\perp$ ${}^hZ$ such that $Z_{j,0} = 0$ for $j = 1, 2, .., n$ then the general form of H_{k0} is

$$\mathrm{H}_{k0} = \frac{1}{div_{k0}} \begin{bmatrix} div_{k0}\, R_{k0}(1,1) & div_{k0}\, R_{k0}(1,2) & div_{k0}\, x_{k0} \\ div_{k0}\, R_{k0}(2,1) & div_{k0}\, R_{k0}(2,2) & div_{k0}\, y_{k0} \\ div_{k0}\, R_{k0}(3,1) & div_{k0}\, R_{k0}(3,2) & 1 \end{bmatrix} \tag{15}$$

The inverse of element $H_{k0}(3,3)$ defines the divergence, div_{k0}. and . The divergence, a measure of how the object is scaled in the image, is given by

$$div_{k0} = \frac{{}^cZ_h}{{}^cZ_h + Z_{k0}} \tag{16}$$

and is nonzero for ${}^cZ_h > 0$. In practice ${}^cZ_h$ should always be chosen to be meet this condition since ${}^cZ_h \leq 0$ would physically mean p_0 was at the focal point with infinite divergence or inverted behind the focal plane. If $Z_{k0} \leq (-\ {}^cZ_h)$ this would indicate the object had been displaced to a position at or behind the focal plane. Choosing ${}^cZ_h$ to be positive and assuming all displacements of the object are measurable in the image the divergence parameter is never zero. Additionally, the parameters x_{k0} and y_{k0} are image plane translations corresponding to the perspective projections of X_{k0} and Y_{k0}.

3 Path Interpolation

By measuring p_k in the pixel frame and using the pre-computed virtual home pose p_0 a numerical least squares method [1] is used to solve for the elements of the normalized H_{k0} given in Eq. 15. The symbolic form of the homography is then decomposed between any two arbitrary object image poses.

3.1 Homography Decomposition

The numerical solution for the elements of H_{k0} is represented here as

$$H_{k0} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \tag{17}$$

and is used to decompose the parameterized form into the 6 independent parameters: γ_k, α_k, β_k, x_{k0}, y_{k0}, and div_{k0} . Referring to Eqs. 7 and 15 the angle γ_k can be found from

$$\gamma_k = \arctan\left(-\frac{b}{a}\right) \tag{18}$$

where the particular quadrant of γ_k can be determined by the corresponding signs of a and b.

The divergence div_{k0} and image translations x_{k0} and y_{k0} are given as follows:

$$div_{k0} = \left(\frac{a^2+b^2+d^2+e^2+g^2+h^2}{2}\right)^{\frac{1}{2}} \tag{19}$$

$$x_{k0} = \frac{c}{div_{k0}} \qquad y_{k0} = \frac{f}{div_{k0}} \tag{20}$$

The angle α_k is given as

$$\alpha_k = \arccos\left(\left(\frac{div_{k0}^2-(g^2+h^2)}{a^2+b^2}\right)^{\frac{1}{2}}\right) \tag{21}$$

where $0 \leq \alpha_k < 90^o$. A degenerate case exists if $a^2+b^2=0$ or when $\alpha_k > 90^c$ corresponding to self-occluded planar object feature points in the image.

The angle β_k is given by

$$\beta_k = \arccos\left(\left(\frac{a^2+b^2}{div_{k0}^2}\right)^{\frac{1}{2}}\right) \tag{22}$$

which gives $0 \leq \beta_k < 90^o$ where a degenerate case exists if $div_{k0} = 0$. This can be avoided by choosing an appropriate value for ${}^cZ_h$ as previously mentioned.

The range of α_k and β_k can be increased from -90^o to 90^o including negative rotations by a simple optimization using the numerical values in Eq. 17 and the symbolic formulas for the elements of Eq. 13.

3.2 Path Algorithm

By choosing to express the displacement of an object from p_0 to p_k by the particular parameterization of H_{k0} given in Eq. 15 a unique path m can be defined between any two arbitrary poses of the object. An object image pose lying on the path can be defined as $p_{k,m}$. This unique path m can be represented discretely in the following form

$$s_m = \{p_{1,m}, p_{2,m}, ..., p_{k,m}, ..., p_{N,m}\} \tag{23}$$

where s_m is the set of N image poses of the object between and including the end poses $p_{1,m}$ and $p_{N,m}$ for k = 1,2,...,N. The pose $p_{k,m}$ can be found by from Eq. 10 where H_{k0} is computed by interpolating the rotation angles α_k, β_k, and γ_k, the translations x_{k0}, y_{k0}, and the div_{k0} between the values found at k=1 and k=N by decomposition.

The general procedure for finding the discrete path s_m in the image can now be stated as follows:

Assume: p_0 is known *a priori*

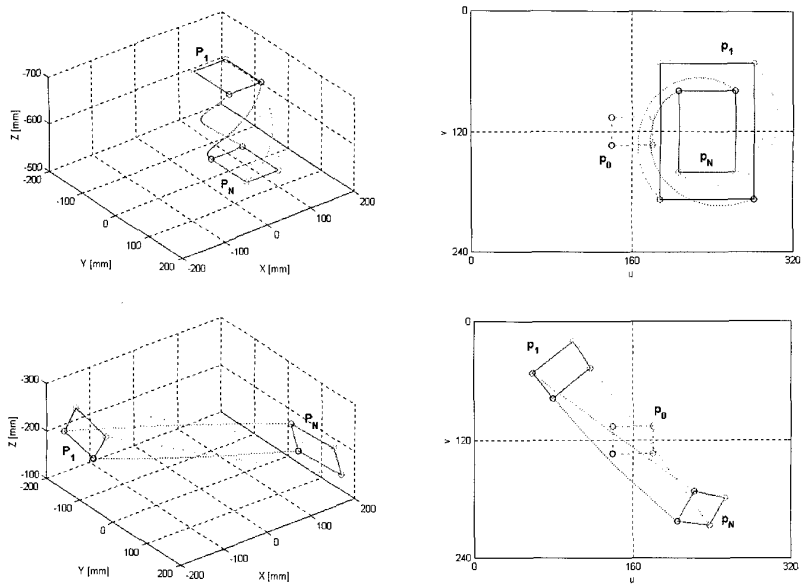
Given: p_1 and p_N for a particular displacement

Procedure: Decomposing the homographies H_{10} and H_{N0} into the respective rotation angles, translations, and divergence from p_0 to p_1 and from p_0 to p_N, the discrete path s_m can be found by choosing a satisfactory value for N and linearly interpolating these rotation angles and translations between p_1 and p_N using Eq. 10.

The particular path s_m found using this method depends entirely on the particular parameterization chosen for H_{k0}. This method for path interpolation depends on having *a priori* knowledge of the object either in the form of a 3-D model or measured points in the image. The path can also be interpolated in the 3-D task space. This can be done by finding X_{k0}, Y_{k0}, and Z_{k0} from x_{k0}, y_{k0}, and div_{k0} respectively by using the perspective projection equations and Eq. 16. The decomposed parameters can be used to reconstruct the 3-D displacement transformation T_{k0} and a similar path interpolation procedure can be performed in $\{h\}$.

4 Experimental Results

Our path interpolation and visual servoing along the interpolated path methods were tested in a lab setting. Results are presented from a simulation to test the validity of the path interpolation method between any two arbitrarily specified object poses. The simulation is then constrained to include a model of the robot manipulator thereby restricting the motion of the object



(a) 3-D Displacements (b) 2-D Image Displacements

Figure 1: Path Simulations

to the available workspace. The simulated and experimental results of one visual servoing trial are presented here. In this paper object feature points in the image were measured using a snake tracker. A rectangular box was chosen as the object where the features are all points on one face of the box (coplanar points to ensure a numerical solution for the homography). The box snake consists of a B-spline curve defined by 12 total control points with 4 knots of multiplicity 2 forming a closed loop. The vision system consists of a CCD camera mounted approximately 950 mm directly over the robot. The robot manipulator is a 6 degree of freedom Robix RCS-6 model where the rectangular box being manipulated is affixed to the end-effector. A primary PC handles all image I/O and system management. The Robix RCS-6 is run from a secondary PC via serial port communication.

4.1 Path Simulation

The path simulation takes as input a set of coplanar object feature points given at home position P_0 of the object in $\{h\}$. Our simulation takes as input 2 sets of screw displacement coordinates (3 displacement rotations and 3 displacement translations each) describing the displacement of the object from P_0 to P_1 and from P_0 to P_N in $\{h\}$. The object pose projections p_0, p_1 and p_N are computed in the image frame using Eq. 2 where ${}^cZ_h = 1000$ mm. The homographies H_{10} and H_{N0} are computed using the numerical solution and the path interpolation method described in Section 3.

Two sets of object displacements and the corresponding interpolated paths are shown in Figure 1. The left-hand column is the input 3-D displacement and 3-D interpolated path. The right hand column is the corresponding 2-D displacement of the object and the 2-D interpolated path. The corners of the box are the image features used to indicate each pose along the path. The first displacement is characterized by a 180^o rotation about the ${}^hZ$-axis and a translation in ${}^hX$ and ${}^hZ$. The second displacement is characterized by 60^o rotation about the ${}^hZ$-axis and a translation in ${}^hX$ and ${}^hY$. The *only* correspondence between the 3-D and 2-D interpolated paths is the projection of P_1 and P_N in $\{h\}$ to p_1 and p_N in the image pixel frame p where the 3-D path and 2-D paths were computed independently. However, the *projection* of the 3-D path into $\{p\}$ matches the 2-D path produced by our method within machine precision. This verifies that each object pose along the interpolated path in the image corresponds to an actual physical object pose in 3-D along the screw axis between P_1 and P_N. Additionally, for the second displacement the object was oriented such that p_k was not coplanar with p_0 for k = 1,2,..,N, demonstrating that the camera can be in any orientation relative to the object as long as

the object features can be measured and P_0 is defined as described in Section 2.

4.2 Visual Servoing

The path simulation was constrained to include a model of the Robix RCS-6. The purpose of the simulation is to find displacements for which the object *can* be manipulated along the desired path by the robot while avoiding interior workspace singularities. An iterative servoing method is implemented here which involves reconstructing the 3-D displacement screw coordinates from the objects current pose to the next desired pose on the interpolated path. The simulation is also used here to determine values for the number of interpolated poses N and for the tolerance within which the object is to be iteratively manipulated to each desired pose which ensure convergence along the entire path.

The visual servoing trial presented is defined by a single rotation about joint 1 of the robot where $\Delta\theta_1 = 45^o$ and $N = 31$. The motion is represented in the left-hand column of Figure 2 by four simulated object image poses (p_1, p_{14}, p_{23}, and p_{31}). The right-hand column of Figure 2 shows four images (p_1, p_{14}, p_{23}, and p_{31}) of the actual visual servoing manipulation of the object from p_1 to p_{31}. Also indicated in both columns is the virtual home image pose p_0 and the desired image path between p_1 and p_{31}.

The joint positions corresponding to the robot configurations where the object pose p_k was manipulated to p_k^* within the desired tolerance for each value of k = 1,2,...,31 are shown in Figure 3.

Comparison of the joint positions for the entire motion between the simulation and the actual experimental results indicates a strong correlation. This verifies the path interpolation method combined with the visual servoing technique used here is a viable solution for this particular object displacement.

Some noise does exist which can be partially attributed to errors in calibration of the camera intrinsic parameters and the robot model. Most of this error seems to be attributable to the snake tracker which limits the minimum tolerance within which the object can be manipulated to each desired path pose. As can be seen in Figures 2 and 3 these sources of error did not play a significant role in the convergence of the object along the path from the initial pose to the desired final pose.

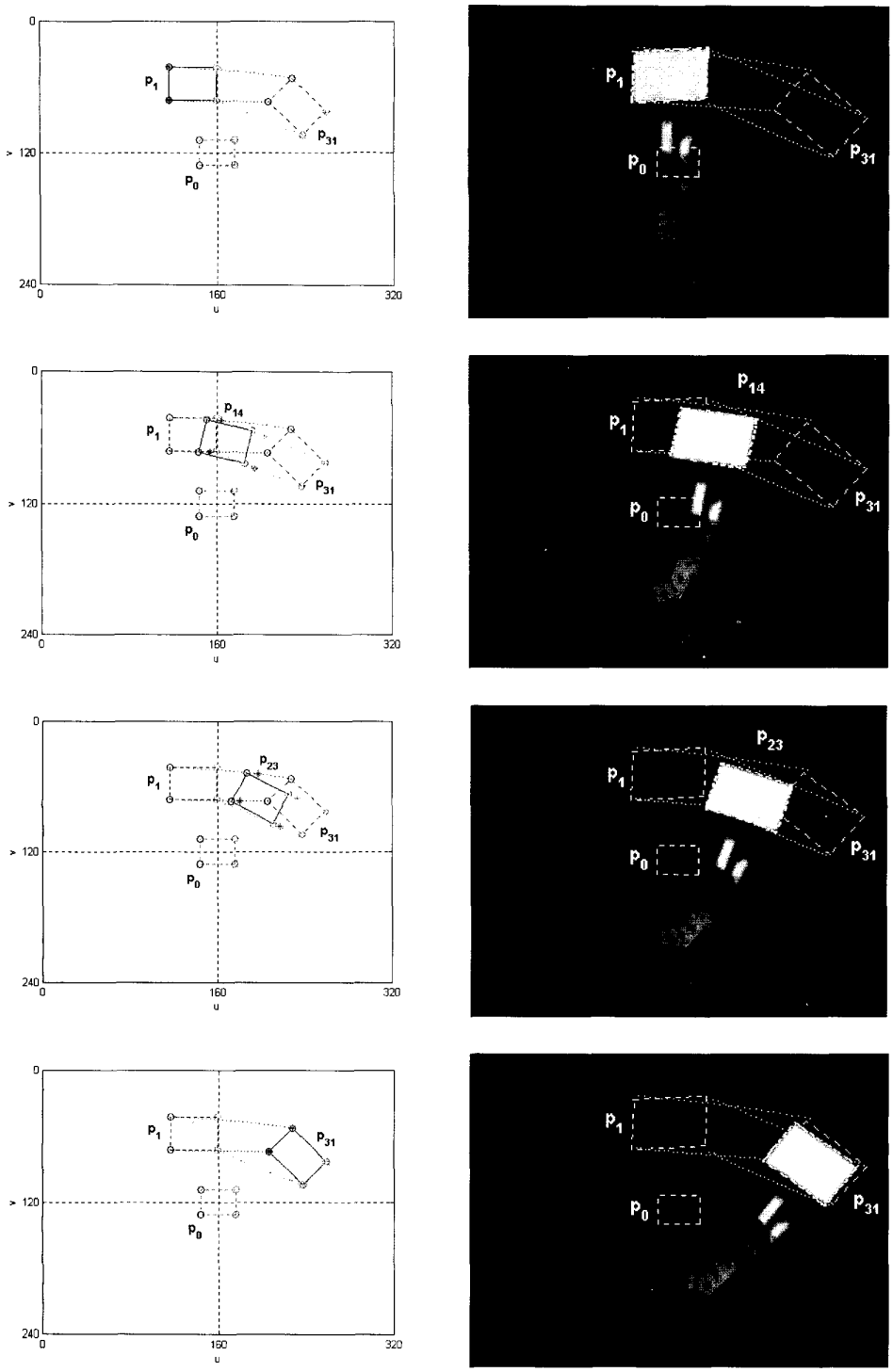
The simulation including the robot model was originally designed to aid in finding object paths which avoided interior workspace singularities as described in section 1. However, object poses have been observed to exist along certain interpolated paths which create control instabilities but do not represent interior workspace singularities. For these poses the object frame jacobian is well-conditioned and invertible but yet the next desired path pose is not attainable. The path interpolation method has been validated as a means to avoid workspace boundary singularities. However, observations suggest path dependent singularity-type conditions can exist in addition to workspace boundary singularities and interior workspace singularities. Although, the path is defined to be continuous and all interpolated path poses lie within the workspace boundary the path may not be represented continuously in the joint space of the robot. In order to successfully manipulate an object along interpolated paths between any two object image poses additional information possibly in the form of a kinematic map of the manipulator will be required in order to avoid *all* workspace singularities and joint space discontinuities and to allow true path planning.

5 Conclusions and Future Work

In this paper, a method for interpolating a viable object path between two arbitrary poses based on image feature extraction was presented. This was done by parameterizing and decomposing the 2-D displacement transformation matrix computed from measured object image features at an initial and desired final pose. Decomposing the parameterized form of the homography into rotational and translational displacement components an object path can be interpolated in either the image plane or in the camera frame relative to a known *a priori* home position. This path was validated through path simulations and experimental visual servoing trials where an object was manipulated along the interpolated path between the initial and final desired object poses.

The goal of developing a method by which a desired viable object path could be used in visual servoing to avoid workspace boundary singularities has been achieved . However, this method is not sufficient to guarantee visual servoing convergence for all types of object displacements. Other issues such as interior workspace singularities, discontinuities in the joint space, and special cases where the image feature paths may lie outside the camera's field of view are currently being addressed. Additionally, the subject of future papers will address the problem of handling 3-D objects and moving away from the 2-D planar case by implementing the general form of the homography derived in section 2.

The development of the path interpolation method in this paper is the first step in the development of a



Simulation Experimental

Figure 2: Visual Servoing Results for $\Delta\theta_1 = 45^o$

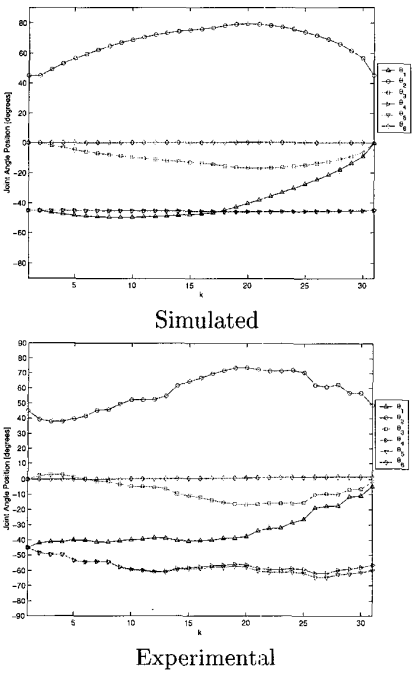


Figure 3: Joint Position Results for $\Delta\theta_1 = 45^o$

true path planning method. A general procedure for visual servoing needs to be developed which incorporates the path interpolation method with knowledge of the reachable workspace and joint space of the robot manipulator in order to more intelligently plan a viable object path.

References

[1] Paul A. Beardsley. *Applications of Projective Geometry to Robot Vision.* PhD thesis, University of Oxford, 1992.

[2] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In *The Confluence of Vision and Control*, LNCS (237), pp. 66–78. Springer Verlag, 1998.

[3] F. Chaumette and E. Malis. 2 1/2 d visual servoing: A possible solution to improve image-based and position-based visual servoings. *Proc. of the IEEE Int'l Conf. on Robotics & Automation*, 2000.

[4] G. Chesi, E. Malis, and R. Cipolla. Automatic segmentation and matching of planar contours for visual servoing. *Proc. of the IEEE Int'l Conf. on Robotics & Automation*, 2000.

[5] P. I. Corke and S. A. Hutchinson. Real-time vision, tracking and control. *Proc. of the IEEE Int'l Conf. on Robotics & Automation*, April 2000.

[6] P. I. Corke and S. A. Hutchinson. Recent results in visual servo control. In *Workshop on Integrating Sensors with Mobility and Manipulation*, April 2000.

[7] J. J. Craig. *Introduction to Robotics: Mechanics and Control.* Addison-Wesley Publishing Company, 2nd edition, 1989.

[8] O. Faugeras. *Three Dimensional Computer Vision.* MIT Press, Cambridge, MA, 1993.

[9] S. Hutchinson, G. Hager, and P. Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12:651–670, Oct. 1996.

[10] E. Malis, F. Chaumette, and S. Boudet. 2-1/2d visual servoing. *IEEE Trans. on Robotics and Automation*, 15(2):238–250, April 1999.

[11] E. Marchand and F. Chaumette. A new redundancy-based iterative scheme for avoiding joint limits application to visual servoing. *Proc. of the IEEE Int'l Conf. on Robotics & Automation*, April 2000.

[12] Y. Mezouar and F. Chaumette. Path planning in image space for robust visual servoing. *Proc. of the IEEE Int'l Conf. on Robotics & Automation*, April 2000.

[13] B. Nelson and P. K. Khosla. Increasing the tracking region of an eye-in-hand system by singularity and joint limit avoidance. *Proc. of the IEEE Int'l Conf. on Robotics & Automation*, pp. 418–23, 1993.

[14] A. Ruf and R. Horaud. Visual servoing of robot manipulators part i: Projective kinematics. *The International Journal of Robotics Research*, 18(11):1101–1118, November 1999.

[15] C. J. Taylor and J. P. Ostrowski. Robust vision-based pose control. *Proc. of the IEEE Int'l Conf. on Robotics & Automation*, April 2000.

[16] J.J. Uicker., Jr. Matrix methods in the design analysis of mechanisms. Univ. of Wisconsin, ME 751 Course Notes, 1997.