Performance Tests for Visual Servo Control Systems, with Application to Partitioned Approaches to Visual Servo Control

Nicholas R. Gans, Seth A. Hutchinson and Peter I. Corke

The International Journal of Robotics Research 2003 22: 955
DOI: 10.1177/027836490302210011

Nicholas R. Gans
Seth A. Hutchinson

Department of Electrical and Computer Engineering
The Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
405 N. Mathews Avenue
Urbana, IL 61801, USA
ngans@uiuc.edu
seth@uiuc.edu

Peter I. Corke

CSIRO Manufacturing Science and Technology
Pinjarra Hills
AUSTRALIA 4069
pic@cat.csiro.au

Performance Tests for Visual Servo Control Systems, with Application to Partitioned Approaches to Visual Servo Control

Abstract

Visual servoing has been a viable method of robot manipulator control for more than a decade. Initial developments involved position-based visual servoing (PBVS), in which the control signal exists in Cartesian space. The younger method, image-based visual servoing (IBVS), has seen considerable development in recent years. PBVS and IBVS offer tradeoffs in performance, and neither can solve all tasks that may confront a robot. In response to these issues, several methods have been devised that partition the control scheme, allowing some motions to be performed in the manner of a PBVS system, while the remaining motions are performed using an IBVS approach.

To date, there has been little research that explores the relative strengths and weaknesses of these methods. In this paper we present such an evaluation. We have chosen three recent visual servo approaches for evaluation in addition to the traditional PBVS and IBVS approaches. We posit a set of performance metrics that measure quantitatively the performance of a visual servo controller for a specific task. We then evaluate each of the candidate visual servo methods for four canonical tasks with simulations and with experiments in a robotic work cell.

KEY WORDS—visual servoing, partitioned methods, performance, test, comparison

1. Introduction

The use of visual data for robot control has been an objective for many years. It provides dynamic information about the environment but is passive unlike sonar or laser range finding systems, which measure a reflected output signal. It also enjoys the advantage of providing sensory data with which humans are naturally familiar.

Shirai and Inoue (1973) first described the use of visual data in a feedback loop to accurately position a robot. This was the advent of visual servo control, the closed-loop control of a robot end-effector through the use of image data. While progress was initially stunted by technological limitations, the recent decade has seen phenomenal increases in computational power and digital imaging technology. The ability now exists to perform real-time image processing on detailed images using commercially available computers and cameras.

Employing these advances, visual servoing holds a great deal of potential for the control of robots in changing environments. This includes diverse applications such as industrial factories, tracking systems, and steering vehicles (Corke 1993). Visual servoing is also distinctly suited to tasks where the robot must interact or manipulate the environment rather than merely observing.

In general, there are two basic approaches to visual servo control: position-based visual servoing (PBVS; Weiss, Sanderson, and Neuman 1987; Feddema and Mitchell 1989; Martinet, Gallice, and Khadraoui 1996) and image-based visual servoing (IBVS; Sanderson and Weiss 1980; Hutchinson, Hager, and Corke 1996). In PBVS systems, features are detected in an image and are used to generate a three-dimensional (3D) model of the environment. An error between the current pose and desired pose is then computed in the Cartesian task space, and it is this error that is used by the control system.

The International Journal of Robotics Research
Vol. 22, No. 10–11, October–November 2003, pp. 955-981,

In IBVS, an error signal is measured between the image features in two images and is mapped directly to actuator commands. The choice between IBVS and PBVS offers many tradeoffs. PBVS often requires a 3D CAD model of the objects in the image. Additionally, PBVS often performs optimally in three dimensions, i.e., the motions of the robot will typically follow a least-distance path to the goal. However, PBVS surrenders control of the feature point trajectories in the image and can often fail due to lost feature points. Lastly, methods of estimating 3D motions from images are often slow and typically susceptible to noise and calibration errors.

IBVS requires no 3D reconstruction of the environment, but does require an image taken from the goal position. IBVS performs optimally in the image space and it is a simple matter to regulate the trajectory of image features, for instance preventing them from leaving the field of view. However, certain control tasks can lead to singularities in the image Jacobian, resulting in system failure. Image-based systems also surrender control of the Cartesian velocities. Thus, while the task error may be quickly reduced to zero, complicated and unnecessary motions may be performed. This is particularly troublesome when operating in a physically limited or hazardous environment. Finally, while IBVS is robust in the face of signal noise and calibration errors, it requires an estimate of the feature point depth, which may be unavailable or difficult to estimate accurately.

The characteristics of PBVS and IBVS systems have led to the creation of several partitioned methods (Malis, Chaumette, and Boudet 1999; Deguchi 1998; Corke and Hutchinson 1999). Partitioned methods use classic, Jacobian-based IBVS to control certain end-effector motions while using other techniques to control the remaining degrees of freedom. Section 2 will provide necessary background on PBVS and IBVS, as well as the use of epipolar geometry to determine camera motion. The partitioned systems will also be described using a single notational framework in Section 2.

While there have been some theoretical analyses of these systems (Chaumette 1998; Corke and Hutchinson 1999), to date, there has been little quantitative analysis of these methods. Martinet (1999) performed an experimental investigation of IBVS and several PBVS systems, but only tested three different positioning tasks. We seek to evaluate numerous systems over a wide range of motions. We present such an evaluation for the three methods in Malis, Chaumette, and Boudet (1999), Deguchi (1998), and Corke and Hutchinson (1999) along with the classic PBVS and IBVS approaches. To arrive at a meaningful evaluation, we posit a set of performance metrics that measure quantitatively the performance of a visual servo controller for a specific task. We then evaluate each of the candidate visual servo methods for four canonical tasks, under a range of experimental conditions. The metrics, canonical tasks and the experimental conditions are described in Section 3. The most interesting results from these tests will be presented in Section 4.[1]

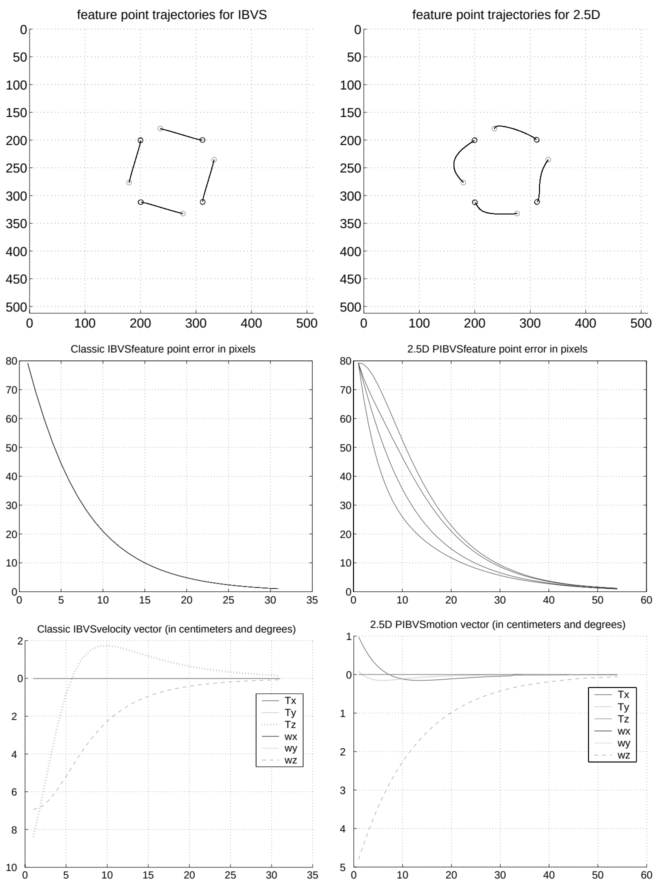

Fig. 1. Contrasting performance of IBVS and $2\frac{1}{2}$D visual servoing for optical axis rotation.

As we detail in Section 4, each system has precise strengths and weaknesses. However, these systems operate in fundamentally different ways, and were designed with different control tasks in mind. As an example, Figure 1 shows the performance of IBVS and $2\frac{1}{2}$D visual servoing (Malis, Chaumette, and Boudet 1999) for a task of rotating 60° about the optical axis. Performance here is revealed in terms of feature point trajectories, the feature point error over time, and the camera velocity over time. The systems are both able to successfully zero the feature point error, but the performed motions and the trajectory of the feature points differ greatly.

For this reason, it is difficult and often spurious to contrast the systems. We do not seek to formally rate the systems; rather the goal of this paper is to establish a basis for comparison and present typical system performance so that the proper system can be chosen to suit the task at hand. Thus, while we may truly be comparing “apples and oranges”, it

[1] The entirety of the data will be available as Matlab figures at http://www-cvr.ai.uiuc.edu/˜seth/ngans_vs.

at least becomes possible to select a good apple over a bad orange.

2. Background

In this section we present basic concepts involved in IBVS, review the established methods of determining camera motion from images using a homography matrix (a method used in several partitioned IBVS systems, and able to provide a PBVS method), and introduce the three partitioned IBVS systems—those developed by Malis, Chaumette, and Boudet (1999), Deguchi (1998), and Corke and Hutchinson (1999).

2.1. Camera Motion from a Homography Matrix

Two of the partitioned IBVS systems, and one possible PBVS system, derive specific camera motions by computing the homography matrix relating coplanar–planar feature points in two images. As in Figure 2, for the 3D points $F_i,\ i = 1, \dots, n$ lying on the plane π, we define $\mathbf{f}_i^*$ and $\mathbf{f}_i$ as the homogeneous coordinates of the corresponding points in two images. Throughout this paper, variables superscripted with * will refer to variables in the goal image and values accented with ˆ will refer to estimated values. These points are related by

$$\mathbf{f}_i = \mathbf{H}\mathbf{f}_i^*, \tag{1}$$

in which $\mathbf{H}$ is the 3×3 homography matrix. This can further be decomposed as

$$\mathbf{H} = \mathbf{R}\left(\mathbf{I}_3 - \frac{\mathbf{t}n^{\mathrm{T}}}{d}\right) \tag{2}$$

where $\mathbf{I}_3$ is a 3×3 identity matrix and $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and translation vector, respectively, relating the two camera coordinate frames, $\mathbf{n}$ is the normal of the plane π, and d is the distance from the second camera origin to the plane π.

There are numerous methods for computing $\mathbf{H}$ given two sets of image points. We have focused our attention on linear solutions since visual servoing, in general, requires quicker calculations than iterative methods may provide. The major drawback to linear methods is that they are susceptible to noise.

After computing the homography, it is necessary to decompose $\mathbf{H}$ as in eq. (2). Methods to perform this decomposition are detailed in Faugeras and Lustman (1988) and Zhang and Hanson (1996). Decomposing the homography is not a trivial exercise and generally cannot be solved uniquely. Additional information or views are required to select from multiple solutions. The sets of solutions for multiple views will typically intersect at one solution. This operation is a simple matter in visual servoing as the view from the previous iteration can be used. However, for the first iteration, it will be necessary to perform an extra motion to acquire a unique view.

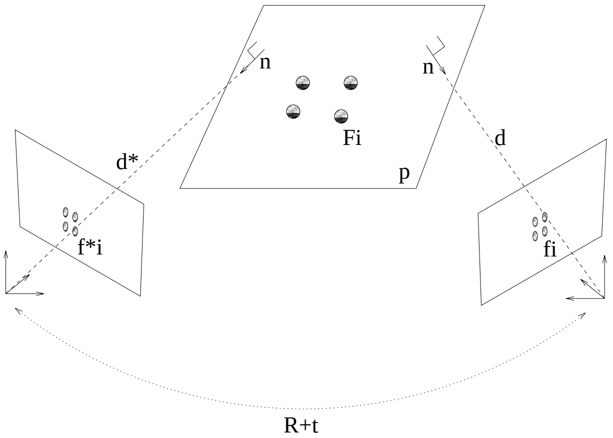

Fig. 2. Notation for two images of points in a plane.

2.2. Position-Based Visual Servoing

PBVS has a long history (Weiss, Sanderson, and Neuman 1987; Feddema and Mitchell 1989; Martinet, Gallice, and Khadraoui 1996). In PBVS the error signal exists in Cartesian space. Having stored a goal pose $\mathbf{P}^* = [X^*, Y^*, Z^*, \theta^*\mathbf{u}^*]^{\mathrm{T}}$ where $\theta\mathbf{u}$ is the vector/angle representation of the orientation (vector/angle is traditionally used in describing the control, but any method of representing the rotation matrix in three variables is allowable) and Γ is a scalar or matrix gain coefficient, we can use methods such as the homography detailed previously or that given in DeMenthon and Davis (1992) to determine the current pose of the robot $\mathbf{P} = [X, Y, Z, \theta\mathbf{u}]^{\mathrm{T}}$. We can then set a proportional control law

$$\dot{\mathbf{P}} = -\Gamma(\mathbf{P} - \mathbf{P}^*). \tag{3}$$

If we define the origin of the world coordinate frame as coinciding with the goal camera position, this simplifies to

$$\dot{\mathbf{P}} = -\Gamma(\mathbf{P}). \tag{4}$$

Using this method, we can incorporate epipolar methods such as those described above by defining the position of the goal view as the origin. We calculate $\mathbf{P}$ in eq. (4) as follows: $\mathbf{T} = [T_x, T_y, T_z]^{\mathrm{T}} = \hat{d}\,\mathbf{t}$, where $\hat{d}$ is an estimate of d. Given knowledge of the geometry of the feature point locations, it is possible to accurately estimate $\hat{d}$ and so determine $\mathbf{t}$ to the proper scale. From the rotation matrix $\mathbf{R}$, we extract the roll, pitch and yaw angles, $\omega_z, \omega_x, \omega_y$, obtaining the velocity screw $u = k[T_x, T_y, T_z, \omega_x, \omega_y, \omega_z]$ in which k is a scalar gain constant, or a 6×6 gain matrix.

We choose to explore the use of the homography in PBVS since it shares a few characteristics with IBVS such as requiring two images and depth estimation. We will refer to this method as *homographic visual servoing* (HVS).

2.3. Image-Based Visual Servoing

Let $\mathbf{r} = (x, y, z, \omega_x, \omega_y, \omega_z)^{\mathrm{T}}$ represent coordinates of the end-effector, and $\dot{\mathbf{r}} = (T_x, T_y, T_z, \Omega_x, \omega_y, \Omega_z)^{\mathrm{T}}$ represent the corresponding end-effector velocity. Let $\mathbf{f} = (u, v)^{\mathrm{T}}$ be the image-plane coordinates of a point in the image and let $\dot{\mathbf{f}}$ be the corresponding instantaneous feature point velocity. The feature point velocity can be approximated using the first-order differences by $\dot{\mathbf{f}} \approx (u - u^*, v - v^*)^{\mathrm{T}}$. For a stationary collection of 3D points, the image Jacobian relationship is given by

$$\dot{\mathbf{f}} = J(\mathbf{r})\dot{\mathbf{r}}, \tag{5}$$

with

$$J = \begin{bmatrix} \frac{\lambda}{z} & 0 & -\frac{u}{z} & -\frac{uv}{\lambda} & \frac{\lambda^2+u^2}{\lambda} & -v \\ 0 & \frac{\lambda}{z} & -\frac{v}{z} & \frac{-\lambda^2-v^2}{\lambda} & \frac{uv}{\lambda} & u \end{bmatrix} \tag{6}$$

in which λ is the focal length for the camera. Derivations of this can be found in a number of references including Espiau, Chaumette, and Rives (1992), Chaumette (1998), and Hutchinson, Hager, and Corke (1996). Concatenating the matrix J for three more feature points creates a matrix with full rank.

The simplest approach to IBVS is to merely use eq. (5) to construct the proportional feedback control

$$\dot{\mathbf{r}} = \Gamma J^{-1}(\mathbf{r})\dot{\mathbf{f}} \tag{7}$$

in which $\dot{\mathbf{f}}$ is the desired feature motion on the image plane, Γ is a gain scalar or matrix, and $\dot{\mathbf{r}}$ is the control input, an end-effector velocity. In the general case that the Jacobian is not square, the pseudo-inverse, $\mathbf{J}^+$, is used.

IBVS provides a least-distance solution to the image error. Thus, the feature points will tend to move in straight lines to their goal configurations. This often results in complicated and unnecessary camera motions.

2.4. $2\frac{1}{2}$D Visual Servoing

The first group to partition the motion calculations in a visual servo system (Malis, Chaumette, and Boudet 1999) developed a system with the goal to correct two shortcomings of IBVS. The first is the need for depth estimation when calculating the image Jacobian. The second is the fact that IBVS often does not converge to zero error when the initial position is far removed from the goal position. Deriving the rotation $\mathbf{R}$ from the homography matrix (2), and inserting it into the control law, this system is coined *$2\frac{1}{2}$D visual servoing* ($2\frac{1}{2}$D).

$2\frac{1}{2}$D uses a different feature point vector and image Jacobian than the other IBVS systems we investigate. We will denote the unique parameters with a $\sim$. The feature point velocity vector is augmented with depth and rotation information

$$\dot{\tilde{\mathbf{f}}} = (u - u*, v - v*, \log \rho, \theta\mu)^{\mathrm{T}} \tag{8}$$

where ρ is the ratio $\frac{z}{z^*}$, and $\theta\mu$ is the angle and rotation axis of the rotation matrix $\mathbf{R}$ extracted from the homography matrix as in eq. (2). Furthermore, ρ can be directly calculated from the homography matrix as

$$\rho = \det(\mathbf{H})\frac{[\mathbf{f}^{*T}, 1]\mathbf{n}^*}{[\mathbf{f}^{\mathrm{T}}, 1]\mathbf{n}}. \tag{9}$$

Malis, Chaumette, and Boudet (1999) define the motion control law as

$$\dot{\mathbf{r}} = -\Gamma\tilde{\mathbf{J}}^{-1}(r)\dot{f}, \tag{10}$$

$$\tilde{\mathbf{J}}^{-1} = \begin{bmatrix} \hat{d}^*\rho\mathbf{J}_v{}^{-1} & -\hat{d}^*\rho\mathbf{J}_v{}^{-1}\mathbf{J}_\omega \\ \mathbf{0} & \mathbf{I}_3 \end{bmatrix} \tag{11}$$

where $\mathbf{J}_v$ and $\mathbf{J}_\omega$ are the translational and rotational portions of the standard image Jacobian (6), composed of the first three and last three columns respectively, and $\hat{d}^*$ is an estimate of the distance between the focal point and the feature point plane.

Note that while four or more points are required to compute the homography matrix, the partitioned Jacobian, and therefore the velocity, are determined using only one point. In the presence of noise or quantization error, this may cause motions that would not be present in a least-squares solution.

Given its embedded depth estimation, we expect the $2\frac{1}{2}$D system to perform well given little or no external data on the distance to goal. Additionally, using motion estimation methods to compute rotations should allow it to perform well in the face of extreme rotations. However, reliance on $\mathbf{H}$ also appears to be an Achilles' heel, adding a great deal of complexity to calculate and decompose, as well as a susceptibility to noise, as discussed in Section 2.1.

2.5. The Method of Deguchi

Deguchi (1998) addresses the fact that only the translation portion of the Jacobian is dependent upon the distance to the goal. Similar to $2\frac{1}{2}$D, Deguchi's method (which we will refer to as KD) partitions the system into translational and rotational components. Solving the planar homography, as in eq. (2), delivers the scaled translation $\frac{\mathbf{T}}{d}$. The translational velocity of the end-effector is then computed as

$$\dot{\mathbf{r}}_v = [T_x T_y T_z]^{\mathrm{T}} = \hat{d}(\frac{\mathbf{T}}{d}) \tag{12}$$

where $\hat{d}$ is an estimate of the distance from the camera goal position to the plane containing the image features. The equation for the rotational velocities becomes

$$\dot{\mathbf{r}}_\omega = [\omega_x \omega_y \omega_z]^{\mathrm{T}} = \mathbf{J}_\omega^+(\dot{\mathbf{f}} - \mathbf{J}_v\dot{\mathbf{r}}_v) \tag{13}$$

with $\mathbf{J}_\omega$ and $\mathbf{J}_v$ the same as eq. (10).

Continuous depth estimation is no longer required since the distance $\hat{d}$ is a constant and needs to be generated only once. Additionally, $\hat{d}$ appears simply as a gain factor to the translational motion component; thus, it is not necessary that the estimate be highly accurate.

We expect that KD will have similar strengths and weaknesses to $2\frac{1}{2}$D. It does not require explicit depth estimation; however, the use of the homography matrix adds complexity and increased noise susceptibility.

2.6. The Method of Corke and Hutchinson

The method of Corke and Hutchinson (1999) (referred to as PC&SH) was designed to avoid the task problem due to large rotations about the optical axis. As IBVS systems proceed, the feature points converge to their goal configuration in straight lines. Thus, rotation about the optical axis will be accompanied by a backwards translation, a phenomenon referred to as *camera retreat* (Chaumette 1998; Corke and Hutchinson 1999).

As the rotation approaches 180°, this retreat approaches infinity. To counter this, Corke and Hutchinson decouple the optical axis components of translation and rotation (T_z and ω_z) from the Jacobian, and calculate these motions using simple image features.

To calculate T_z, σ is defined as the square root of the area of the regular polygon enclosed by the feature points. Here we take a departure from the technique described in Corke and Hutchinson (2000). Rather than defining

$$T_z = \gamma_{T_z}(\sigma^* - \sigma) \tag{14}$$

we define

$$T_z = \gamma_{T_z} \ln(\frac{\sigma^*}{\sigma}) \tag{15}$$

where $*$ indicates a feature in the goal image, and γ_{T_z} is a scalar gain coefficient. This delivers a T_z which varies linearly with motion along the optical axis.

Regarding ω_z, the variable θ is defined as the angle between the horizontal axis of the image and the line segment connecting two points in the image. This leads to

$$\omega_z = \gamma_{\omega_z}(\theta^* - \theta). \tag{16}$$

Defining a vector $\dot{\mathbf{r}}_z$ consisting of the translation and rotation along the optical axis, the equation for x and y motions is

$$\dot{\mathbf{r}}_{xy} = [T_x, T_y, \omega_x, \omega_y]^{\mathrm{T}} = \Gamma \mathbf{J}^+_{xy}(\dot{\mathbf{f}} - \mathbf{J}_z \dot{\mathbf{r}}_z) \tag{17}$$

where $\mathbf{J}_{xy}$ and $\mathbf{J}_z$ are matrices built from the x and y components of the image Jacobian and the z components of the image Jacobian, respectively.

3. Descriptions of Canonical Tasks, Performance Metrics, Test Conditions and the Methodology of Simulations

When comparing a diverse set of visual control systems, it is extremely important to avoid inadvertent bias. Thus, a rigid set of standards must be determined and applied throughout testing. Towards this goal, we have formulated a series of control tasks which typify common visual servo operations. In addition, we have a set of quantitative metrics with which to evaluate performance. We have also categorized work conditions that visual servo systems often experience difficulty handling.

3.1. Tasks

Many of the approaches to visual servoing have been developed in response to specific problems that are task-dependent. Therefore, to evaluate the various methods, we have selected a set of four tasks that we believe represent the most interesting tasks encountered by visual servo systems.

- **Task 1: Optical axis rotation.** The first task that we consider corresponds to a pure rotation about the optic axis. The difficulty of this task for IBVS systems was first noted in Chaumette (1998). By testing the systems over a large range of initial rotations, we can easily see how different systems perform. We evaluated performance over rotations ranging from 30° to 210°, specifically to gain insight into performance into rotations greater than 180°.

- **Task 2: Optical axis translation.** The second task is a pure translation along the optic axis, with initial positions ranging from 1 m retreated from goal to 1 m advanced through the goal. We have chosen to isolate this direction of motion, since visual servo systems are intrinsically dependent upon depth estimation. Translation perpendicular to the optical axis does not typically prove difficult for visual servo systems and is of little practical interest, thus is not considered as a separate control task.

- **Task 3: Camera *y*-axis rotation.** The third task corresponds to a pure rotation of the camera about the *y*-axis of the camera coordinate frame. These results should be indicative of rotation about any axis parallel to the image plane. The initial values range from 10° to 80° of rotation. While pure rotations of just a few degrees will remove features from the CCD array of a standard camera, in simulation we can allow ourselves an infinite image plane to perform this test.

- **Task 4: Feature point rotation/general motion.** The final task is to rotate the 3D feature points about an axis perpendicular to the optical axis and lying in the feature point plane. The visual servo system will need to perform rotation and translation in order to zero the image error. The feature points will be rotated through a range of positions from 10° to 80°. After 80° the feature points in the image plane become nearly collinear, and visual servoing becomes impossible.

Examples of the initial and goal positions for each task can be seen in Figure 3.

3.2. Performance Metrics

Before a quantitative evaluation can be performed, it is necessary to posit a set of performance metrics that can be quantitatively evaluated. We have chosen the following performance metrics for our analysis.

- **Number of iterations to convergence.** Visual servoing was considered successful and halted if the average feature point error was less than one pixel. If the average error varied less than 0.1 pixels for five iterations the system was considered to have converged to a constant value and servoing was halted. The number of iterations can obviously be increased or decreased by altering the coefficients of the gain matrix. Thus, the number of iterations itself is not extremely meaningful but proves insightful when comparing the performance of multiple systems or a system versus task errors.

- **Error at termination.** At the halt of visual servoing, the remaining pixel error of each point from its goal position was calculated. Visual servoing was halted if the error was successfully zeroed or converged to steady state, as discussed above. Additionally, if over 300 iterations had been performed without convergence visual servoing was halted. Finally, visual servoing was halted if the camera had retreated more than 10 m from the goal, advanced through 0 m depth, or if the feature points moved to more than 3000 pixels from the principle point.

- **Maximum feature excursion.** At each iteration while visual servoing, the current norm of each feature point to the principle point (center of image) was calculated. The maximum value, in pixels, attained over the entire process was then reported.

- **Maximum camera excursion.** At each iteration, the current Cartesian distance of the camera from its goal position was calculated. The maximum value attained was reported.

- **Maximum camera rotation.** Maximum camera rotation can be difficult to track as singular positions can give rise to enormous angle measures (roll/pitch/yaw angles, for instance) while resulting in little or no actual camera rotation. To gain a useful measure we transformed the rotations into single axis/angle form and tracked the magnitude of the angle of rotation. The maximum value obtained was reported.

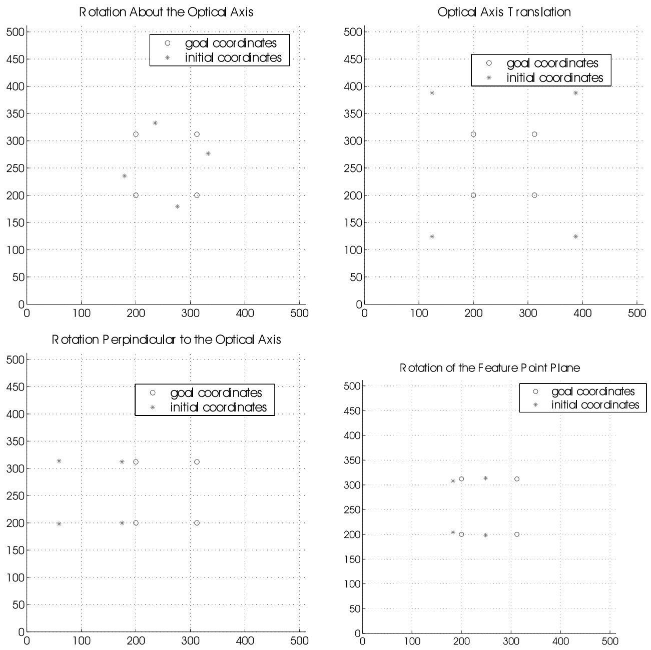

Fig. 3. Typical feature point motions during IBVS optical axis rotation.

3.3. Test Conditions

There is an extremely broad set of test conditions that can affect performance of visual servo systems. These include signal noise, camera calibration error, robot kinematic error and quantization effects. We have selected two extremely important test conditions and performed simulations of several visual servo systems under our prescribed testing methodology.

- **Noise in measured pixel coordinates.** Signal noise will cause an erroneous offset in the location of feature points. This will naturally generate errors into the calculated camera motion. To measure the effects noise can have on different systems, we simulated signal noise in feature detection by adding an offset to the image feature coordinates. This offset is a two-dimensional (2D), zero-mean Gaussian random variable with variance being ranging from 0 to 0.8 pixels. Since noise is a random process, the experiment was performed for the full range of motions and variance 100 times the results averaged in order to smooth the results and reduce the effect of outliers. These results are presented as 3D graphs with variance and the rotation/translation appropriate for the task as independent parameters.

- **Method of depth estimation.** As mentioned above, image-based approaches require some estimate of depth. Several methods have been previously investigated, including using a constant depth value (often the depth value for the goal configuration), using an estimation scheme to reconstruct the depth, or, in the case of simulations, using the exact depth. The method of depth estimation is an input parameter for our simulations. The results of these tests are 2D graphs showing the performance of each system for each task using the three different depth estimation methods with the appropriate rotation/translation as independent variable.

3.4. Simulation Methodology

All of our experiments were conducted in simulation using Matlab and the Machine Vision Toolbox and Robotics Toolbox (Corke 1996), which are publicly available at http://www.cat.csiro.au/cmst/staff/pic/. For each simulation, feature points consist of the corner points of a square in three dimensions. The square was simulated as 0.1×0.1 m^2, and the camera was positioned 1.4 m from the plane. The use of homography in three of the visual servo systems requires that the features be planar. While the placement of feature points in this plane should have little effect on system performance, using the corners of a square for feature points allows for uniformity of results that an oddly shaped configuration might not.

Images were projected using a simulated camera with 0.00001 m^2 pixels and a focal length of 0.0078 m^2. The camera plane was allowed to be infinite. However, if a feature point strayed more then 3000 pixels from the principal point, visual servoing was halted to prevent the presence of extreme outliers in the data set. Additionally, if a system had not zeroed the error or converged to steady-state error within 300 iterations, visual servoing was halted. The feature point locations were represented by floating point numbers and were not rounded, thus the effects of quantization error were ignored.

The gain for each system was a 6×6 diagonal matrix, allowing for the individual tuning of each degree of Cartesian freedom. The gains were selected in order to zero an error for the corresponding degree of freedom in approximately 30 iterations, while motion in the remaining degrees of freedom was held at zero. This is significantly faster than visual servoing systems are typically run, but represents a realistic goal of zeroing an error in 1 s of video signal.

Under several conditions detailed below, visual servoing will be halted. If the error was successfully zeroed or converged to steady state, the system will be considered to have been successful; otherwise, visual servoing will be considered to have failed. In the case that certain metrics are exceeded, visual servoing will be halted. Since these excessive results indicate an unbounded output, the system will be considered to have become BIBO unstable.

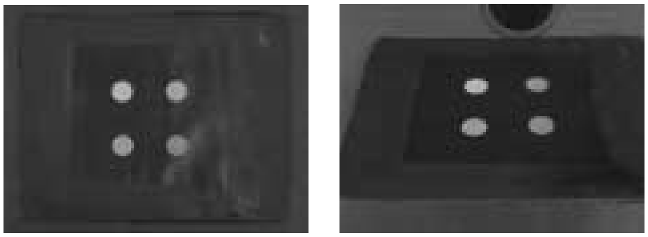

Fig. 4. Typical feature point motions during IBVS optical axis rotation.

3.5. Experiment Methodology

We performed experiments to verify the simulated results for different depth estimation. It is not practical to experimentally verify the reaction to increased noise since this would involve adding artificial offsets to the feature point locations, which is little different than what was done in the simulations. Furthermore, the time necessary to perform Monte Carlo analysis of the random effects of noise is extravagant.

The experiments were performed on a PUMA 560 robot arm, which has six degrees of freedom. The camera was a Sony V-500 with a 1.2 mm focal length lens. The images consisted of four colored dots on a black background, arrayed at the corners of in a 5.08 cm square. The feature points consisted of the centers of these dots, and could generally be isolated to within 0.1 pixels. The image was centered to within one pixel in the image. The goal position was approximately 60 cm above the feature point plane. The camera was coarsely calibrated using the methods detailed in Tsai (1987) and Lenz and Tsai (1989). No averaging was done to reduce the effects of signal noise, but obvious outliers were not included in the final results.

An example of the initial and goal images can be seen in Figure 4.

Visual servoing proceeded until all feature points were within one pixel of their goal positions (zeroed the error), the average pixel error had not deviated more than within 0.5 pixels for more than ten iterations (steady state), more than 500 iterations were performed (timeout), the robot reached its joint limits (failure) or a feature point was lost (failure).

4. Results

In this section we present the data collected for each of the four systems in Section 3: classic IBVS; $2\frac{1}{2}$D visual servoing ($2\frac{1}{2}$D) of Malis, Chaumette, and Boudet (1999); the method

(KD) of Deguchi (1998); and the method (PC&SH) of Corke and Hutchinson (2000). Each system was tested for the four control tasks described and test conditions detailed in Section 3. Each division of this section will detail one control task under one of the test conditions. At the beginning of each division, we briefly present notable results, followed by separate subdivisions for each performance metric detailing individual system performance.

Graphs are presented in groups of four, one for each system. They are ordered, in clockwise fashion, as IBVS, KD, PC&SH, and $2\frac{1}{2}$D. Tests conducted under the effects of additive white noise are presented as 3D graphs with the noise variance increasing along the right-most axis while the independent variable (amount of rotation or translation) increases along the left axis. The dependent variable (the performance metric) detailed in each section increases along the vertical axis. Tasks performed under differing depth estimation methods are 2D graphs with overlaid plots for each method. The independent variable increases along the horizontal axis, while the dependent variable under consideration increases with the vertical axis. Note that results for each system can vary to such a degree that the dependent variable is often plotted on a different scale.

IBVS has noted difficulty for severe rotations about the optical axis and when the current view is far removed from the goal position. We thus expect it to have difficulty for most extreme motions. Both $2\frac{1}{2}$D and KD decouple the rotation and translation motion and generally perform well, but both suffer worse from the effects of noise than the other systems due to the susceptibility of the homography matrix computations to noise.

The fact that the translational component of the homography matrix is affected by noise worse than the rotational component causes KD to suffer the worse detriment. PC&SH was designed specifically to address the difficultly IBVS experiences during large rotations about the optical axis. The z-axis rotations and translation are dependent upon the relative positions of the feature point, which will cause problems when rotations about the x-axis and/or y-axis deform the locations of feature points. We expect its performance to suffer during these motions.

Both KD and $2\frac{1}{2}$D were also designed to remove explicit dependence upon depth estimation for control, so we expect good performance under a variety of depth estimation schemes, including the use of constant depth. KD does perform quite well for all depth estimations, while $2\frac{1}{2}$D typically shows solid performance for constant methods but markedly better when using true depth. It is somewhat misleading to claim that IBVS and PC&SH are dependent upon depth information. While the current depth does explicitly appear in the Jacobian, any chosen value will merely have the effect of scaling the translation. A poor choice of depth estimation may indeed result in failure, but convergence will be possible for an astute choice.

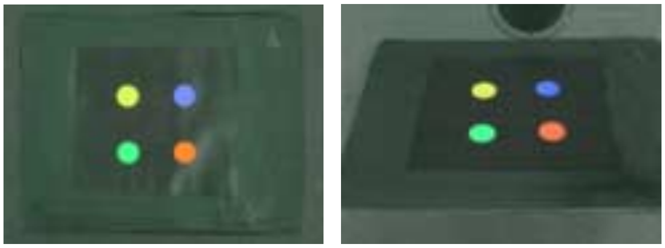

Fig. 5. Typical feature point trajectories during IBVS optical axis rotation.

For the tests conducted under the effects of additive white noise, the noise variance increases along the right-most axis while the independent variable (amount of rotation or translation) increases.

4.1. Simulation: Rotation About the Optical Axis with Noise

Rotation about the optical axis presents a classical problem for IBVS for large values of the rotation angle. Since IBVS attempts to zero the error of each feature point in a least-distance manner (a straight line) the camera will retreat then move forward again as it rotates. Feature point trajectories resulting from initial positions corresponding to a moderate and a large optical axis rotation are shown in Figure 5.

For large rotations the camera will theoretically experience an infinite retreat resulting in a Jacobian singularity. In reality, the camera will usually reach the limits imposed by the particular kinematics of the robotic system used. HVS, being a PBVS system, should not suffer any unnecessary motions. The partitioned systems we are considering decouple the rotation from the translation motions, they will not suffer from camera retreat and are expected to succeed for a full range of motions.

4.1.1. Remaining Pixel Error

We begin with the remaining pixel error, as seen in Figure 6. IBVS is able to zero the remaining pixel for rotations under approximately 160°, even under the effects of increasing levels of noise, as described in Section 3.3. There is an extremely sharp increase in error after 160°. At this point the camera retreat has essentially brought the feature points to the principle point and cannot zero the error. These cases also show a slight increase in pixel error as noise levels increase.

HVS shows moderate dependence upon noise levels, but no dependence upon the angle of rotation. KD is able to reduce the pixel error to zero for all rotations except those extremely close to 180°. Pixel error increases rapidly with noise for these failed cases.

The system $2\frac{1}{2}$D shows a slightly more complicated result. It is able to zero the error for all rotations under low noise.

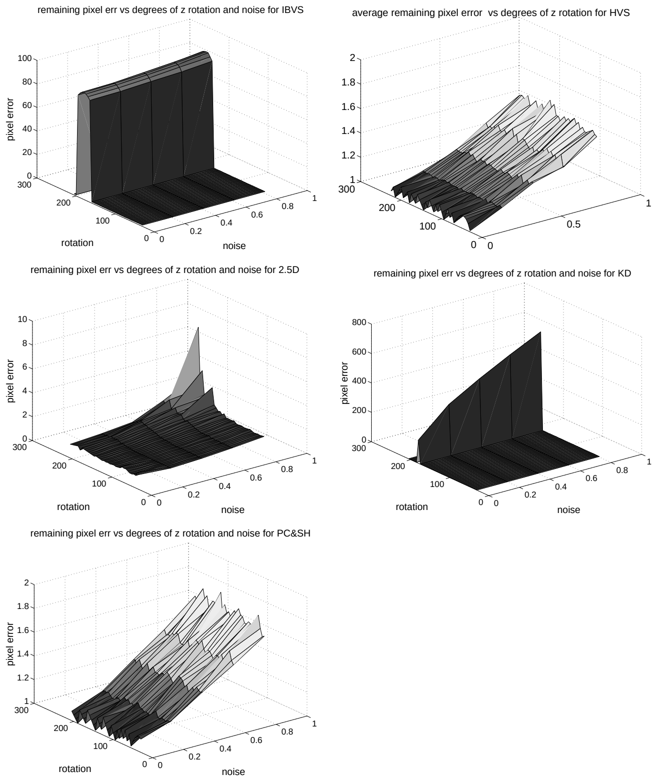

Fig. 6. Average remaining pixel error versus degrees of z-axis of rotation versus noise variance.

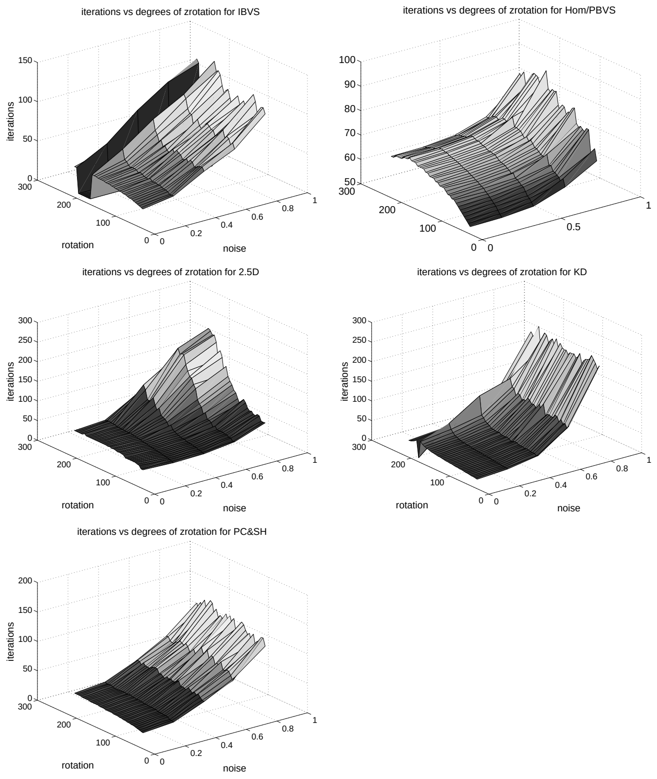

Fig. 7. Average iterations until convergence versus degrees of z-axis of rotation versus noise variance.

However, there is an increase in pixel error as noise levels increase. Under higher noise levels, pixel error begins to show a dependence on the angle of rotation as well. The greatest error is below seven pixels, much lower than the failed cases of IBVS or KD. The jagged appearance of this graph suggests the presence of outliers.

Finally, PC&SH is able to greatly reduce the pixel error for all rotations, with no increase as rotation increases. There is a slight increase as noise increases but of only a few pixels. This indicates that the system converges to constant error for all levels of rotation.

4.1.2. Number of Iterations Until Convergence or Failure

We now turn our attention to the average number of iterations until convergence or failure, as seen in Figure 7. IBVS undergoes an increase in iterations as noise increases. At peak noise it requires about four times as many iterations to converge as for zero noise. There is also a shallow increase in iterations as rotation angle increases. After about 160° (mirroring the sharp increase in remaining pixel error) the number of iterations drops immediately to a negligible number, indicating that failure generally occurs very quickly.

The number of iterations for HVS slightly increases as the rotation angle increases. Additionally, noise levels over 0.5 pixels cause a sharp increase in the iterations required for convergence.

KD shows an extreme increase in iterations as noise levels increase, requiring about twelve times as many iterations under high noise. There is a shallow increase in iterations as rotation increases, with a brief spike towards zero very near 180°.

The $2\frac{1}{2}$D system shows positive dependence upon both noise and rotation. These increases are fairly shallow if only one of the independent variables is large; however, under large rotation and high noise levels the number of iterations rises dramatically, up to about ten times greater than the lowest value.

PC&SH shows a very slight increase in iterations as rotation approaches 180° and a sharper increase as noise levels grow, requiring perhaps four times as many iterations under high noise levels than at zero noise.

4.1.3. Maximum Feature Point Excursion from the Principal Point

The maximum excursions of feature points from the principal point during this test are shown in Figure 8. Since this test

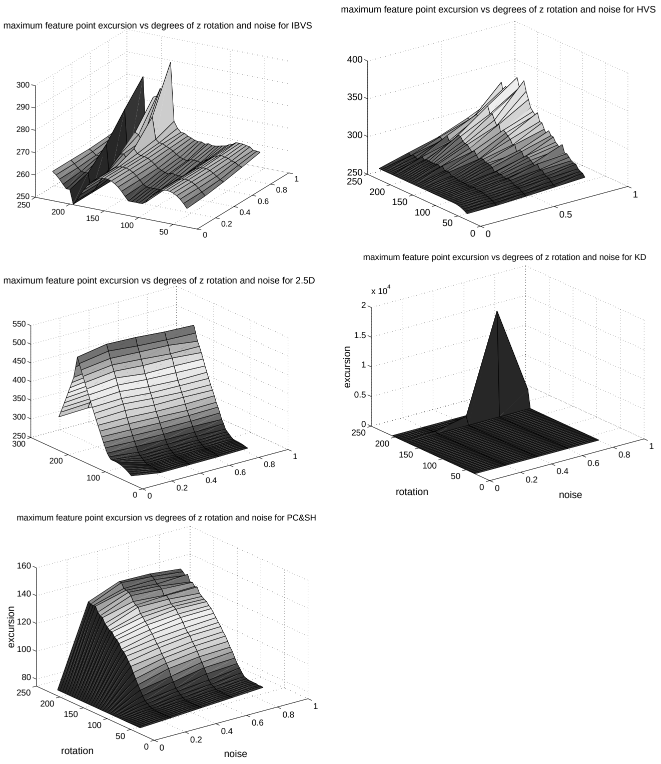

Fig. 8. Average maximum feature point excursion versus degrees of z-axis rotation versus noise variance.

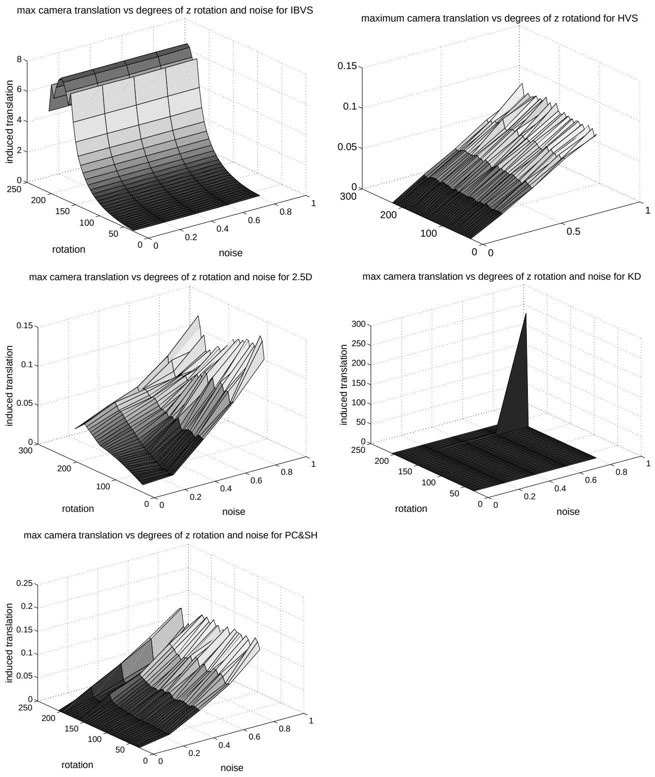

Fig. 9. Average maximum camera translation versus degrees of z-axis rotation versus noise variance.

involves rotation of the camera about the optical axis, and since the feature points are symmetric about the principle point in the goal image, we expect the graphs to normally be very flat as the maximum feature point excursions should correspond with the initial and final positions. At the goal configuration, the distance of the feature points from the principal point is 79 pixels.

IBVS is very flat for lower rotations, showing only a slight increase with increasing noise levels. At approximately 160° it takes a sharp increase followed by a sharp decrease.

The graph for HVS is flat for no noise, but for higher noise levels the amount of feature point excursion increases as the amount of rotation increases.

KD is extremely flat for almost the entire range of rotations, and any dependence upon noise is obscured by the large scale of this graph. For values very near 180° and significant noise, there is a ridge of extremely high feature point excursion, as high as several thousand pixels. This indicates an unbounded output as the cause of failure.

The $2\frac{1}{2}$D system undergoes a steady increase in maximum feature point excursion as the amount of rotation increases; this indicates that the feature points are undergoing large motions in the image plane.

PC&SH has little excursion for the entire range of rotations when noise levels are low. However, as noise levels increase, there is a sharp increase in the feature point excursion for rotations over 100°.

4.1.4. Maximum Camera Translation

During pure rotation about the optical axis, the maximum camera translation is arguably the most interesting feature. Plots for this metric are seen in Figure 9.

As expected, IBVS has a substantial amount of camera retreat. The maximum camera translation increases in a roughly exponential fashion as the camera rotation increases and camera retreat becomes more significant. At about 165° the camera retreat exceeds 10 m from the goal, and visual servoing is halted.

HVS has minimal camera translation, as would be expected for a PBVS method. There is a slight increase as noise levels cause erroneous motions.

KD has one extremely sharp point of camera motion at 180° of rotation and 0.8 pixel standard deviation white noise. This is when the most strenuous conditions are experienced and, likely, an outlier results from an unbounded output.

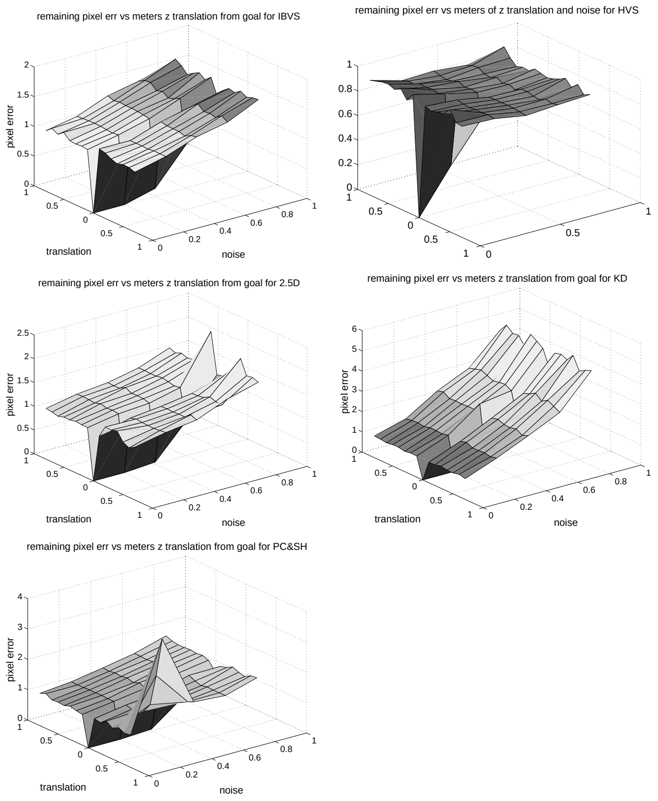

Fig. 10. Average remaining pixel error versus meters of translation versus noise variance.

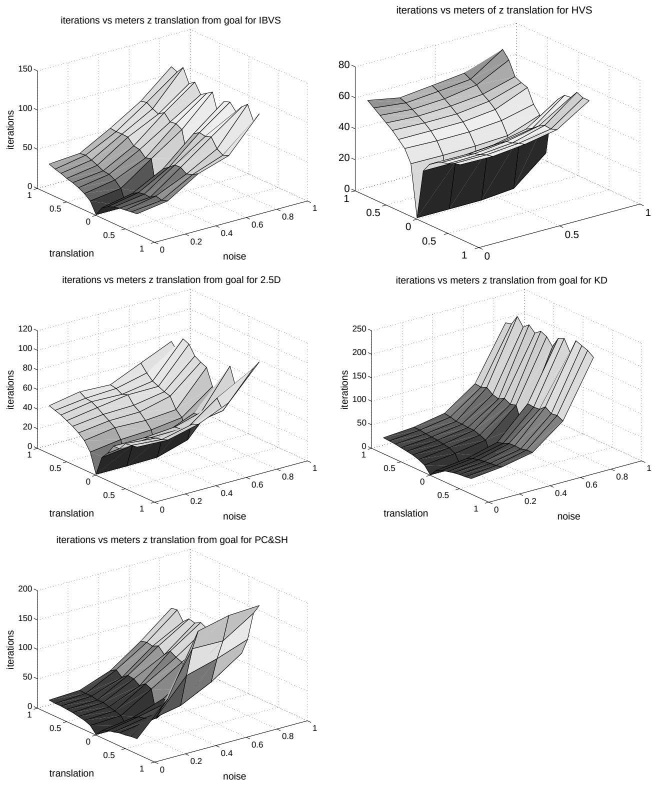

Fig. 11. Average iterations until convergence versus meters of translation versus noise variance.

Both PC&SH and $2\frac{1}{2}$D suffer negligible camera motion while experiencing slight increases as noise increases, and $2\frac{1}{2}$D shows a slight increase as rotation approaches 180°.

4.2. Simulation: Translation Along the Optical Axis with Noise

There are no stability issues for translation along the optical axis, and the systems perform very similarly for most performance metrics. Indeed, IBVS and $2\frac{1}{2}$D share the same function to generate translation velocity, so we expect them to operate nearly identically. Distinguishing results include the convergence rate of PC&SH which is not symmetric for negative and positive translation distance and KD's large increase in pixel error and iterations as noise increases.

4.2.1. Remaining Pixel Error

Graphs of remaining pixel error, as seen in Figure 10, have a similar appearance for all systems. IBVS shows a slight positive correlation with increasing noise, but no correlation to translation distance or direction. The $2\frac{1}{2}$D system has almost identical performance with the presence of a few spikes during higher noise. This is hardly surprising as $2\frac{1}{2}$D uses the IBVS translation Jacobian.

HVS shows no real correlation to noise or the amount of rotation, the graph is flat except for the natural dip at 0 m of translation. KD appears to only have correlation with increasing noise, although pixel error increases much more steeply than with the other systems. PC&SH appears to perform slightly better versus noise than IBVS and $2\frac{1}{2}$D, with the exception of a spike at -1 m translation and 0.2 pixel variance white noise. This is likely due to the presence of an outlier.

4.2.2. Number of Iterations Until Convergence or Failure

The graphs for the number of iterations are very similar for each system, as seen in Figure 11. For low noise levels IBVS, HVS, KD, and $2\frac{1}{2}$D all reveal a roughly inverse exponential increase as distance from goal increases. Additionally, the number of iterations as a function of distance appears to be symmetric for both positive and negative translations. Each of these systems has an increase in iterations as the amount of noise increases.

HVS seems to be the least affected by noise. For the others, the effects of noise eventually dominate any dependence on translation distance. KD, by far, suffers the worst increase in iterations as noise levels increase, exhibiting an apparent exponential relation. The $2\frac{1}{2}$D system appears to have only a slight average increase with rising noise, but there are several spikes at higher noise levels, indicating occasional runs that required a larger number of iterations.

PC&SH alone has a non-symmetric graph. Negative translations (camera retreats) require more iterations to zero. Additionally, negative translations suffer more from increasing noise as far as the number of iterations is concerned. This because the function governing z-translation in PC&SH is not perfectly symmetric about zero, i.e. the rate of change of the polygonal area of the feature points changes at a different rate for advancement and retreat of the camera. Using a logarithmic function of the areas lessens this effect, as would using different gains for forward or backward motion.

4.3. Simulation: Rotation About an Axis Perpendicular to the Optical Axis with Noise

It is important to note that this experiment could not be performed with a real-world system, as large rotations without corresponding translation would remove the feature points from a real camera's imaging surface. However, it does shed light on the systems' abilities to zero errors corresponding to large rotation.

IBVS is known to perform poorly when the initial image is far separated from the goal image, and indeed tends to fail completely for rotations over 35°. PBVS is regarded to perform well when a large motion separates the goal from the current position, so we expect HVS to perform fairly well.

By decoupling the translation and rotation components of motion, $2\frac{1}{2}$D and KD are able to achieve superior performance. The $2\frac{1}{2}$D system remains stable as rotation increases, while KD performs very well up to about 75°, after which the system becomes unstable. PC&SH will experience problems as large rotation alter the feature points and the relative positions, causing false motions along the optical axis. While PC&SH remains stable over the whole range, it generally has a larger error than other systems for the same amount of rotation (assuming they are still stable as well).

4.3.1. Remaining Pixel Error

The average remaining pixel error is seen in Figure 12. IBVS is able to zero the error until about 35°, at which point the pixel error becomes erratic but generally very high, indicating the presence of a large number of outliers and unstable, unbounded outputs. There is no noticeable correlation to noise.

HVS experiences a slight increase as the noise levels increase, but with the exception of one outlier, no discernible relation to the angle of rotation.

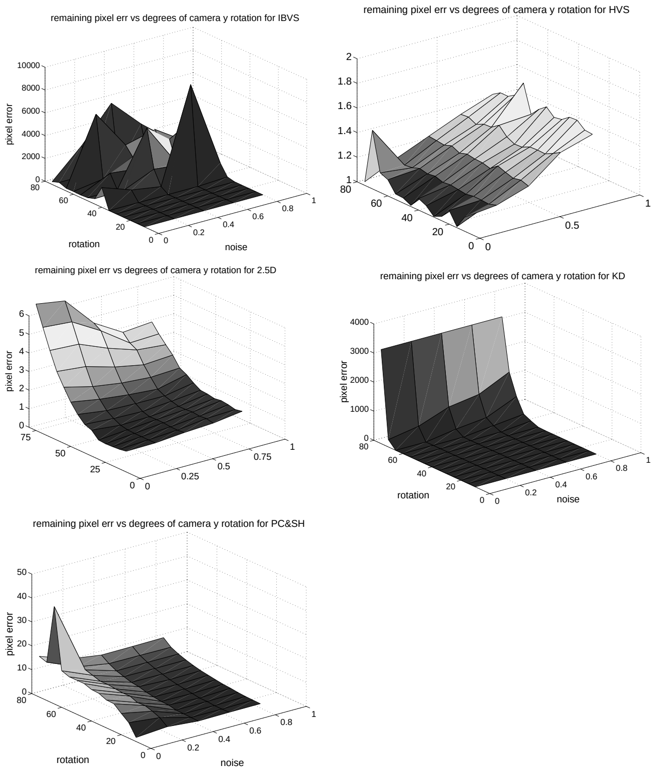

Fig. 12. Average remaining pixel error versus degrees of y-axis rotation versus noise variance.

The $2\frac{1}{2}$D system can zero the error up to 50°, with a slight increase as noise increases for these smaller rotations. After 50° the error begins to increase with rotation. However, this system still performs quite well, with the greatest error being under six pixels.

KD is able to zero the error for the longest range, up to almost 75°, with no visible correlation with noise. After this point, however, it becomes extremely unstable and the remaining pixel error immediately streaks upwards.

PC&SH shows good stability, with no unbounded outputs for the entire range of rotations and noise levels. This is at the price of a generally higher pixel error even for lower rotations when other systems could successfully zero the error. There is also a spike at about 65° and zero error variance, indicating an outlier.

4.3.2. Number of Iterations Until Convergence or Failure

The number of iterations, seen in Figure 13, mirrors the pixel error results. IBVS starts at about 150 iterations for most error levels even for small rotation, and increases until about 35°, where it plunges to zero, indicating that the unbounded outputs seen in the pixel error occur very early during visual servoing.

The number of iterations HVS requires increases smoothly as the amount of rotation increases, and also slightly as the amount of noise increases.

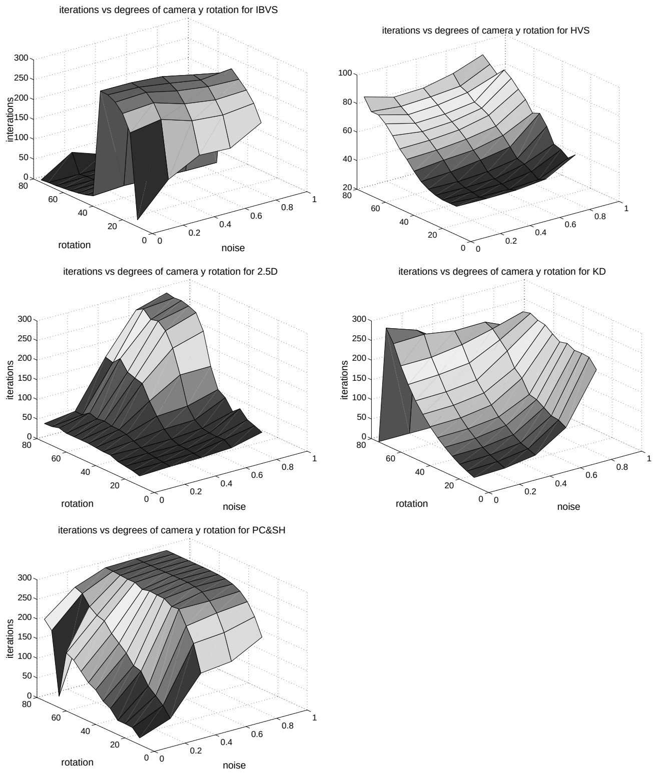

Fig. 13. Average iterations until convergence versus degrees of y-axis rotation versus noise variance.

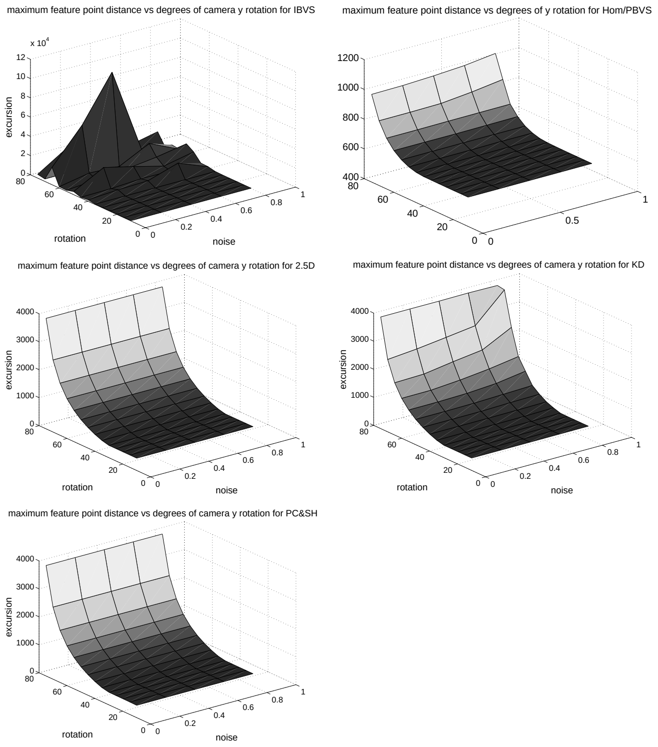

Fig. 14. Average maximum feature point excursion versus degrees of y-axis rotation versus noise variance.

The number of iterations for KD shows a correlation upon both rotation and noise, with the greater dependence being on the amount of rotation. This trend continues until about 75° where it plunges, indicating, as with IBVS, that unbounded output occurs very early.

The $2\frac{1}{2}$D system has a rather flat region when either rotations or noise levels are low, with a sharp increase as both begin to increase together. PC&SH also has a positive dependence upon both rotation and noise variance, although the noise level dependence begins and remains higher then with the other systems. There is a single dip corresponding to the spike in pixel error.

4.3.3. Maximum Feature Point Excursion from the Principal Point

The graphs of system performance for the metric of maximum feature point excursion are seen in Figure 14. IBVS has minimal feature point excursion up to about 35°. After this point there is a rough, but generally very large amount of excursion. There is no discernible dependence on noise.

HVS, KD, $2\frac{1}{2}$D, and PC&SH have flat maximum feature point excursions up to around 40°. After this point the error exponentially increases with rotation angle with no discernible dependence on noise. These results most likely correspond to the initial feature point positions. As the angle of rotation increases, the initial positions will become increasingly distant from the principal point, and the maximum feature point excursion will increase as well.

4.3.4. Maximum Camera Translation

For maximum camera translation, refer to Figure 15. IBVS shows characteristics similar to earlier graphs. Values are constant for small rotations with no noticeable dependence, but after 35° there is a large increase in camera translation, indicating that the unbounded outputs and system failure manifest themselves as large camera translations.

KD shows no translation at all for minimal noise. In fact, there is no noticeable dependence at all on rotation, but a small, linear increase in translation as noise increases. In contrast, HVS, $2\frac{1}{2}$D and PC&SH all show a slight positive dependence on the amount of rotation, but none upon increased noise.

4.4. Simulation: Rotation of Feature Points with Noise

Rotation of the feature points about the world frame results in an image displacement requiring potentially significant

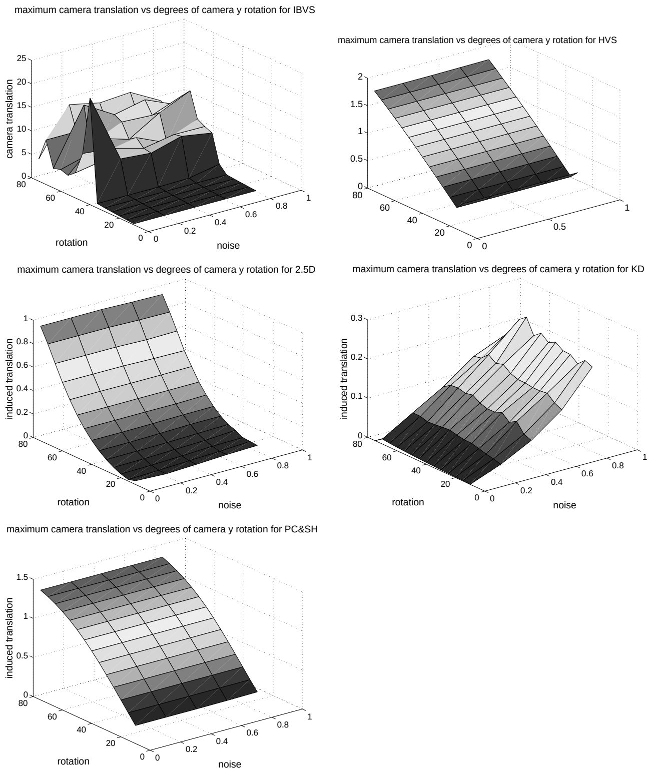

Fig. 15. Average maximum camera translation versus degrees of y-axis rotation versus noise variance.

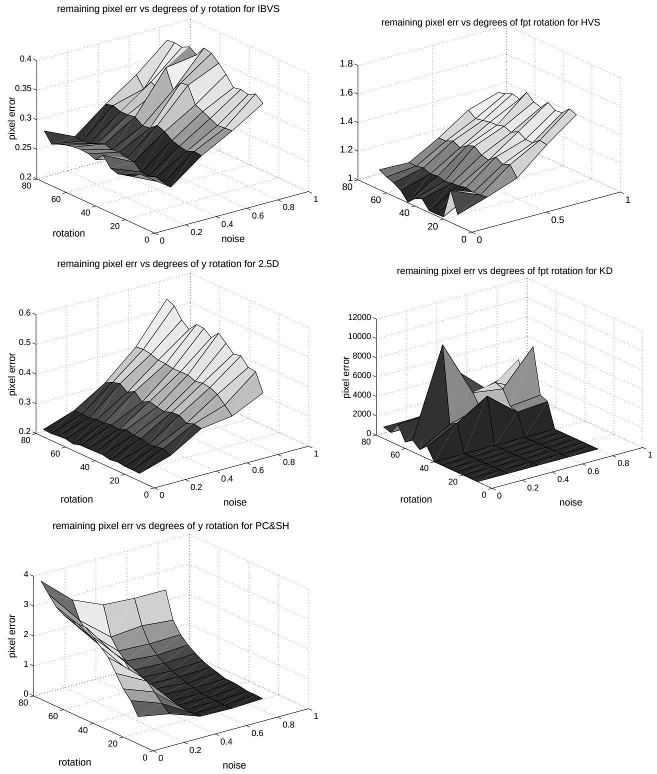

Fig. 16. Average remaining pixel error versus degrees of feature point rotation versus noise variance.

rotation and translation about all of the camera axes. This represents some of the most strenuous tasks available. The individual systems will be susceptible to many of the same issues as they were during camera rotation about an axis perpendicular to the optical axis. That is, IBVS is known to have stability problems when large motions are required, and PC&SH will suffer as the polygon enclosed by the feature points changes shape. HVS should be able to handle large motions well, with a potential weakness to noise effects.

4.4.1. *Remaining Pixel Error*

The remaining pixel error for this test is seen in Figure 16. IBVS performs quite well; it is able to greatly reduce the error for the entire range of motions and for all levels of noise variance. In general, as noise increases, the amount of pixel error increases, but the highest value is still below 0.4 pixels. There does not appear to be a correlation with the amount of rotation during low noise levels. However, for higher noise levels, a slight, positive dependence upon the rotation appears to take hold.

HVS has a remaining pixel error slightly above 1 pixel, even when there is no noise. As the noise levels increase, the remaining error increases slightly as well in a linear manner.

KD is able to greatly reduce the amount of error up to 40°, at which point the remaining error is extremely high and irregular, indicating unbounded outputs. The graph of performance for $2\frac{1}{2}$D is very similar in appearance to that of IBVS. There is a positive dependence upon noise levels, and a slight dependence upon the magnitude of the motion. The greatest error is below 0.6 pixels.

PC&SH has a strong, positive correlation with increasing feature point rotation as the feature point polygon is altered, but a slight negative correlation with increasing noise. This is certainly a curious result, but has a simple explanation. The system is able to quickly reduce the error to a certain level, but has trouble reducing any further. With lower levels of noise, the system reaches convergence and servoing is halted. However, when noise variance is higher, the feature points do not converge, and servoing continues and the system is able to slightly reduce the error further. An example of this phenomenon can be seen in Figure 17.

4.4.2. *Number of Iterations Until Convergence or Failure*

The iterations needed for convergence are shown in Figure 18. IBVS appears to have an inverse exponential increase as the amount of rotations increase, as well as a very slight negative correlation with increasing noise levels. HVS has a moderate

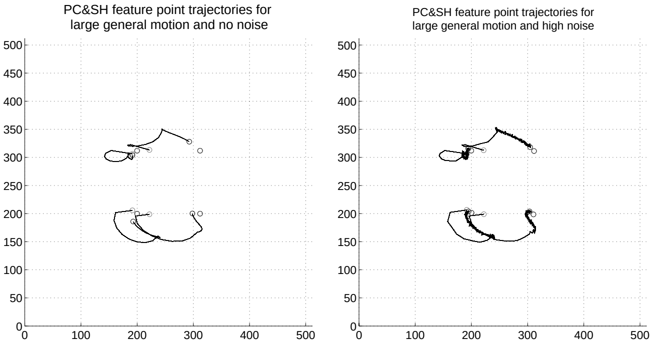

Fig. 17. Feature point trajectories in the image plane for PC&SH performing a large general motion under different noise levels.

dependence on both the amount of rotation and noise levels. At higher noise levels, it requires more iterations than the other systems.

KD has a positive dependence upon both rotation angle and noise variance up to 40°. After this, the number of iterations quickly drops, indicating that the unbounded outputs occur early. The $2\frac{1}{2}$D system has a very slight positive dependence on rotation amount, but this is overshadowed by a stronger dependence upon noise level. PC&SH has a fairly strong positive dependence upon both the motion and noise variance values.

4.5. Simulation: Rotation About the Optical Axis with Differing Depth Estimation

For the most part, using different depth estimation methods results in little change during pure rotation about the optical axis. This is not surprising as the goal and initial distance are the same, and thus the true depth should also remain the same at every iteration. Indeed, many of the plots for the different depth estimation methods align almost perfectly and are difficult to differentiate.

IBVS has proven difficulties when it comes to handling large rotation angles, but the remaining error is dramatically reduced at the expense of increased convergence rate if a constant depth estimation is used rather than true depth, since camera retreat will be restrained. The other systems typically perform the same regardless of the depth estimation technique used. KD and $2\frac{1}{2}$D can zero the error for all rotations but 180°. HVS and PC&SH are able to zero the error for the entire range of rotation.

4.5.1. Remaining Pixel Error

The remaining pixel error is shown in the graphs of Figure 19. When using true depth, IBVS fails after 165°. If either constant depth is used, however, the error is zeroed for all but 180°. The $2\frac{1}{2}$D system effectively zeros the error for all depth estimation methods.

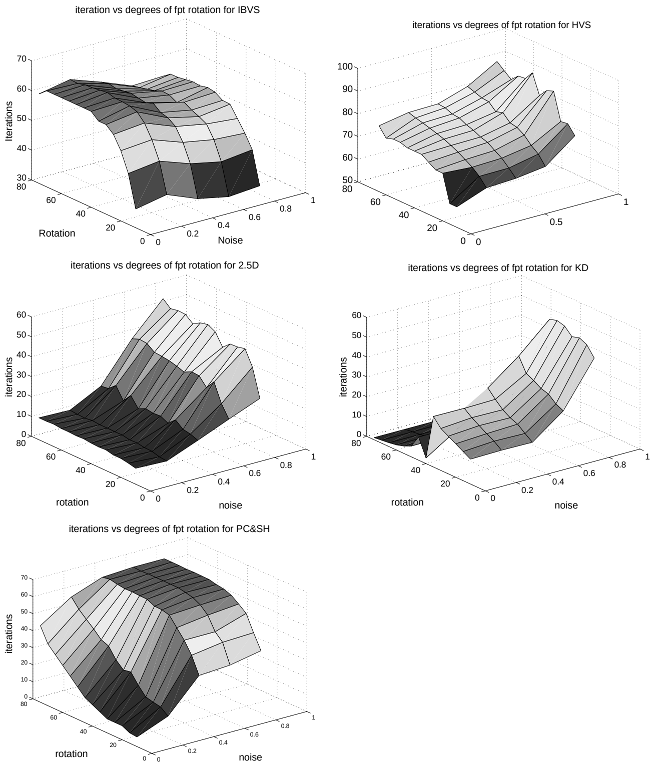

Fig. 18. Average iterations until convergence versus degrees of feature point rotation versus noise variance.

The performance of KD, HVS and PC&SH is not dependent upon the choice of depth estimation. While KD fails for rotations of 180°, HVS and PC&SH are able to zero the error over the entire range.

4.5.2. Number of Iterations Until Convergence or Failure

Graphs of the number of iterations needed for convergence are seen in Figure 20. All systems exhibit a nonlinear increase in iterations as the rotation angle increases. Using a constant depth estimate for IBVS results in a system that requires more iterations to convergence, but grants the ability to converge for rotations that fail using true depth. The $2\frac{1}{2}$D system, for the most part, requires slightly more iterations if using a constant depth estimate.

The performance of KD is not affected by the choice of depth estimation and the three plots are indistinguishable. The results of KD are very similar to those for IBVS using a constant depth estimation, which is to be expected since they both use the rotation portion of the image Jacobian.

HVS and PC&SH also perform identically for all depth estimation schemes; they both show a general nonlinear, possibly inverse exponential increase.

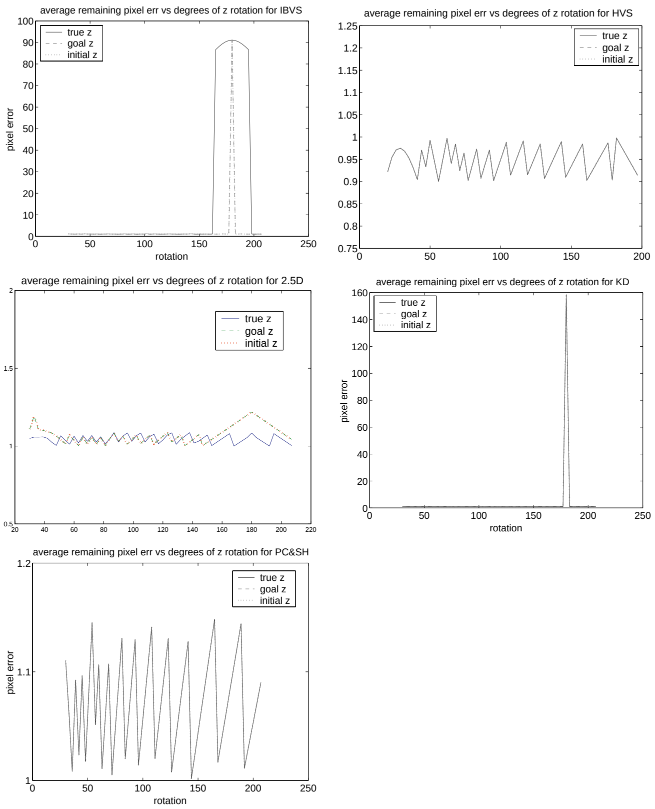

Fig. 19. Average remaining pixel error versus degrees of z-axis rotation with differing depth estimation.

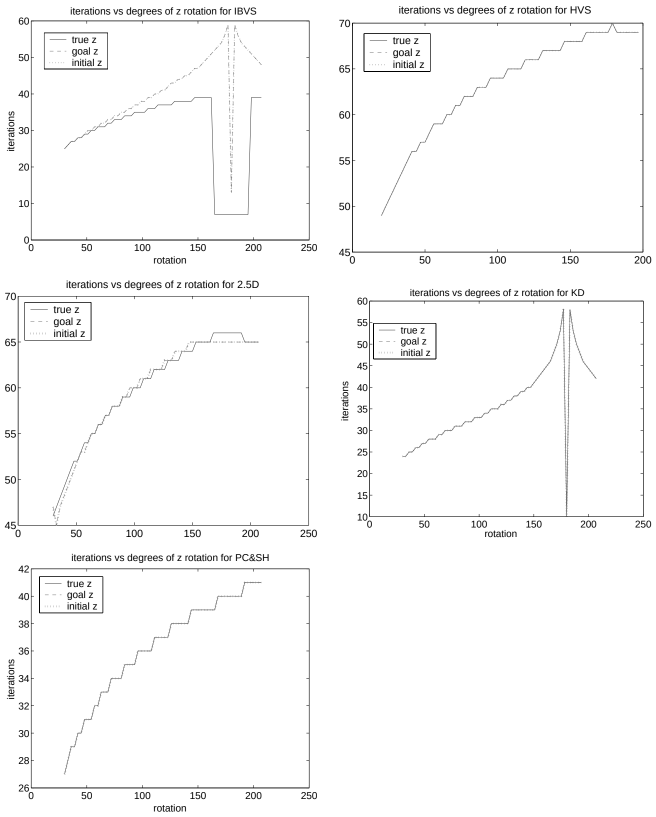

Fig. 20. Average iterations until convergence versus degrees of z-axis rotation with differing depth estimation.

4.5.2. Maximum Camera Ttranslation

Maximum camera translation is shown in Figure 21. IBVS experiences an exponentially increasing camera translation as camera retreat becomes more prevalent at large rotations due to the previously discussed phenomenon of camera retreat. Using a constant for depth estimation slows the growth of this translation, allowing the system to converge for all values except 180°. $2\frac{1}{2}$D undergoes an approximately linear increase in camera translation as task rotation increases, reaching a maximum of about 0.225 m at 180°. Constant depth estimation methods result in a larger translation for lower rotations than using true depth and, in contrast to IBVS, the translation performed by $2\frac{1}{2}$D is exclusively in the xy plane. HVS, KD and PC&SH experience zero motion for all amounts of rotation.

4.6. Simulation: Translation Along the Optical Axis with Differing Depth Estimation

In the case of a pure translation along the optical axis, the use of different depth estimation should have little effect other than to scale the velocity vector. Thus we expect that the only major difference in performance for each system should appear in the number of iterations to goal. Indeed, for all systems except PC&SH using the initial depth results in significantly more iterations. PC&SH does not use depth estimation when calculating translation along the optical axis, thus there is no cost or benefit for the choice of depth estimation. Additionally, HVS and KD experience a slight increase in remaining pixel error if initial depth is used.

4.6.1. Remaining Pixel Error

Graphs for performance in terms of remaining pixel error are seen in Figure 22. In general, the remaining pixel error does not vary much for differing depth estimation. Generally, the amount of remaining error varies by only a few tenths of a pixel, and often the plots overlap and are indistinguishable. HVS and KD experience a slightly greater increase than the others when using the initial depth as a constant depth estimate, with an increased error from 0.5 to 1 pixel. Initial depth will have the effect of reducing the translation velocity, and perhaps prevents these systems from reaching zero error.

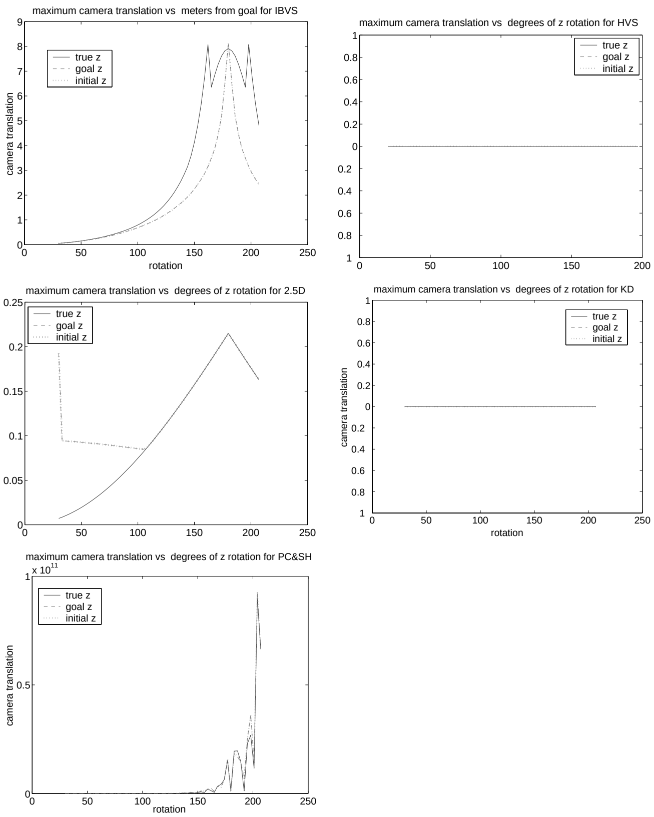

Fig. 21. Maximum camera translation versus degrees of z-axis rotation with differing depth estimation.

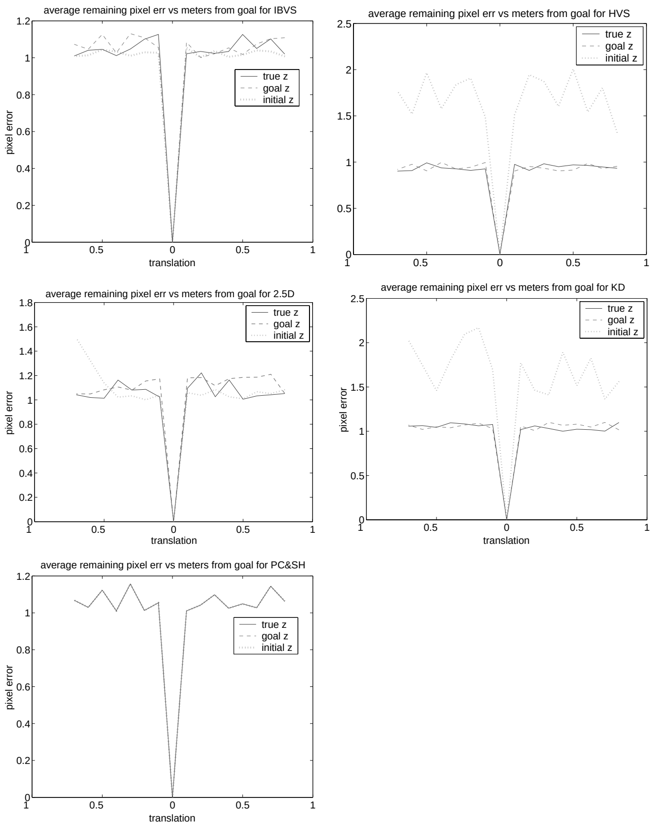

Fig. 22. Average remaining pixel error versus meters of z-axis translation with differing depth estimation.

4.6.2. Number of Iterations Until Convergence or Failure

In Figure 23 we present results of the number of iterations needed for convergence. The shapes of the graphs are very similar for all systems, although convergence rates do vary between systems. For IBVS, HVS, KD, and $2\frac{1}{2}$D, using the initial depth as a constant depth estimate requires about twice as many iterations to converge. This is simply because the initial depth is larger then either the goal or instantaneous depth, which effectively reduces the gain on the translation parameters. PC&SH does not use depth estimation for calculating the motion along the optical axis, thus it is immune for all these effects.

4.7. Simulation: Rotation About a Perpendicular Axis with Differing Depth Estimation

Since depth estimation affects only the translational velocity, it is not expected to have a great effect on performance for a pure rotation such as this. However, there are several notable results. Using a constant value rather than true depth extends the range of stable recovery for IBVS by nearly 15°. HVS and KD reveal almost no relation to the depth estimation method, which is not surprising since rotation is completely decoupled from translation, and only translation is affected by the depth estimation scheme.

In contrast, the use of constant depth increases the remaining pixel error for $2\frac{1}{2}$D for almost all cases and causes it to fail about 20° sooner. PC&SH does pick up a few outliers of large motions if a constant depth is used, particularly at large values of initial rotation.

4.7.1. Remaining Pixel Error

The remaining pixel errors for each system are shown in Figure 24. For IBVS there is little difference in performance for different depth estimation methods up to about 35°. After this point, the error when using absolute depth immediately becomes very high, whereas using constant depth allows good performance for an additional 10°. Using the initial distance as a depth estimate generally results in greater error, particularly for larger rotations.

The $2\frac{1}{2}$D system shows remarkably better error reduction using absolute depth as an estimate rather than constant depth.

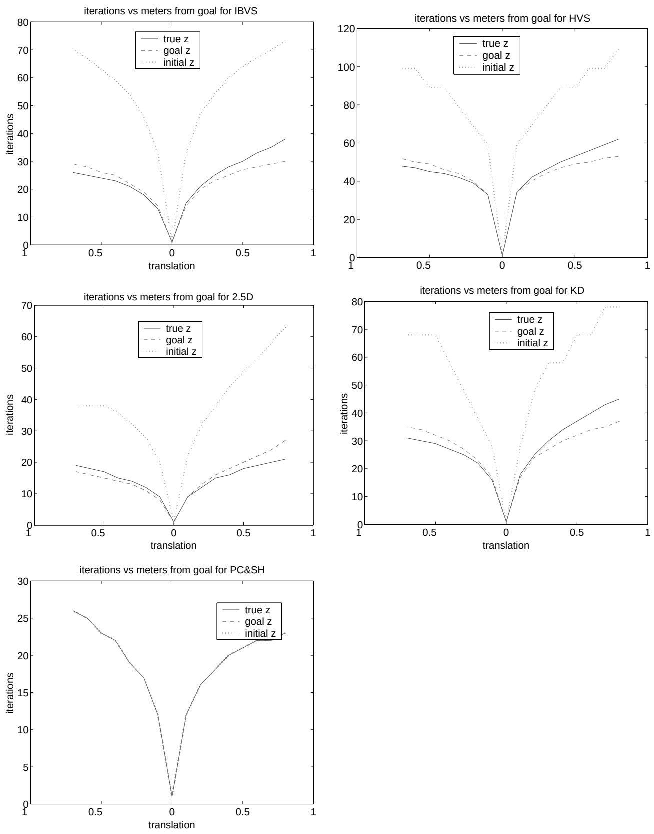

Fig. 23. Average iterations until convergence versus meters of *z*-axis translation with differing depth estimation.

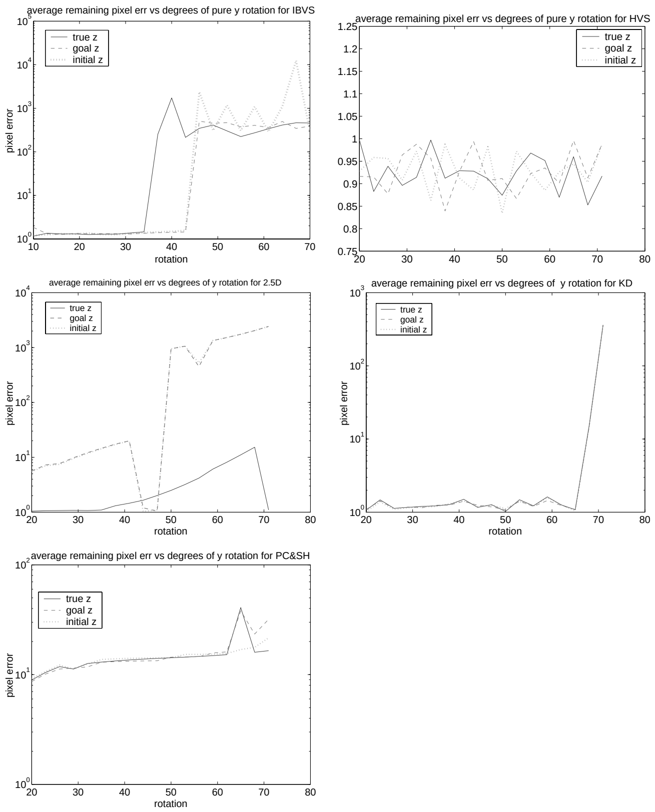

Fig. 24. Average remaining pixel error versus degrees of *y*-axis rotation with differing depth estimation.

Either constant depth estimate will provide roughly the same performance. HVS, KD and PC&SH performances do not appear to be much affected by the choice of depth estimation, although PC&SH is not able to reduce the error effectively for any rotation.

4.7.2. Number of Iterations Until Convergence or Failure

The number of iteration to convergence are shown in Figure 25. All methods of depth estimation for IBVS show a very rapid rise in iterations, quickly reaching the imposed ceiling of 300 iterations. IBVS using true depth has a sharp drop in iteration at about 35°, mirroring the increase in pixel error, whereas using a constant depth sees this drop in iterations about ten pixels later.

The $2\frac{1}{2}$D system has roughly similar performance for all depth estimation for rotation below 35°. After this point, constant depth estimation schemes take a brief ramp up then plummet to just a few iterations. True depth ramps up as well, although this increase begins at a greater rotation angle and continues longer, reaching a higher number of iterations.

HVS and KD are not at all affected by depth choice of estimation scheme; both show a very slowly increasing relationship with the rotation angle. PC&SH has a roughly linear increase in iterations with increasing rotation angle, although using a constant depth estimation method results in slightly fewer iterations for larger rotations.

4.7.3. Maximum Feature Point Excursion from the Principal Point

Figure 26 shows the maximum feature point excursion during this test. IBVS shows a constant level of feature point excursion up to 35° of rotation. After this point the system experiences large amounts of feature point excursion for all depth estimation schemes, although using the initial depth typically causes a greater amount of excursion than either true depth or goal depth.

The $2\frac{1}{2}$D system initially reveals a constant value for maximum feature point excursion for all methods of depth estimation. When using constant depth, however, there is an enormous spike, around 50° when using goal depth, and a much

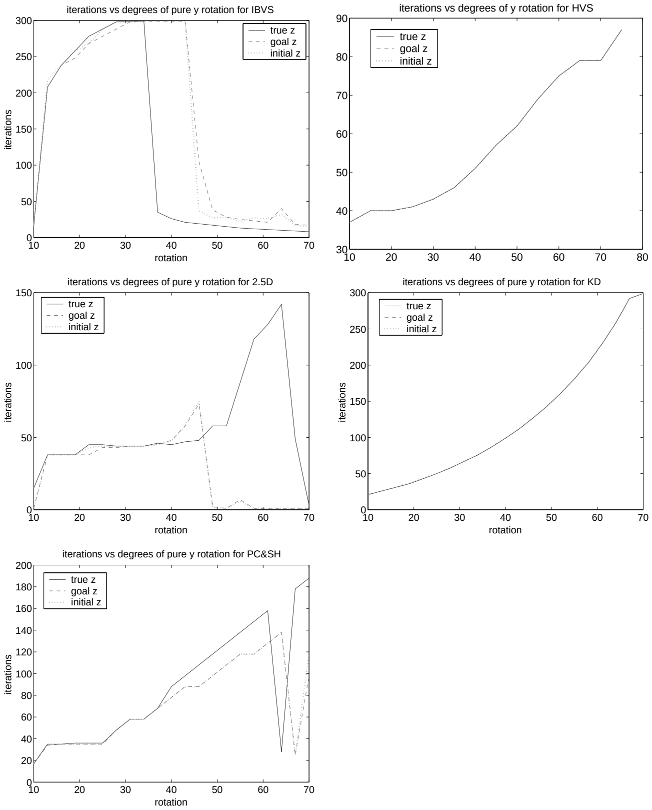

Fig. 25. Average iterations until convergence versus degrees of y-axis rotation with differing depth estimation.

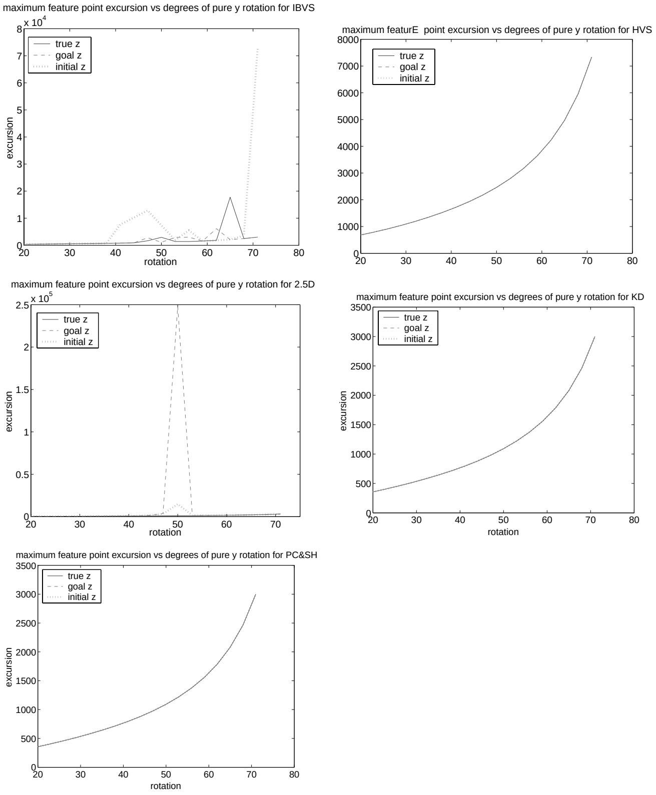

Fig. 26. Average maximum feature point excursion versus degrees of y-axis rotation with differing depth estimation.

more subtle spike when using the initial depth. After this spike the maximum excursion returns to its original, constant value.

Once again, HVS, KD and PC&SH show no variation in performance between depth estimation methods; all experience an exponential increase as rotation increases.

4.7.4. Maximum Camera Translation

The maximum camera translation is seen in Figure 27. IBVS follows a shallow linear increase for all depth estimation methods until about 35°, when the translation drastically increases and becomes extremely erratic for true depth. Using a constant depth allows the system to continue on the original path for an additional ten pixels or so before becoming large and erratic itself.

HVS exhibits more translation if a constant depth estimation is used, but the magnitude of this translation is so small as to be inconsequential.

The $2\frac{1}{2}$D system is indistinguishable between depth estimation for values up to around 43°, before which it follows a shallow increase with rotation angle. After this point, initial depth begins an exponential increase in translation. Goal depth undergoes a brief spike to high translation, after which it returns low and follows the same exponential increase as initial depth. When using true depth, $2\frac{1}{2}$D becomes irregular after 50°, but remains below 2 m.

KD initially follows an apparent parabolic dependence upon the rotation angle for each depth estimation method. The parabola for true depth has slightly higher values then a constant depth, but peaks at only about 0.475 m. At 35° the maximum translation immediately plunges to zero translation. The plots for both constant depth estimation schemes are nearly identical and, like true depth, also undergo a discontinuity and drop to zero, in their case at 45°.

Using initial depth or true depth for PC&SH results in identical maximum camera translation, a slow linear increase with rotation angle. Using goal depth results in the same plot, with the exception of a sharp spike up to 12 m translation between 45° and 50°.

4.7.5. Maximum Camera Rotation

Finally, for this test, we present the maximum camera rotation angle, seen in Figure 28. Each depth estimation method increases in a linear fashion, at approximately the same rate

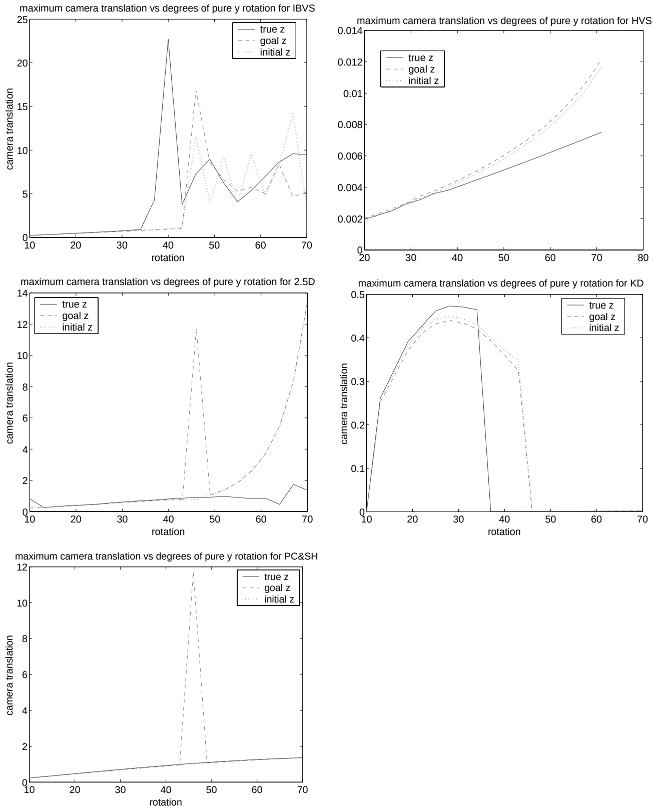

Fig. 27. Average maximum camera translation versus degrees of *y*-axis rotation with differing depth estimation.

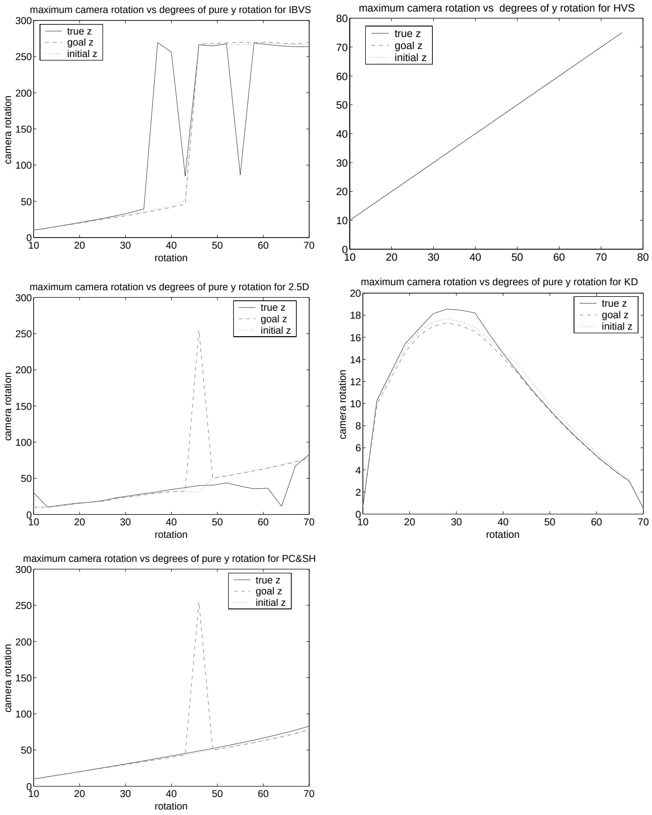

Fig. 28. Average maximum camera rotation versus degrees of *y*-axis rotation with differing depth estimation.

for IBVS for smaller rotations. At about 35° the maximum rotation becomes erratic, oscillating between very high and very low values. The constant depth estimation methods continue the linear increase until approximately 42° where they leap to a roughly constant value around 260°. HVS has no dependence upon the depth estimation, and maximum rotation is simply the initial pose.

Similar to its performance viewing maximum translation, $2\frac{1}{2}$D initially follows the same linear increase for all depth estimation methods. At 42° the system using goal depth takes a large spike upwards, while initial depth takes a small downward motion. Both constant depth methods then resume roughly the same trajectory they had earlier. About this time, using true depth begins to result in an erratic, although generally lower amount of rotation.

KD also has a maximum rotation that appears similar to its maximum translation. Each depth estimation scheme follows a parabolic path initially, switching to a linear decrease between 30° and 40°. PC&SH follows the same, roughly linear increase in rotation for all depth estimation schemes, with the exception of a spike in rotation when using the goal depth.

4.8. Simulation: Rotation of Feature Points with Differing Depth Estimation

Zeroing the image error during this test is generally difficult and requires motions along all degrees of freedom. The choice of depth estimation will affect the rate of convergence for translational motions. There is very little difference in IBVS for various choices of depth, and it generally performs quite well. The $2\frac{1}{2}$D system notably becomes unstable and fails regularly after 60° when a constant depth is used. Similarly, PC&SH becomes unstable after 70° when a constant depth is used. On the other hand, using a constant depth for KD extends its range of stable performance by about 5°.

4.8.1. Remaining Pixel Error

Remaining pixel errors are shown in Figure 29. IBVS and HVS show little deviation from around one pixel of error for all methods of depth estimation. The $2\frac{1}{2}$D system is able to zero the error for all rotations using absolute depth, but when using a constant depth estimation method results in a dramatic increase in remaining error after the feature points are rotated approximately 60°.

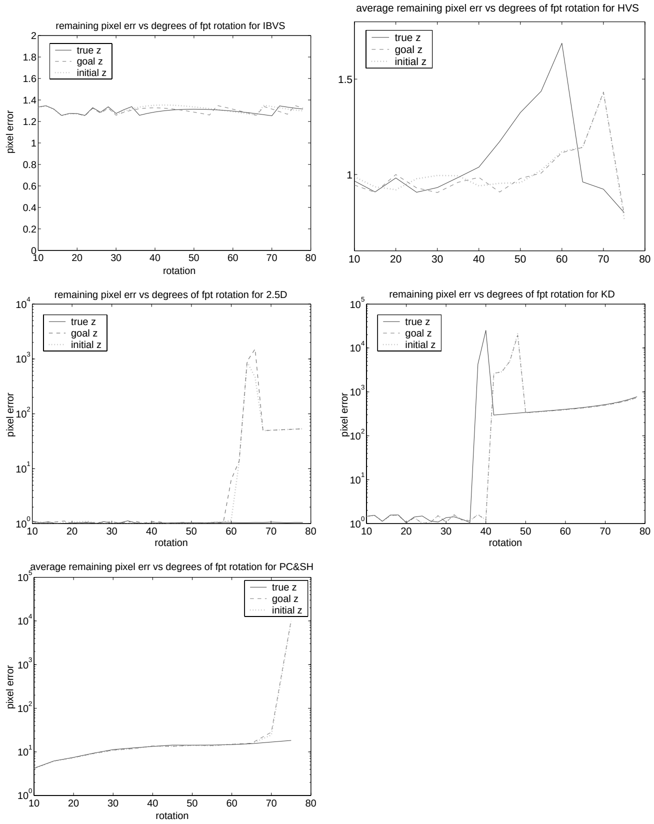

Fig. 29. Average remaining pixel error versus degrees of feature point rotation with differing depth estimation.

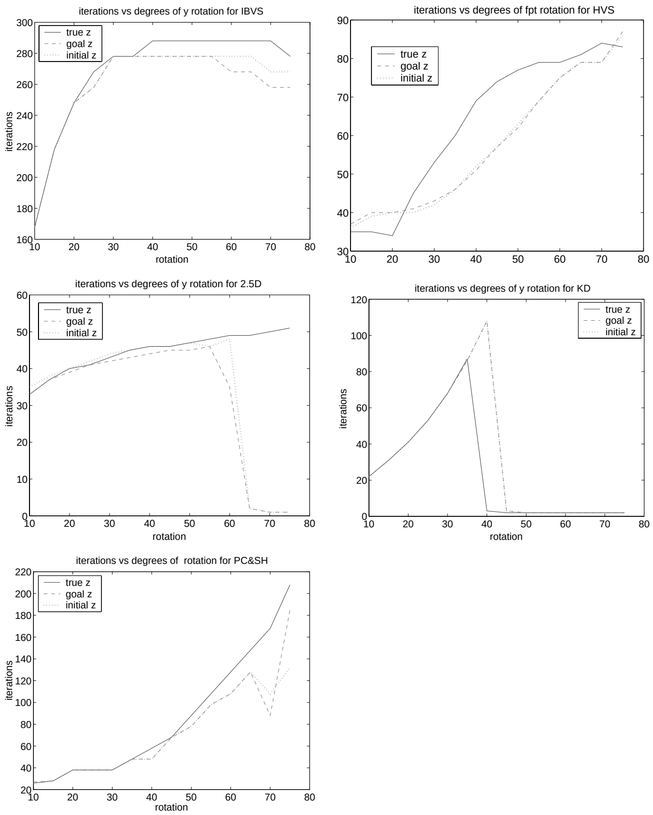

Fig. 30. Average iterations until convergence versus degrees of feature point rotation with differing depth estimation.

Similarly, PC&SH is able to reduce the error to around ten pixels for all rotations if true depth is used, but suffers a dramatic increase around 70° if either constant method is used to estimate depth. KD is able to reduce the error to around one pixel for rotations of the feature points under 35°. At this point, the true depth method immediately spikes, then settles to about 500 pixels of error. The constant depth methods continue to reduce the error to one until 40°, then they spike and settle to the same value as true depth. For each of these systems, failure is due to extremely large camera rotations during the early stages of visual servoing from which the systems are unable to recover within 300 iterations.

4.8.2. Number of Iterations Until Convergence or Failure

Graphs for the number of iterations until convergence are seen in Figure 30. IBVS quickly rises and then plateaus and eventually decreases for each depth estimation system. In general, the constant depth methods have a slightly lower number of iterations. HVS apparently requires more iterations using the constant depth estimation than either constant method. The $2\frac{1}{2}$D system initially shows all methods following a linear or extremely slow growing inverse exponential increase with rotations for each depth estimation method. Around 60° both constant depth estimation schemes plummet to very few iterations.

For KD, each system follows an identical exponential increase in iterations for lower rotations. After about 35°, the true depth method plunges to very few iterations, while the constant depth methods continue to increase for another 5° of rotation before they too drop. IBVS experiences an erratic, but roughly slow growing exponential increase in iterations as the amount the feature points are rotated increases. Generally, the number of iterations is lower when a constant depth estimation is used.

4.8.3. Maximum Feature Point Excursion from the Principal Point

The maximum feature point excursions are plotted in Figure 31. All depth estimation methods for IBVS result in an exponential increase with increasing rotation of the feature point plane. The two constant depth estimation methods experience a larger amount of excursion than using true depth.

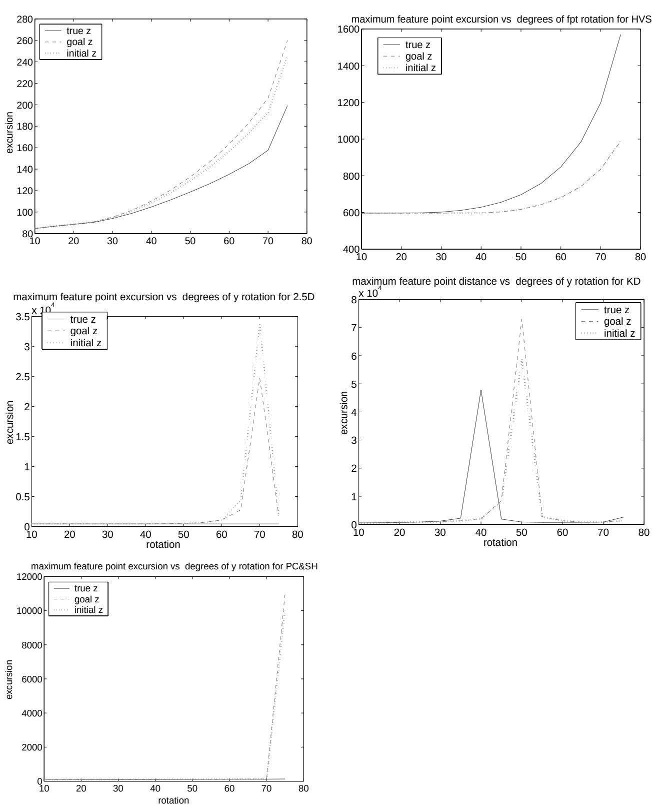

Fig. 31. Maximum feature point excursion versus degrees of feature point rotation with differing depth estimation.

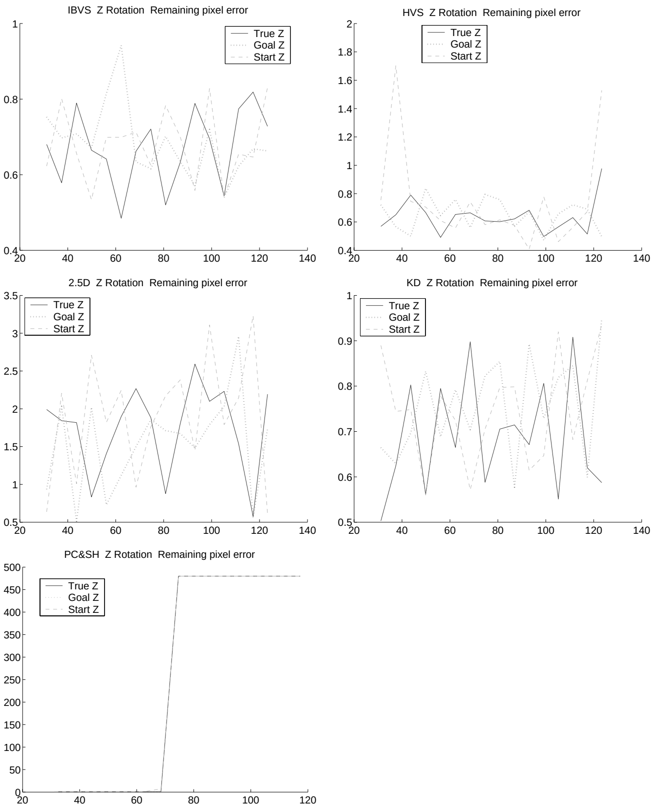

Fig. 32. Experiment: remaining pixel error versus degrees of z-axis rotation with differing depth estimation.

On the other hand, $2\frac{1}{2}$D has a constant value when true depth is used, but using either constant depth estimation scheme results in a large spike at 70°. KD also mostly holds to a constant value, but experiences a spike for true depth at 40° and both constant depth methods at 50°. Finally, PC&SH maintains a constant maximum feature point excursion for all rotations if true depth is used, but experiences a spike for both constant methods at 80°.

4.9. Experimental Results: Rotation About the Optical Axis with Differing Depth Estimation

The camera was rotated about the optical axis over a range of values. Due to kinematic limits of the robot wrist, 120° was the largest rotation attainable, so the range is from 30° to 120° in degree increments. Perhaps the strongest conclusion we can draw is that the effects of noise have a much greater effect on performance than depth estimation method for this task.

4.9.1. Remaining Pixel Error

No system shows a strong dependence on the method of depth estimation. IBVS and KD are able to consistently zero the feature point error. HVS and $2\frac{1}{2}$D are slightly less consistent, although they both reduce the error to within two pixels.

PC&SH failed due to losing the feature points after 70°. For initial offsets greater than 30° PC&SH tended to have large, erroneous rotations and translations. These motions always occur during the first one or two iterations. Returning to eq. (17), we see that errors in the signal can generate an erroneous Jacobian and lead to large, incorrect vector $\mathbf{J}_z\dot{\mathbf{r}}_z$ being subtracted from the feature point error. KD suffered from similar effects, but the results were not as extreme.

4.9.2. Number of Iterations Until Convergence or Failure

Again there is no evident dependence upon the method of depth estimation. IBVS shows a dependence upon the amount of translation, while HVS, $2\frac{1}{2}$D, and KD do not. $2\frac{1}{2}$D requires the largest number of iterations. During visual servoing $2\frac{1}{2}$D converged quickly to around two pixels of error, but had a great deal of trouble reducing the error further, generally taking several hundred iterations to converge to a steady error. PC&SH was failed after 70° of rotation, and as seen here this failure happened very quickly, within just a few iterations.

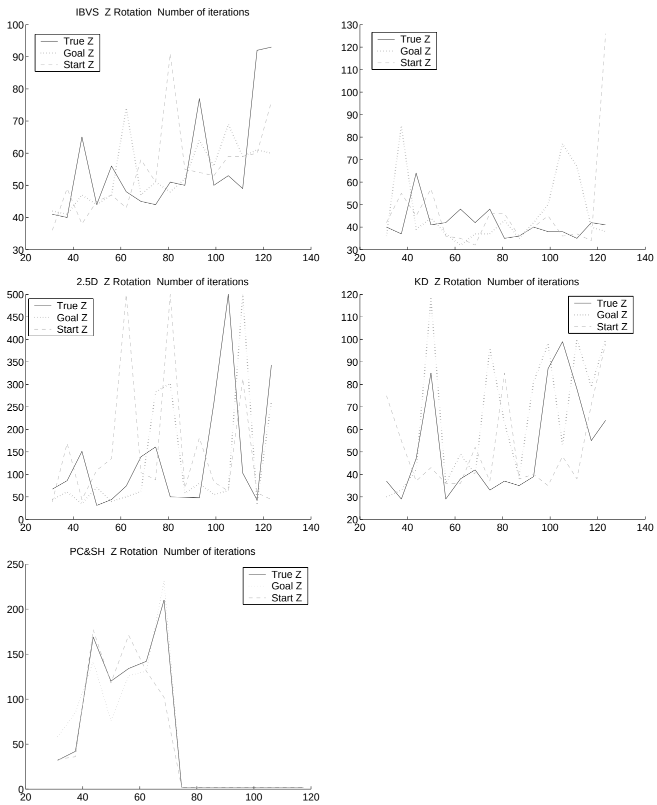

Fig. 33. Experiment: iterations until convergence versus degrees of z-axis rotation with differing depth estimation.

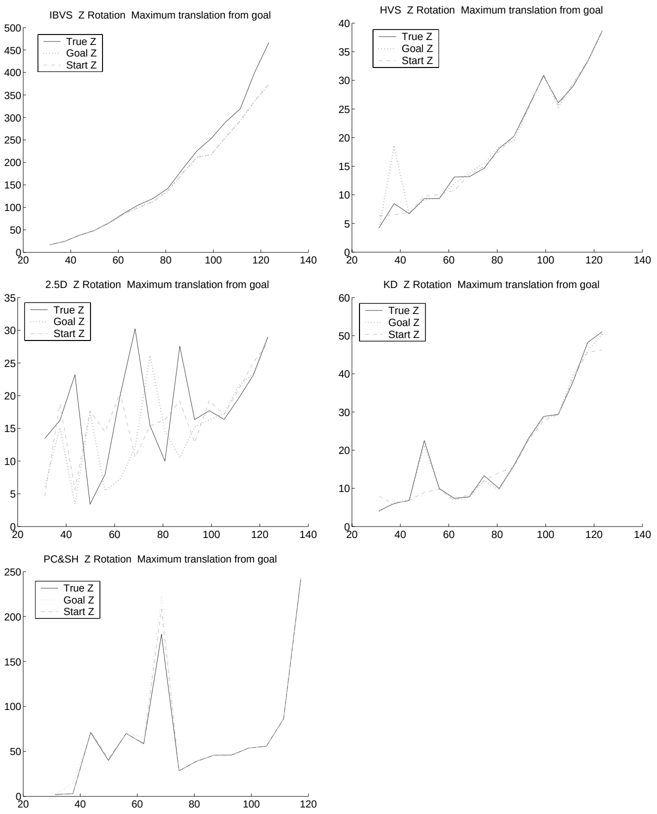

Fig. 34. Experiment: maximum camera translation versus degrees of z-axis rotation with differing depth estimation.

4.9.3. Maximum Camera Translation

Maximum translation for IBVS has a strong correlation with the amount of rotation, as the system experiences camera retreat, moving up to a half meter away from the image plane at 120°. This is close to the kinematic limits of the robot, but IBVS always successfully completed the error. As in the simulations, the system retreats less using a constant depth estimation method.

HVS, $2\frac{1}{2}$D KD and PC&SH show no noticeable dependence upon the depth estimation method. These three systems increase in translation as the rotation increases. For $2\frac{1}{2}$D and PBVS this is generally due to errors in the calibration of the camera offset from the end-effector, and is only a few centimeters. KD and PC&SH suffers from the erroneous motions described above, with PC&SH being affected far worse.

4.9.4. Maximum Feature Point Excursion

As in the simulations, IBVS has the least amount of feature point excursion, although there is noticeable excursion since the effects of calibration errors of the camera offset from the end-effector become strong for large rotations. HVS, $2\frac{1}{2}$D and KD surrender control of the feature point position, so have larger feature point excursion. KD is the worst of these three due to the incorrect motions described earlier. PC&SH fails due to feature point loss after 70°.

4.10. Experimental Results: Translation Along the Optical Axis with Differing Depth Estimation

The systems were tested for a range of translation from −10 cm (away from the image) to 10 cm (towards the image). Maximum translation from goal and maximum feature point excursion from image center were practically identical for all cases and all depth estimation methods. The maximum translation was the initial offset the feature point excursion was the initial image for positive optical axis translations (towards the feature point plane) or the goal position for negative translation.

4.10.1. Remaining Pixel Error

IBVS, HVS and KD are all able to reduce the error below one pixel with no discernible dependence on the depth estimation

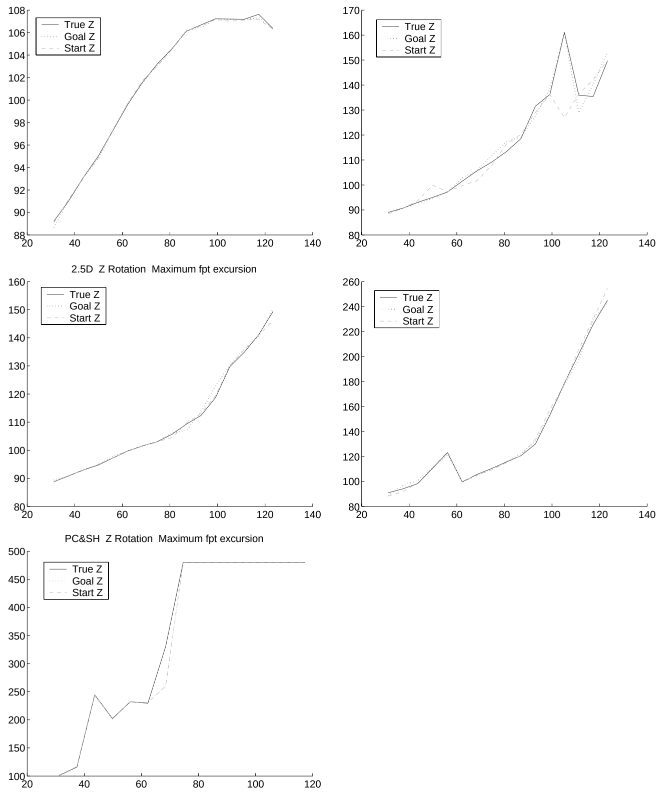

Fig. 35. Experiment: maximum feature point excursion versus degrees of z-axis rotation with differing depth estimation.

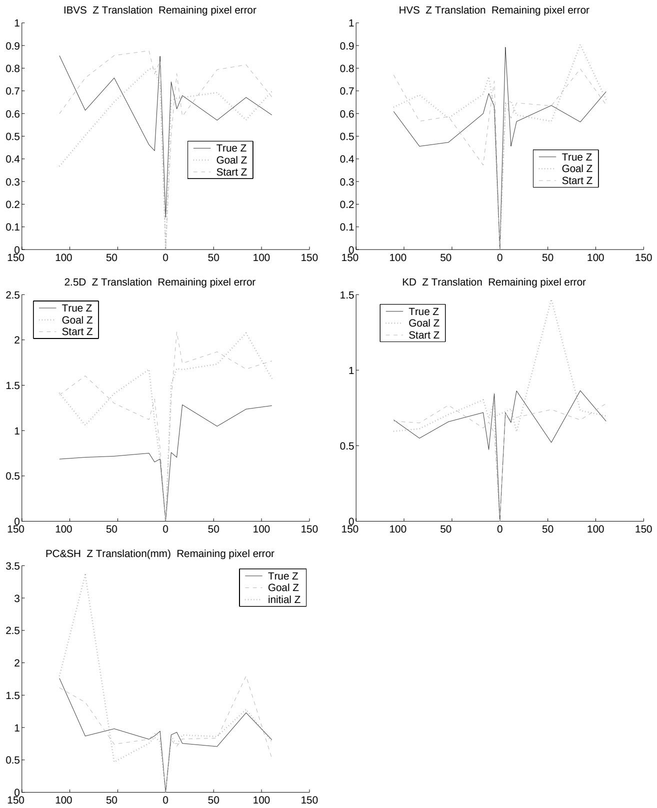

Fig. 36. Experiment: remaining pixel error versus meters of z-axis translation with differing depth estimation.

used. $2\frac{1}{2}$D has minor trouble zeroing the error when a constant depth estimate is used. It converges at a rate on a par with true depth, but once within two pixels it cannot reduce much further and reaches a steady error less than two pixels. PC&SH shows mixed results, able to zero errors for motions starting below 5 cm, but occasionally only reaching a steady error for larger motions. The reason for this remaining error is seen in the maximum camera rotation.

4.10.2. Number of Iterations Until Convergence or Failure

IBVS, HVS and KD show similar results, again with no clear dependence on the depth estimation. All three appear to require more iterations to zero motions towards the feature points. Due to the fact that they sometimes could not zero the error, but rather reduced it to low steady amount, both $2\frac{1}{2}$D and PC&SH show a larger number of iterations. It is not well reflected in this graph, but $2\frac{1}{2}$D does converge well, but has trouble reducing below one pixel.

IBVS, HVS, $2\frac{1}{2}$D and KD all show very little camera rotation, as expected for strict translation. There is no clear dependence on depth estimation except for $2\frac{1}{2}$D, which is more consistent and exhibits slightly less rotation using true depth.

Similar to the results for optical axis rotation, PC&SH undergoes large rotations when the initial pose is far away on the optical axis, particularly when the initial pose is farther from the image. These rotations always occur during the first one or two iterations when the induced translation has a large component along the z-axis.

4.11. Experimental Results: Rotation of Feature Points with Differing Depth Estimation

In order to preserve consistency during experiments and to have a means to determine distance to the goal position we could not simply rotate the feature point plane for each experiment. Thus, the experiments for rotation of the feature point plane involved finding a system of initial poses for the camera that were equivalent to rotating the feature point plane. To accomplish this the camera was put in its normal initial position, the feature point plane was rotated the desired amount, and the system was allowed to visual servo until the error was zeroed. This position was then recorded. During the experiments the camera would capture a goal image as usual, then

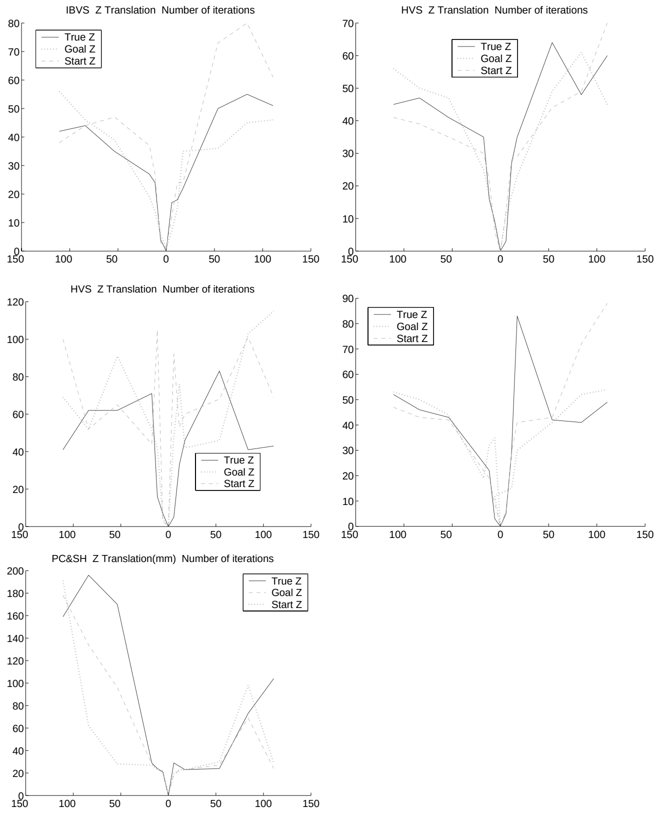

Fig. 37. Experiment: iterations until convergence versus meters of z-axis translation with differing depth estimation.

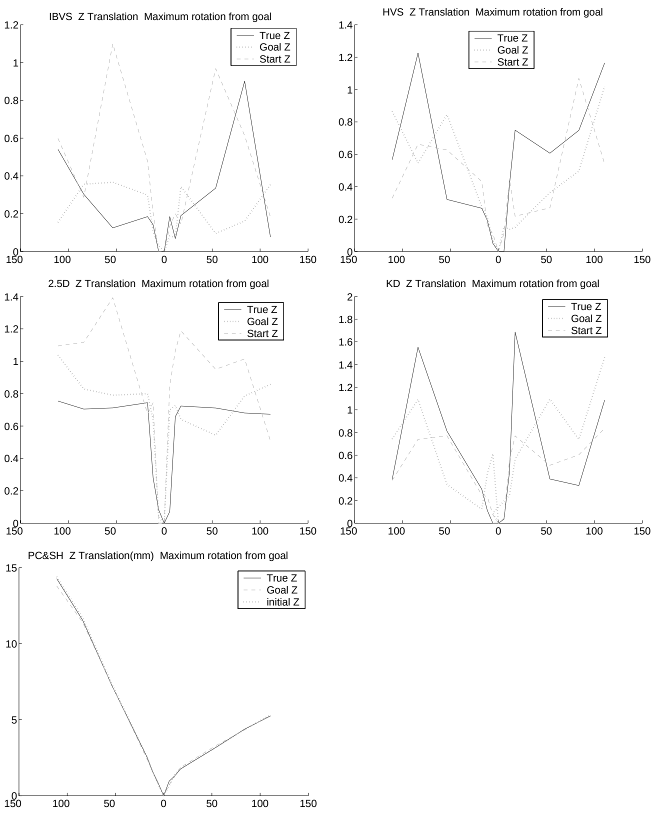

Fig. 38. Experiment: maximum camera rotation versus degrees of z-axis translation with differing depth estimation.

be moved to the appropriate stored position and return to its initial position via visual servoing.

Rotations of the feature point plane of 40° required the robot to move very close to its joint limits, and 45° proved impossible to achieve. Since they were so close to the joint limits at 40°, some systems, such as IBVS, that tend to have erroneous motions at start, would quickly hit these joint limits.

4.11.1. Remaining Pixel Error

All systems have more trouble reducing the error below one pixel for this task than they did for the previous systems. HVS and $2\frac{1}{2}$D are able to successfully reduce the error for the whole range of motions. IBVS, KD and Pc&SH are successful up to 30° of feature point rotation. After this, IBVS and PC&SH hit the joint limits as mentioned earlier. KD hits the joint limits at 35°, and if using a constant depth estimation method loses a feature point at 40°. IBVS may reveal a higher pixel error if constant depth estimation methods are used.

4.11.2. Number of Iterations Until Convergence or Failure

Once again there is little discernible dependence upon the depth estimation method used. When successful, IBVS, HVS, KD and PC&SH perform similarly, requiring between 150 and 200 iterations to converge to zero the error. $2\frac{1}{2}$D often requires more iterations; once again it has some trouble reducing the error below two pixels although it converges on a par with the other systems. When IBVS, KD and PC&SH fail they do so quickly, usually within ten iterations.

5. Conclusions

Visual servoing, and robotics in general, is a constantly evolving field. As innovations continue to be made, it becomes increasingly important to explore the different methods in order to gain insight into the characteristics, strengths, and weaknesses of each. Focusing on the field of partitioned image-based visual servo systems, we have performed several standardized tests of robustness in the face of imaging error and system performance against difficult tasks. These data can be used to select appropriate visual servo systems for specific tasks and conditions or to provide direction for future research.

We were able to determine several key characteristics of each system. IBVS has noted difficulty for severe rotations

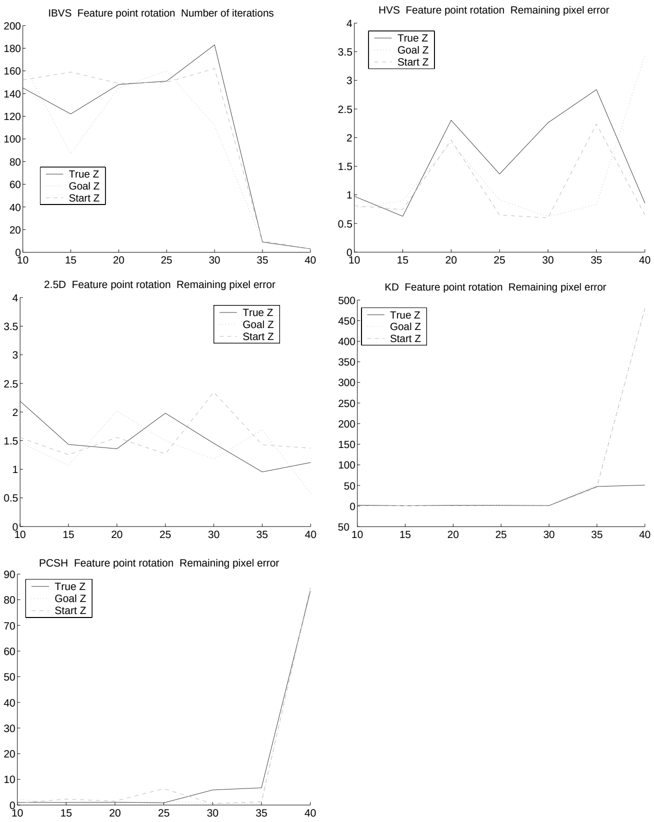

Fig. 39. Experiment: remaining pixel error versus degrees of feature point rotation with differing depth estimation.

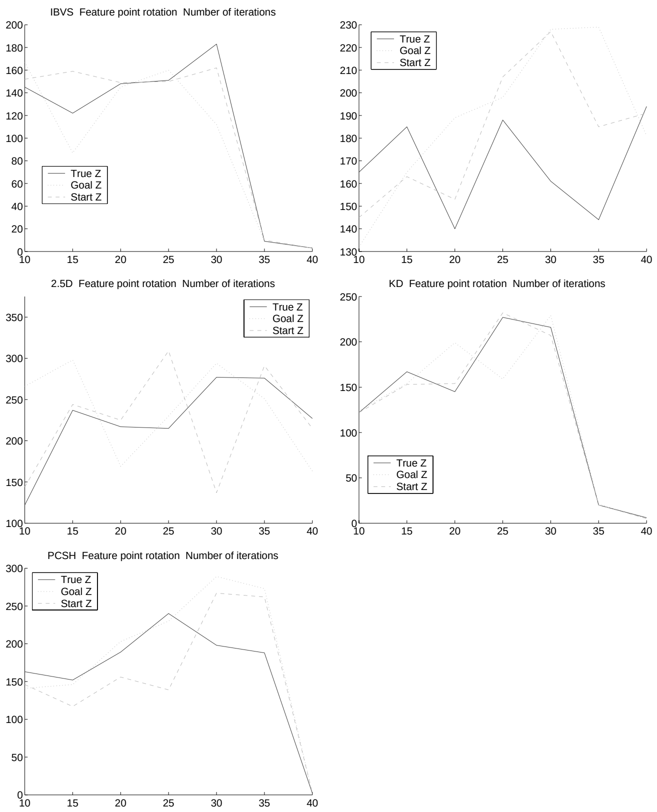

Fig. 40. Experiment: iterations until convergence versus degrees of feature point rotation with differing depth estimation.

about the optical axis due to increasing camera retreat as the angle of rotation increases. IBVS also has difficulty zeroing the error for extremely large general motions (those including rotations and translation about an arbitrary axis), and in all cases tends to have a larger amount of camera translation than the other systems. It performs very well in the face of noise and for smaller general motions.

The $2\frac{1}{2}$D and KD systems perform well, but the performance of both is degraded by noise to a greater extent than either IBVS or PC&SH. This is due to the susceptibility of the homography matrix computations to noise. The performance of KD suffers the most from the effects of signal noise since the translational component of the homography matrix is affected by noise more than the rotational component. Neither system experiences the camera retreat that afflicted IBVS.

PC&SH was designed to address the phenomenon of camera retreat that IBVS experiences during rotations about the optical axis. By decoupling the optical axis motions, camera retreat is prevented. PC&SH avoids the use of the homography matrix, and so performs well in the face of signal noise. However, the z-axis rotations and translation are dependent upon the relative positions of the feature points. Thus, it performs poorly when rotations about the x-axis and/or y-axis alter the relative positions of feature points.

The method of depth estimation does not generally have much effect on system performance. In general, the choice of depth affects only the magnitude of the translation. Both KD and $2\frac{1}{2}$D decouple the rotation and translation motions, so the depth estimation truly appears as a translation gain coefficient. In the case of IBVS, camera retreat is reduced and system stability improved with the use of a constant for depth estimation, at the expense of a slower rate of convergence. The $2\frac{1}{2}$D system, on the other hand, experienced a slightly reduced stability when a constant depth was used and faced with large rotations about an arbitrary axis. PC&SH and KD experienced little difference, good or ill, from different depth estimation methods.

References

Chaumette, F. 1998. Potential problems of stability and convergence in image-based and position-based visual servoing. In D. Kriegman, G. Hager, and S. Morse, eds., *The Confluence of Vision and Control, Lecture Notes in Control and Information Sciences*, Vol. 237, Springer-Verlag, Berlin, pp. 66–78.

Corke, P. 1993. Visual control of robot manipulators—A review. In K. Hashimoto, ed., *Visual Servoing, Robotics and Automated Systems*, Vol. 7. World Scientific, Singapore, pp. 1–31.

Corke, P. I. 1996. Robotics toolbox for MATLAB. *IEEE Robotics and Automation Magazine* 3(1):24–32.

Corke, P. I., and Hutchinson, S. A. 1999. A new partitioned approach to image-based visual servo control. In *Proceedings of the 31st International Symposium on Robotics and Automation*.

Corke, P. I., and Hutchinson, S. A. 2000. A new partitioned approach to image-based visual servo control. In *Proceedings of the 39th Conference on Decision and Control*, December, pp. 2521–2526.

Deguchi, K. 1998. Optimal motion control for image-based visual servoing by decoupling translation and rotation. In *Proceedings of the International Conference on Intelligent Robots and Systems*, October, pp. 705–711.

DeMenthon, D., and Davis, L. S. 1992. Model-based object pose in 25 lines of code. In *European Conference on Computer Vision*, pp. 335–343.

Espiau, B., Chaumette, F., and Rives, P. 1992. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation* 8:313–326.

Faugeras, O., and Lustman, F. 1988. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence* 2(3):485–508.

Feddema, J., and Mitchell, O. 1989. Vision-guided servoing with feature-based trajectory generation. *IEEE Transactions on Robotics and Automation* 5:691–700.

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

Lenz, R., and Tsai, R. 1989. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation* 5(3):345–358.

Malis, E., Chaumette, F., and Boudet, S. 1999. 2 1/2 D visual servoing. *IEEE Transactions on Robotics and Automation* 15:238–250.

Martinet, P. 1999. Comparison of visual servoing techniques: Experimental results. In *Proceedings of the European Control Conference, ECC'99*, August.

Martinet, P., Gallice, J., and Khadraoui, D. 1996. Vision based control law using 3d visual features. In *Proceedings of WAC '96*, May, Montpellier, France, Vol. 3, pp. 497–502.

Sanderson, A. C., and Weiss, L. E. 1980. Image-based visual servo control using relational graph error signals. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1074–1077.

Shirai, Y., and Inoue, H. 1973. Guiding a robot by visual feedback in assembling tasks. *Pattern Recognition* 5:99–108.

Tsai, R. 1987. A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation* 3(4):323–344.

Weiss, L. E., Sanderson, A. C., and Neuman, C. P. 1987. Dynamic sensor-based control of robots with visual feedback. *IEEE Journal of Robotics and Automation* 3:404–417.

Zhang, Z., and Hanson, A. 1996. 3D reconstruction based on homography mapping. 1996. In *ARPA Image Understanding Workshop*, Palm Springs, CA.