



Multi-Attribute Utility Analysis in the Choice of Vision-Based Robot Controllers

Nicholas Gans & Seth Hutchinson

To cite this article: Nicholas Gans & Seth Hutchinson (2008) Multi-Attribute Utility Analysis in the Choice of Vision-Based Robot Controllers, International Journal of Optomechatronics, 2:3, 326-360

To link to this article: <https://doi.org/10.1080/15599610802303678>



Published online: 14 Jul 2010.



Submit your article to this journal [↗](#)



Article views: 73

MULTI-ATTRIBUTE UTILITY ANALYSIS IN THE CHOICE OF VISION-BASED ROBOT CONTROLLERS

Nicholas Gans¹ and Seth Hutchinson²

¹*Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, Florida, USA*

²*Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, Illinois, USA*

Multi-attribute utility analysis is an ideal tool for comparing the disparate performance of multiple visual servo controllers. Its strength lies in the fact that very different metrics can be compared and that it takes into account human preferences and risk attitudes. In this article, multi-attribute utility analysis is used to choose between multiple visual servo controllers and choices of camera lenses. The resulting visual servo controllers are suited to the needs of a specific user for specific tasks.

1. INTRODUCTION

Robots and other automated systems are used in tasks varying from surgery to industrial assembly to house cleaning. By performing tasks that are difficult, hazardous, or tedious robots can improve human safety and health, as well as improve the quality of consumer goods while reducing costs and prices.

Robots are designed to serve the needs of a human user or users, so design should meet the performance needs of specific human users. Multi-attribute utility analysis (MAUA) (Merkhofer and Keeney 1987; Keeney and Raiffa 1993) is a method used by systems engineers to directly gauge a human's preferences and attitude toward risk. It provides a way to equate distinct system attributes, including those that do not have a natural metric.

Additionally, MAUA can account for human preferences and attitudes toward risk. For example, different applications can have different acceptable failure rates, and different people using the same system may be willing to accept different failure rates as well. MAUA can tailor system performance to specific needs.

In a design application, MAUA is performed offline during the design phase to select among competing designs, components, gains, etc., using metrics such as failure rate, cost, speed, and even subjective measurements such as user satisfaction. MAUA is traditionally used to evaluate economic impacts of design choices (Thurston 1990). It can be used in technical design of a system (Diller and Waemkessel 2001), but appears to be a little-known tool in the design of control systems, robotics, and visual servoing.

Address correspondence to Nicholas Gans, Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32608, USA. E-mail: ngans@ufl.edu

NOMENCLATURE

$\mathbf{X}(t), \mathbf{X}^*$	current and goal pose in $SO(3)$	\mathbf{s}	image feature vector
$\mathbf{u}\theta$	axis/angle representation of rotation	\mathbf{e}_i	image feature error
\mathbf{e}_p	camera pose error	\mathbf{L}_i	IBVS interaction matrix
$\dot{\xi}$	camera velocity	$U_i(x_i)$	single-attribute utility function
\mathbf{L}_p	PBVS interaction matrix	$U(x_1, x_2, \dots, x_n)$	multi-attribute utility function

Visual servo control is the use of image data in the closed-loop position control of a robot end-effector. There are two basic approaches to visual servo control: image-based visual servoing (IBVS) and position-based visual servoing (PBVS). In IBVS, an error signal is measured in the image and regulated to zero. The derivative of the error signal is mapped to actuator commands required to reduce the image error. In PBVS systems, features are detected in an image, and used to generate a 3-D model of the environment. The error is then computed in the Cartesian task space. There are extensive resources detailing these methods (Weiss et al. 1987; Feddema and Mitchell 1989; Espiau et al. 1992; Martinet et al. 1996; Hutchinson et al. 1996; Chaumette and Hutchinson 2006a).

It is well known that both methods have specific strengths and shortcomings (Chaumette 1998). IBVS methods control the image but do not explicitly control the robot pose. Therefore, features are well regulated and unlikely to leave the field of view but the robot may make very large, possibly damaging, motions. PBVS methods control the pose, and the robot will move along least distance paths to the goal pose. However, features may leave the field of view, resulting in a system failure. It can be seen that IBVS and PBVS are complimentary in their strengths and weaknesses.

Numerous approaches have been presented that combine aspects of IBVS and PBVS (Deguchi 1998; Malis et al. 1999; Fang et al. 2005; Chaumette and Hutchinson 2006b; Hafez et al. 2007). The authors have previously proposed several switched-system approaches to visual servoing (Gans and Hutchinson 2002; Gans and Hutchinson 2003; Gans and Hutchinson 2007). A random switching rule can be applied where the system randomly selects between using IBVS or PBVS at each iteration of the control loop or at specific time intervals. As time, and the number of switches, increases, performance is expected to be influenced by both IBVS and PBVS. Experiments show that the resulting system performs reasonably in regulating both the image and position errors, rather than extremely well in one and poorly in the other.

The switched system in Gans and Hutchinson (2003) used a binary switching rule. At each iteration, the choice to use IBVS or PBVS was made by a binary random process with equal probability of 0.5 of choosing either one. However, this may not be the best switching rule for all configurations or all uses. A system that is biased to select IBVS will provide better control of the image features, while a PBVS biased system will better control the position. A robot with a short reach may require strict control of pose, while a camera with a narrow field of view will require strict control of image features. Additionally, failure of a visual control system can vary in severity. Failure in an assembly task may cause a small increase in scrap rate, but failure could be fatal in a medical task.

For these reasons, users selecting a binary coefficient can have very different performance goals and very different attitudes toward risk. The field of decision theory and MAUA provides a means of selecting a binary coefficient. MAUA provides an ideal method of gaging a user's performance needs, preferences, and attitude toward risk (Keeney, and Raiffa 1993), and this information can be directly applied to select a switching rule.

The internal parameters of a digital camera, including focal length, projection center, and pixel dimensions will affect the performance of a visual servo system. A lens with a long focal length will have a telescoping or zooming affect. This will improve precision of a visual servo controller but decrease the camera field of view and increase the risk of features leaving the field of view. A lens with a short focal length will have a wide field of view, but will be less precise as the image will not be as finely resolved. Selection of a lens for a particular system requires balancing these tradeoffs. MAUA provides a method to select a lens that will reflect a users preferences and attitudes toward risk.

In this article, we present two experiments in the use of MAUA to design vision-based control systems. The first experiment selects a binary switching rule for a switched visual servo system. This experiment is intended to give a clear and thorough introduction to MAUA. A decision maker's preferences and risk attitudes toward position control, feature control, and failure rate are used to create a three-attribute utility function. This function is then used to rate the utility of several switched systems, each using a different probability in the binary random switching rule.

The second experiment involves selecting between IBVS and PBVS, along with choosing a lens. The task under consideration is an unmanned air vehicle (UAV) refueling task, with attributes reflecting each system's ability to correctly dock with a fuel drogue, to keep a fuel tanker plane and drogue in the camera field of view, and the time needed to dock.

The article proceeds as follows. Section 2 discusses image-based and position-based visual servoing and introduce the switched system. Section 3 gives the background of MAUA while detailing an experiment into the use of MAUA to choose a switched-system visual servoing system suited to the needs of a specific user. Section 3 details the use of MAUA in the selection of a control system and camera optics for a UAV refueling/docking task.

2. VISUAL SERVOING

2.1. Position-Based Visual Servoing

The task in PBVS is to regulate the error between the current camera pose and the goal pose. Given a current camera pose $\mathbf{X}(t)$ and goal pose \mathbf{X}^* (throughout, the superscript $*$ denotes values at the goal configuration), the transformation relating them is described by a translation $\mathbf{d} \in \mathbb{R}^3$ and rotation of the camera frame $\mathbf{R} \in SO(3)$.

Locally, $SO(3)$ can be parameterized by the three-tuple $\mathbf{u}\theta$, in which θ is an angle of rotation about the axis defined by the unit vector \mathbf{u} . Given a collection of feature points in the image, there are numerous methods to extract $\mathbf{X}(t)$ and thus

\mathbf{d} and $\mathbf{u}\theta$ from $\mathbf{X}(t)$ (Longuet-Higgins 1981; Huang and Faugeras 1989; Faugeras and Lustman 1988; Zhang and Hanson 1996; DeMenthon and Davis 1992).

For a PBVS system, the error \mathbf{e}_p is defined in terms of the rigid body motion that relates \mathbf{X} to \mathbf{X}^*

$$\mathbf{e}_p = \begin{bmatrix} \mathbf{d} \\ \mathbf{u}\theta \end{bmatrix}. \quad (1)$$

If the camera is moving with velocity $\xi = (\mathbf{v}^T, \omega^T)^T$ (in twist coordinates) the relationship between the error derivative and the camera velocity is given by

$$\dot{\mathbf{e}}_p = \begin{bmatrix} \mathbf{R} & 0 \\ 0 & \mathbf{L}_\omega \mathbf{R} \end{bmatrix} \xi = \mathbf{L}_p \xi \quad (2)$$

in which (Malis et al. 1999)

$$\mathbf{L}_\omega(\mathbf{u}, \theta) = I - \frac{\theta}{2} \mathbf{u}_\times + \left(1 - \frac{\text{sinc}\theta}{\text{sinc}^2 \frac{\theta}{2}} \right) \mathbf{u}_\times^2 \quad (3)$$

and $\mathbf{u}_\times \in \mathbb{R}^{3 \times 3}$ is the skew symmetric matrix associated with \mathbf{u} . Note that by definition, $\text{sinc}(0) = 1$.

Since \mathbf{L}_ω is nonsingular when $\theta \neq k\pi$, $k \in \mathbb{Z} \setminus \{0\}$ (Malis et al. 1999), camera velocity defined by the feedback control law

$$\xi = -\lambda_p \mathbf{L}_p^{-1} \mathbf{e}_p \quad (4)$$

gives the closed-loop error derivative

$$\dot{\mathbf{e}}_p = -\lambda_p \mathbf{e}_p \quad (5)$$

in which λ_p is a positive gain scalar. See Deng et al. (2002) and Chaumette and Hutchinson (2006a) for detailed proofs of stability and robustness.

While the pose error tends monotonically to zero, there is no control over the position of the image points. If there is any rotation present, the feature points will move along curves as the camera undergoes rotation and translation, this is seen in Figures 1–3. Figure 1 shows the trajectories of four feature points as the camera moves from its initial pose to the goal pose. The points marked by a O are the features seen when the camera is at the initial pose and the points marked with a * are as seen when the camera is at its goal pose. A large curved path is traced out in the image and the features are close to leaving the image. Figure 2 shows the camera pose error vector over time; all pose error elements converge to zero. Figure 3 is the feature point error in pixels for the four points over time. The pixel errors initially increase before eventually converging to zero.

The limited imaging surface of a camera makes it possible for the feature points to leave the image. In this case the system may no longer be able to reconstruct the pose error estimate, and the task cannot be completed. For the purposes of this article, failure of a visual servo system is defined to be any situation in which it does not successfully zero the error.

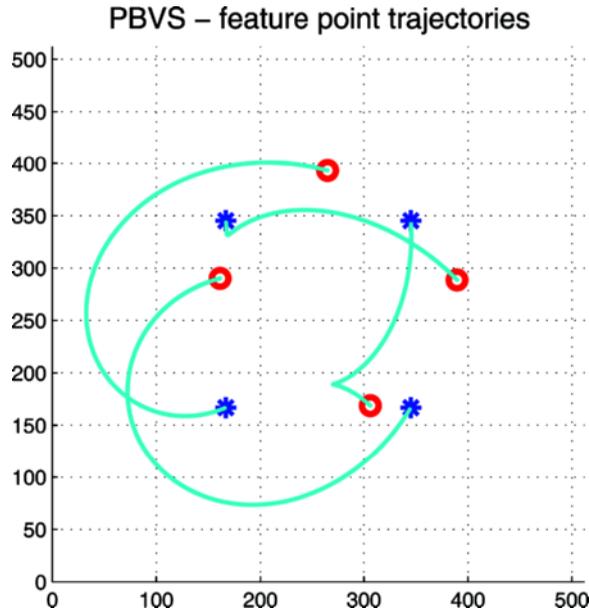


Figure 1. Example of typical trajectories for feature points in PBVS.

2.2. Image-Based Visual Servoing

In image-based visual servo control, the control law is a function of an error that is measured in the image. If $\mathbf{s}(t)$ denotes the vector of image features that are extracted from computer vision data, and \mathbf{s}^* is the feature vector in a goal image,

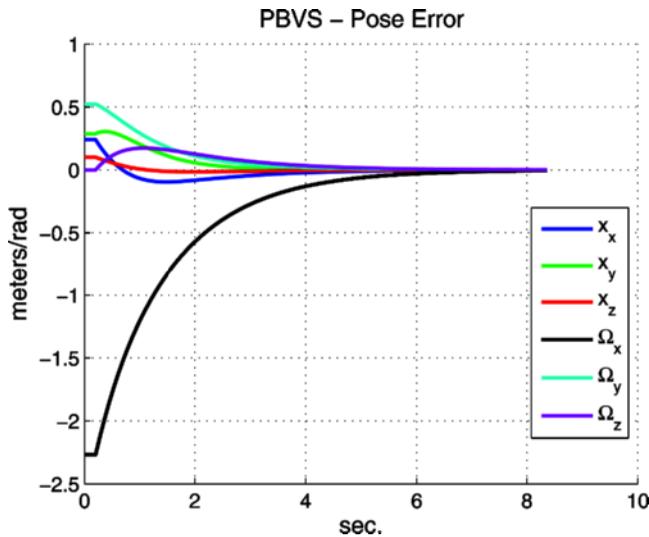


Figure 2. Example of typical camera pose error in PBVS.

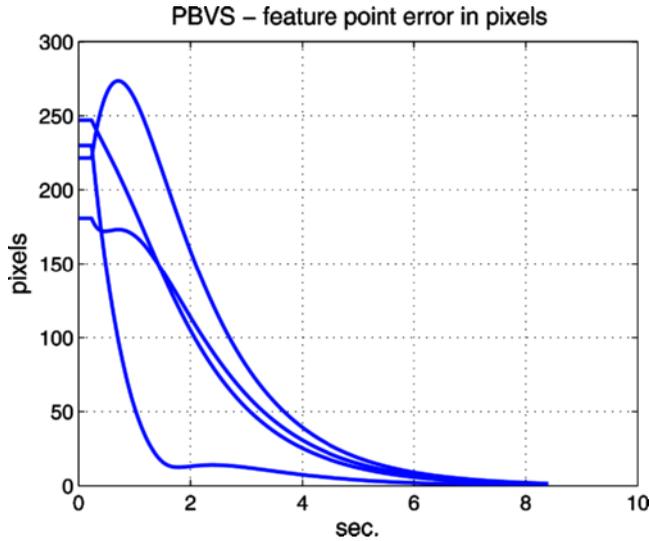


Figure 3. Example of typical feature point error in PBVS.

the error is defined in the image feature space,

$$\mathbf{e}_i(t) = \mathbf{s}(t) - \mathbf{s}^* \quad (6)$$

The relationship between camera velocity and the measured feature values is given by

$$\dot{\mathbf{e}}_i = \dot{\mathbf{s}} = \mathbf{L}_i \dot{\boldsymbol{\zeta}} \quad (7)$$

in which \mathbf{L}_i is the image Jacobian (also called the interaction matrix) (Weiss et al. 1987; Feddema and Mitchell 1989; Espiau et al. 1992; Hutchinson et al. 1996).

The most common choice of image features is image points. In this case the feature vector is a concatenation of n point coordinates $\mathbf{p}_j = [x_j \ y_j]^T \in \mathbb{R}^2, j \in 1 \dots n$, such that

$$\mathbf{s} = [\mathbf{p}_1^T \dots \mathbf{p}_n^T]^T \quad (8)$$

and the interaction matrix is an $2n \times 6$ matrix given by concatenating n submatrices \mathbf{L}_{ij} given by

$$\mathbf{L}_{ij} = \begin{bmatrix} \frac{1}{z_j} & 0 & \frac{-x_j}{z_j} & -x_j y_j & 1 + x_j^2 & -y_j \\ 0 & \frac{1}{z_j} & \frac{-y_j}{z_j} & -1 - y_j^2 & x_j y_j & x_j \end{bmatrix} \quad (9)$$

where z_j is the depth of the j th feature point and the camera focal length is assumed (without loss of generality) to be unity.

If \mathbf{L}_{ij} has more than six rows (e.g., four or more points are used as features), a stabilizing camera velocity is given by the feedback control

$$\xi = -\lambda_i \mathbf{L}_i^+ \mathbf{e}_i \quad (10)$$

where $\mathbf{L}_i^+ = (\mathbf{L}_i^T \mathbf{L}_i)^{-1} \mathbf{L}_i^T$ is the general inverse of \mathbf{L}_i , and λ_i is a positive gain scalar. Combining Eqs. (7) and (10) gives the closed-loop feature error derivative

$$\dot{\mathbf{e}}_i(t) = -\lambda_i \mathbf{e}_i(t) \quad (11)$$

The closed-loop system is locally asymptotically stable. It is not globally asymptotically stable, as there exist camera positions or feature point configurations for which \mathbf{L}_i^+ is not full rank. See Espiau et al. (1992), Kelly et al. (2000), Deng et al. (2002), and Chaumette and Hutchinson (2006a) for detailed discussions of stability and robustness.

The feature error tends monotonically to zero. As such, the features will move along short paths in the image to their goal coordinates and are unlikely to leave the field of view. However, the system has no explicit control over the pose of the robot. As described in Chaumette (1998) and Corke and Hutchinson (2001), an IBVS system may perform large camera motions to achieve short image feature trajectories. Most industrial robot systems have a reachable space on the order of meters. Thus,

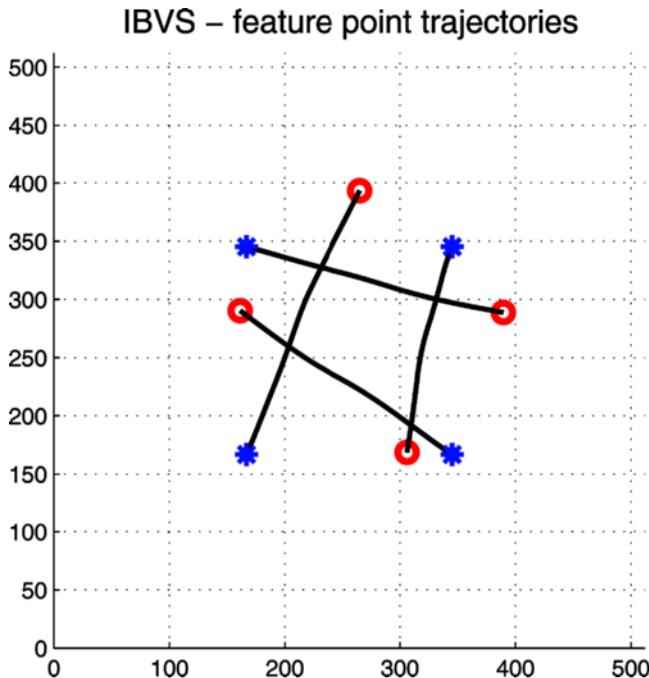


Figure 4. Example of typical trajectories for feature points in IBVS.

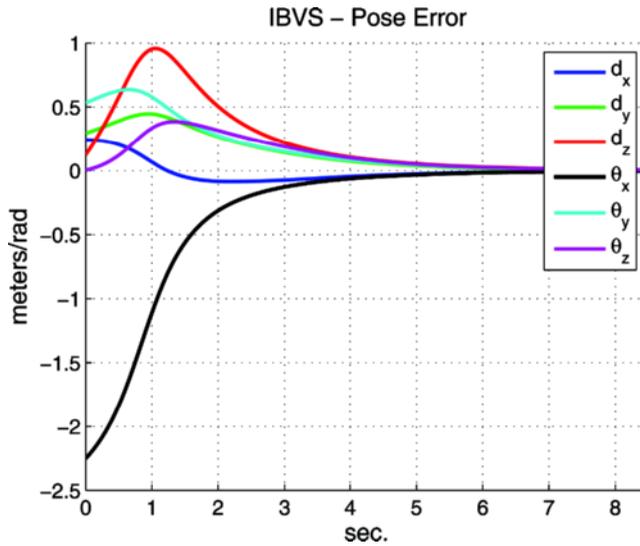


Figure 5. Example of typical camera pose error in IBVS.

camera retreat may cause the robot to extend to its joint limits during visual servoing, resulting in failure.

An example of this phenomenon can be seen in Figures 4–6. Figure 4 shows the feature point trajectories, which move along straight lines in the image. Figure 5 shows the camera pose error vector over time; camera retreat is apparent as the translation along the z -axis increases before eventually converging to zero. Figure 6

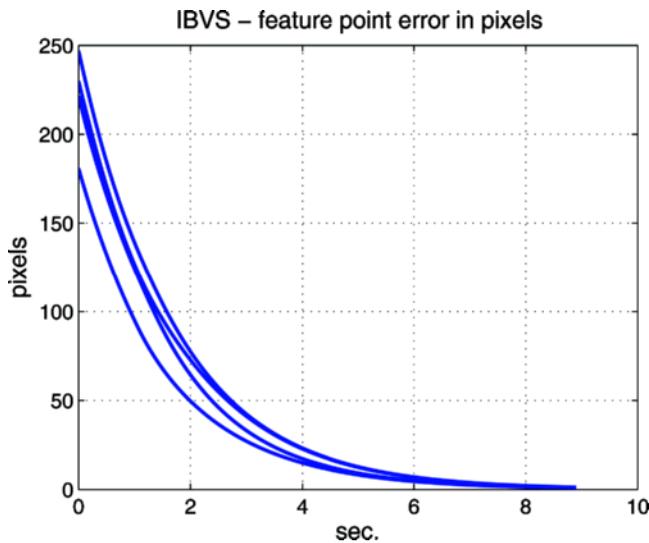


Figure 6. Example of typical feature point error in IBVS.

is the feature point error in pixels for the four points over time, all of which monotonically decrease to zero error.

2.3. Switched-System Visual Servoing

The theory of hybrid, switched control systems (i.e., systems that comprise a number of continuous subsystems and a discrete system that switches between them) has received notable attention in the control theory community (Brockett 1993; Branicky et al. 1998; Liberzon and Morse 1999; Liberzon 2003). In general, a hybrid switched system can be represented by the differential equation

$$\dot{x}(t) = f_{\sigma(t)}(x, t) : \sigma \in \{1 \dots n\} \quad (12)$$

where f_{σ} is a collection of n distinct functions. The solution to Eq. (12) is a pair $\{x(t), \sigma(t)\}$ giving the value of the state and switching variable over time. The functions $x(t)$ and $\sigma(t)$ are continuous from the right to insure both are locally Lipschitz.

In order to mitigate the troubling aspects of IBVS and PBVS, the authors have previously introduced switched-system visual servo controllers (Gans and Hutchinson 2002; Gans and Hutchinson 2003, 2007). These systems switch between using IBVS and PBVS at different points during the period of the task. Switching can be triggered depending on the state of the two errors e_p and e_i , at specific time intervals, or randomly.

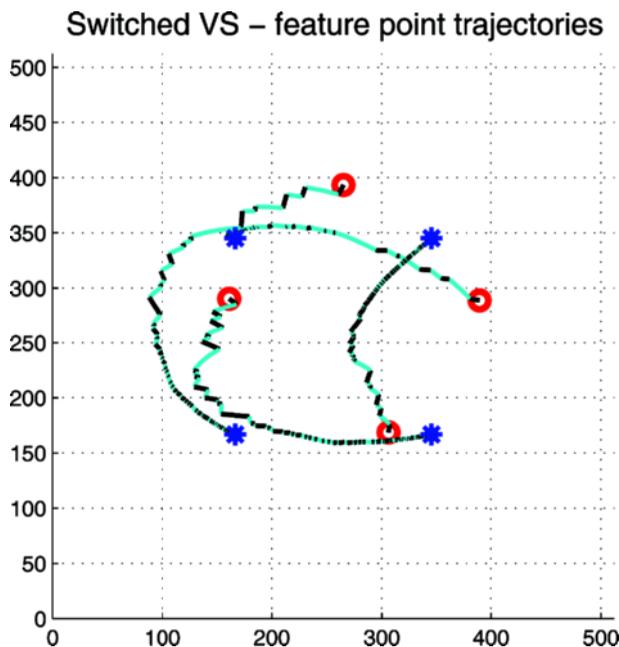


Figure 7. Example of typical trajectories for feature points in random switched visual servoing.

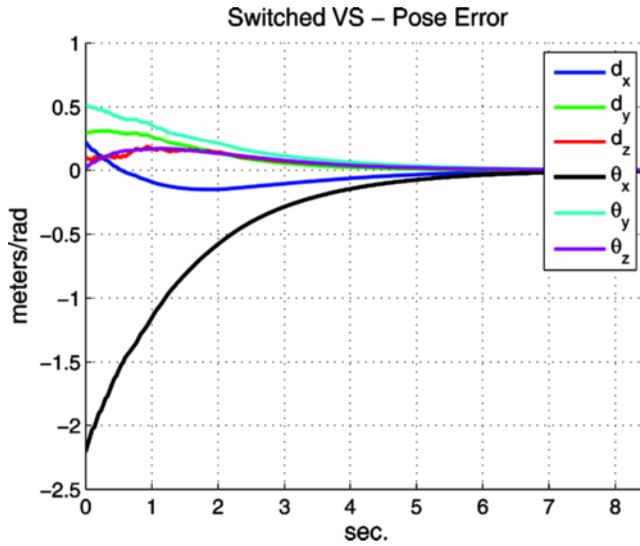


Figure 8. Example of typical camera pose error in random switched visual servoing.

This paper focuses on random switching, using a binary random variable to pick a system at each iteration of the algorithm. In Gans and Hutchinson (2003), a binary random variable was used, that gave with a 50% chance of selecting IBVS or PBVS at each iteration of the control loop. Experimental investigation demonstrated a marked decrease in failures due to extreme robot motions or lost feature points.

An example of performance of this hybrid system is seen in Figures 7–9. Figure 7 shows feature point trajectories. The feature point trajectory is now a slight

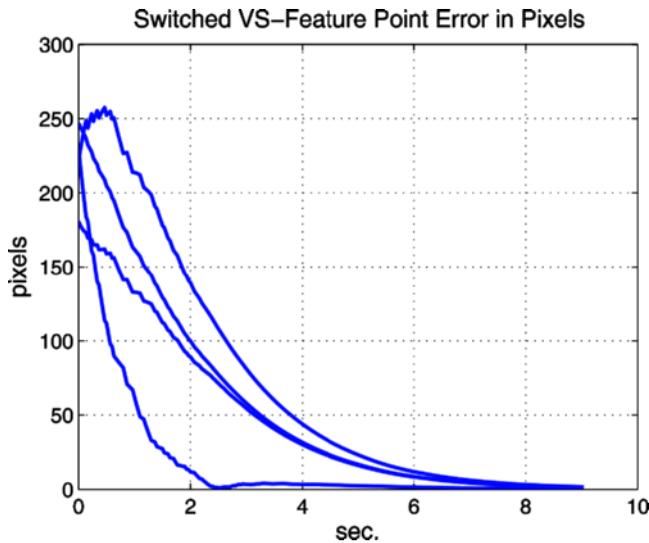


Figure 9. Example of typical feature point error in random switched visual servoing.

curve, as opposed to the previous large curve or straight path. Discontinuous effects of switching are visible in the feature point velocity. Figure 8 shows the camera pose error vector over time. There is some noticeable chattering in the pose error, but the effects are not strong and the pose error converges nicely to zero. Figure 9 is the feature point error in pixels for the four points over time. Some pixel error elements show a slight initial increase before eventually converging to zero.

3. MULTI-ATTRIBUTE UTILITY ANALYSIS IN THE DESIGN OF A SWITCHED SYSTEM VISUAL SERVO CONTROLLER

As discussed in section 2.3., the authors have previously used a switching visual servo system that had a 50% chance of selecting IBVS or PBVS at each iteration. This resulted in fewer system failures. However, while a switched system may be desirable to reduce the chance of failure, some tasks or configurations may be more lenient toward extremes in camera motion than image feature motion. A binary function that favors one system over the other will grant these results. This section details an experiment into the use of multi-attribute utility analysis (MAUA) to choose between multiple visual servo systems, including several binary switched systems. The steps of this experiment will help to introduce and illustrate the concepts of MAUA.

MAUA can be applied to determine the best binary function. An attribute is any quantity measured on a relative scale. An attribute can be objective measurements, statistical data, or even subjective data such as “satisfaction” rated on a scale. MAUA is a well established method for comparing options with multiple objectives. Furthermore, it allows the inclusion of not only preference, but attitudes toward risk (Merkhofer and Kenney 1987; Thurston 1990; Keeney and Raiffa 1993).

The person whose preferences are reflected in the system design is referred to as the decision maker (the decision maker could be several people or a committee, but for simplicity we refer to a single person). Typically this is someone knowledgeable in the field and closely involved in the task at hand. A few examples of decision makers include a project manager in charge of implementing a control system, or a customer for whom a system is being designed. MAUA is intrinsically an individual pursuit. A change of decision makers will require the analysis be run again.

The goal is to maximize the system’s expected utility for the decision maker. Utility is a unitless measurement, which allows vastly different attributes to be compared and incorporates the decision maker’s preference and risk aversion. Each attribute is assigned a utility function $U(x)$ that is monotonic with the attribute. The utility functions can then be weighted and combined to form a multi-attribute utility function (MAUF). If statistical data is available concerning the attributes, it can be used to solve expected value of the MAUF for different weights and pick the system that maximizes the expected utility. This will be elaborated upon below.

Given n attributes, x_1, \dots, x_n , the multi-attribute utility function, $U(x_1, x_2, \dots, x_n)$, is defined by the equation

$$1 + KU(x_1, x_2, \dots, x_n) = \prod_{i=1}^n [Kk_i U_i(x_i) + 1] \quad (13)$$

where U_i are individual utility functions for each attribute, k_i are individual scaling or weighting constants, and K is a normalizing constants that is the non-zero solution to

$$1 + K = \prod_{i=1}^n [Kk_i + 1] \tag{14}$$

There are several basic steps to the development of a MAUF, each of which will be presented here. First a decision maker must be chosen to make value judgments. Second is the choice of suitable attributes and confirmation that the decision maker can view them independently from each other. Third is the development of the individual utility functions and scaling constants. Finally, given information about probable outcomes for the task, the MAUF can be used to evaluate the different options. To gather statistical data, Monte Carlo analysis is performed by simulating each system performing a large number of tasks.

For the purposes of this problem, the decision maker was an electrical engineering graduate student, who has been involved in vision based robot control for two years. The first step is to determine attributes that measured non-ideal performance of a system. In the image plane, define a line segment for each feature from the initial feature point coordinate, \mathbf{p} , to the goal coordinate \mathbf{p}^* . At each iteration, measure the distance from each feature point to the line segment, and note the largest distance. This measurement will be referred to as d_{2D} , and is illustrated in Figure 10. IBVS is expected to move the features along trajectories very close to these line segments, so d_{2D} should remain small for all but the most difficult tasks. PBVS, which offers no control over the image features, is expected to have much larger measurements of d_{2D} .

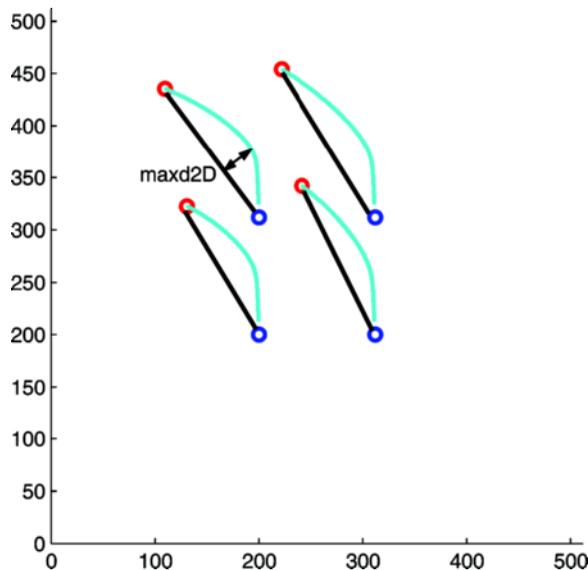


Figure 10. Illustration of d_{2D} .

Similarly, define a three-dimensional line segment in the camera workspace from the initial camera position \mathbf{T} to the goal positions, \mathbf{T}^* . At each iteration, measure the 3-D distance from the camera's current position to the line segment; this measurement will be referred to as d_{3D} . As the analysis is performed in simulation, the 3-D distance from the camera position to the line is readily available. PBVS is expected to have small measurements for d_{3D} , while it will be much larger for IBVS.

For two of the attributes, we used the maximum d_{2D} and maximum d_{3D} seen over the course of a visual servo task. These two attributes are denoted as x_{2D} and x_{3D} . As discussed in section 2, IBVS is expected, on average, to have a small values of x_{2D} and large values of x_{3D} while PBVS is expected to have a small values of x_{3D} and large values of x_{2D} .

The final attribute considered is the failure rate for each system, where failure occurs any time the system fails to reduce the image and pose errors to zero. This can occur when a feature point leaves the field of view, the camera moves outside of a defined boundary, or the system converges to a local minimum other than the origin. The attribute of failure rate is denoted x_{fail} . This gives three attributes.

To use MAUA it is necessary to insure that the decision maker can view these attributes independently in the sense of mutual preferential independence and mutual utility independence (Keeney and Raiffa 1993). Preferential independence exists if a decision maker's preference for attribute Y does not depend on attribute X . For example, a small value of Y might always be preferred to a large value, regardless of the value of X . Preferential independence exists for most decisions.

Utility independence is more complex. It insures that a decision maker's preferences for an attribute Y for *uncertain* outcomes do not depend on X . For example, suppose the decision maker prefers the first of two scenarios for attributes X and Y :

1. 50% chance $Y = Y_1$, 50% chance $Y = Y_2$ and $X = X_1$
2. 30% chance $Y = Y_1$, 70% chance $Y = Y_2$ and $X = X_1$

Then for mutual independence to hold, he or she must also prefer the first option of the scenarios:

1. 50% chance $Y = Y_1$, 50% chance $Y = Y_2$ and 50% chance $X = X_1$, 50% chance $X = X_2$
2. 30% chance $Y = Y_1$, 70% chance $Y = Y_2$ and 50% chance $X = X_1$, 50% chance $X = X_2$

That is, a change in the probable value of X does not affect the preference. Both independence requirements must hold for any values of the attributes, and all attributes must be tested against each other.

For these attributes, the decision maker determined they were mutually independent. The next step was to determine individual utility functions for each attribute, and to build the multi-attribute utility function. The decision maker decided that $x_{2D} = 175$ pixels was the maximum he would accept and $x_{3D} = 1.5$ meters was the largest acceptable x_{3D} , since he felt that exceeding these values would likely lead to system failure. The largest failure rate the decision maker was willing to

accept was 20%. This gives us the following utilities for the extremes of the domains,

$$U_{2D}(175) = 0, \quad U_{3D}(1.5) = 0, \quad U_{\text{fail}}(20) = 0$$

$$U_{2D}(0) = 1, \quad U_{3D}(0) = 1, \quad U_{\text{fail}}(0) = 1$$

Note that it is possible to have x_{2D} and x_{3D} larger than the decision maker's chosen maximums, indeed during simulations the systems often exceeded these limits. Any value larger than these maximums is given a utility of zero.

The decision maker was queried using the certainty-equivalent and probability methods to determine the individual utility functions (Thurston 1991). In these methods, a utility of 1 is assumed for the best outcome, which is 0 for all three attributes here, and a utility of 0 is assumed for the worst acceptable outcome. The decision maker is presented a series of outcomes, and asked to choose the preferred outcome. This allows the designer to gage the decision maker's perceived utility of each attribute at several values of the attribute.

It is important to note that a utility of $x \in (0, 1)$ often does not correspond to the attribute value at that same percentile between the highest and lowest values of the utility function range. For example, the point at exactly half way between the best and worst attribute values often does not have utility of 0.5.

The chosen utility values reveal characteristics of the decision maker. A decision maker who assigns utility x to the same percentile between the highest and lowest values for all x (i.e., a utility of 0.25 to one quarter of the way between best and worst attribute values, a utility of 0.5 halfway between best and worst attribute levels, and so on) is said to be *risk neutral* and will have a linear utility function. A tendency to assign a utility of x to a value below the risk neutral line indicates the decision maker is *risk adverse* and will accept a guaranteed, mediocre performance over an uncertain performance with high risk and high reward. The opposite is known as *risk seeking*, in which the decision maker will accept the risk of bad performance for the chance of getting good performance and prefer this chance to a known moderate outcome. It is not unusual to show different behaviors over different portions of the utility function range.

The decision maker was queried, and utility values were determined for several attribute values. Polynomial functions were then line fitted to the data points using Matlab. The degree of the polynomial was chosen for each function to give the best fit (i.e., least residual error) and ensure a monotonic function. The resulting utility functions are as follows:

$$U_{2D}(x_{2D}) = 3.7 \times 10^8 x_{2D}^3 - 8.6 \times 10^6 x_{2D}^2 - 0.0053x + 1$$

$$U_{3D}(x_{3D}) = 0.14x_{3D}^2 - 0.87x_{3D} + 1$$

$$U_{\text{fail}}(x_{\text{fail}}) = 2.8 \times 10^{-5} x_{\text{fail}}^4 - .0012x_{\text{fail}}^3 + .017x_{\text{fail}}^2 - .13x_{\text{fail}} + 1$$

Plots of the points and fitted functions for the nonlinear functions are seen in Figures 11–13. They are monotonic functions, which is necessary for use in a MAUF. The functions are also close to linear, indicating the decision maker is fairly risk neutral. Note that these functions reflect the preferences of the specific decision maker.

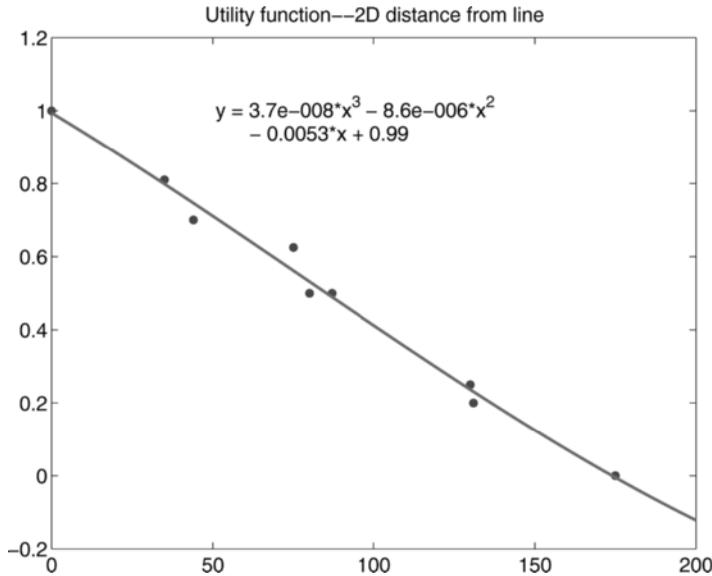


Figure 11. Utility function for 2-D.

The decision maker is then quizzed as follows to determine the value of the scaling constants k_i in Eqs. (13) and (14) (Thurston 1991). The decision maker assigns a utility U_{chosen} to the combined function Eq. (13) when two of the attributes are at their worst and one attribute is at its best value. The k_i can then be solved for

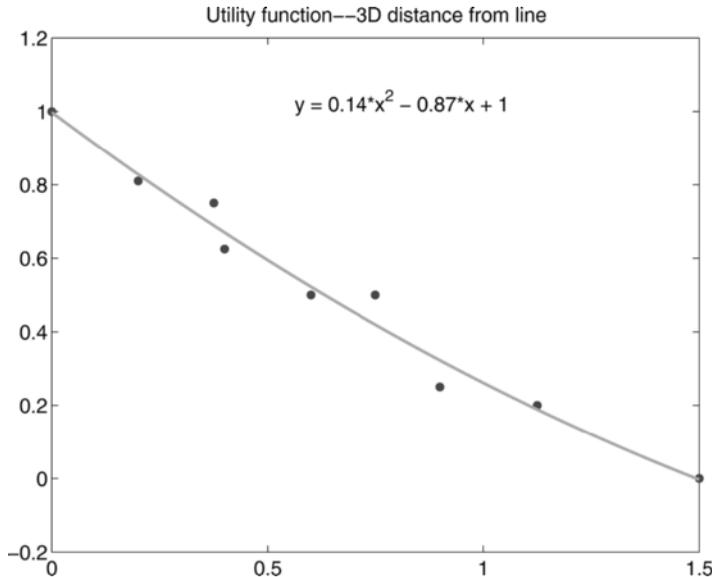


Figure 12. Utility function for 3-D.

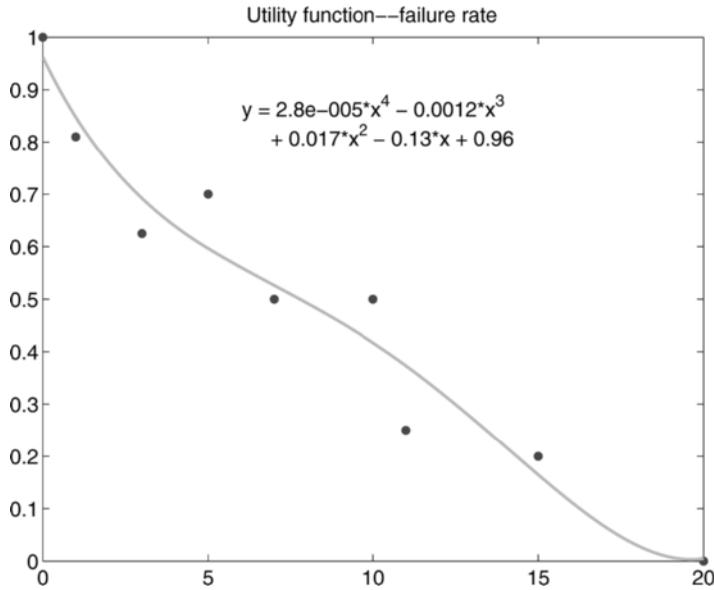


Figure 13. Utility function for failure rate.

the attribute at its best. Using the values found in Eq. (15), k_2D is found as follows:

$$\begin{aligned}
 1 + KU(x_{2D}, x_{3D}, x_{\text{fail}}) &= [Kk_{2D}U_{2D}(x_{2D}) + 1] \times [Kk_{3D}U_{3D}(x_{3D}) + 1] \\
 &\quad \times [Kk_{\text{fail}}U_{\text{fail}}(x_{\text{fail}}) + 1] \\
 1 + KU_{\text{chosen}} &= [Kk_{2D}U_{2D}(0) + 1] \times [Kk_{3D}U_{3D}(1.5) + 1] \\
 &\quad \times [Kk_{\text{fail}}U_{\text{fail}}(20) + 1] \\
 U_{\text{chosen}} &= k_2D
 \end{aligned}$$

This was repeated for all k_i , which were determined to be $k_{d_{2D}} = 0.35$, $k_{d_{3D}} = 0.375$, $k_{\text{fail}} = 0.15$. As stated, K is the non-zero solution to Eq. (14) and was determined to be $K = -0.8187$. This gives us enough information to use the MAUF $U(x_{2D}, x_{3D}, x_{\text{fail}})$ as in Eq. (13). Plots of the MAUF for U vs x_{2D} and x_{3D} for several values of x_{fail} are seen in Figures 14–16.

In order to gain information on the probable performance of the visual servo systems, Monte Carlo analysis was performed for IBVS, PBVS, and several switched systems with differing binary functions. Specifically, three switched systems were tested, along with IBVS and PBVS. The three switched systems had a 25%, 50% or 75% chance of selecting IBVS at each iteration of the control loop. A six-dimensional configuration space (translation and rotation about three axes) was sampled, to acquire 30,000 unique initial camera poses where the initial image was in the camera field of view.

Each visual servo system was tested for its ability to regulate each initial camera pose to the goal pose. The attributes x_{2D} and x_{3D} were recorded for each visual servo system from each initial pose. A failure was recorded if a feature point left the field of view, the camera left the sampled space, or the system converged to a local

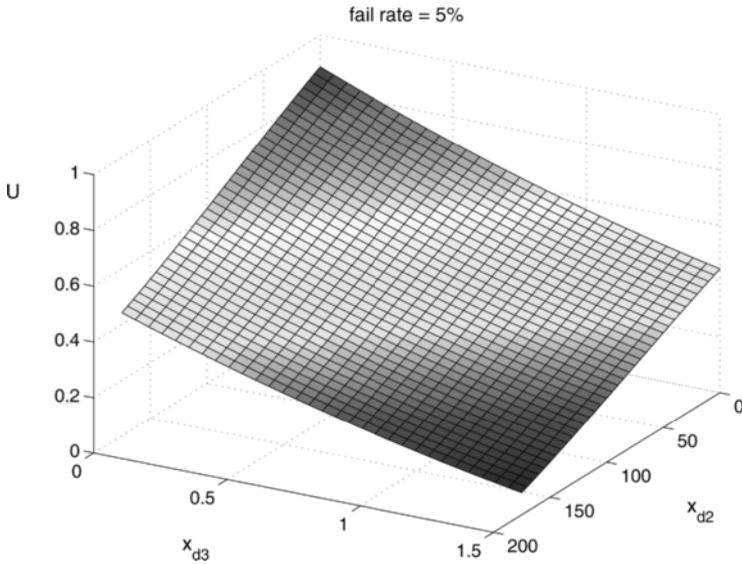


Figure 14. MAUF for varying x_{2D} and x_{3D} when $x_{fail} = 5\%$.

minimum other than the goal pose. The simulation was not halted if a system was considered to have failed, thus large values of x_{2D} and x_{3D} are possible. Running the simulations in Matlab took approximately sixteen hours of computer time on a single core, Pentium 4 processor, though no efforts were made to optimize the Matlab code for speed.

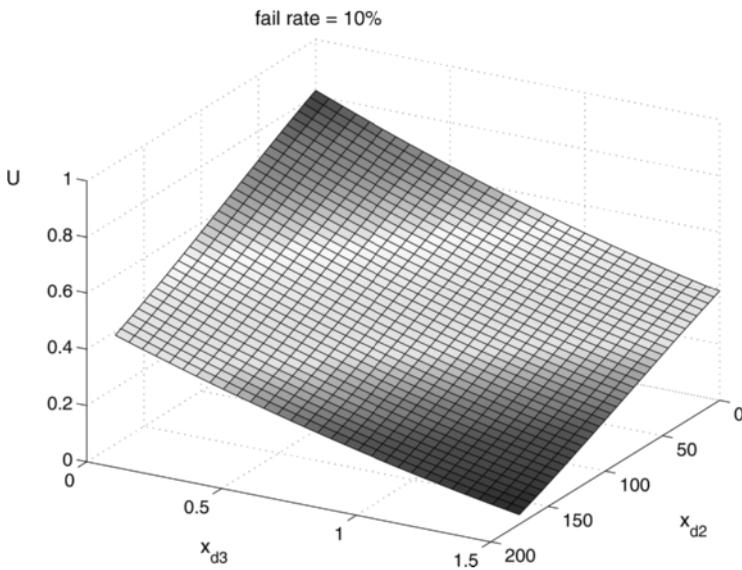


Figure 15. MAUF for varying x_{2D} and x_{3D} when $x_{fail} = 10\%$.

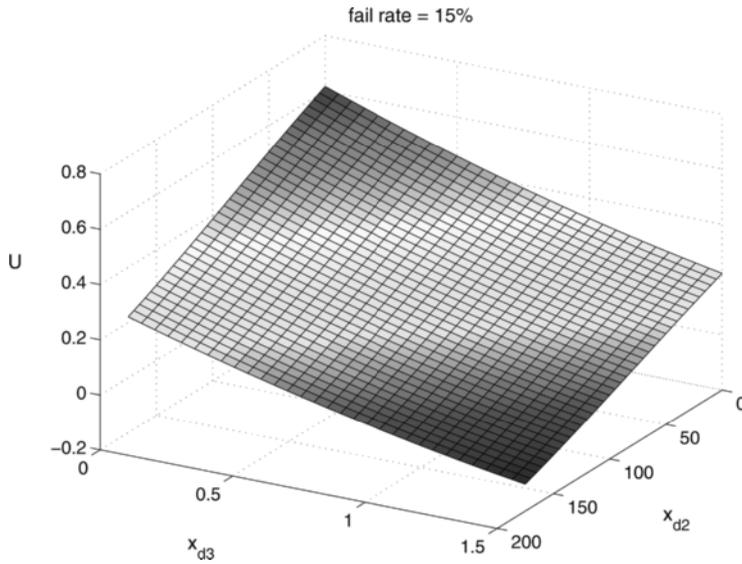


Figure 16. MAUF for varying x_{2D} and x_{3D} when $x_{fail} = 15\%$.

The size of the sampled space was a $4\text{ m} \times 4\text{ m} \times 3.5\text{ m}$ box with the simulated image features at the center of the bottom face. The orientations ranged from -2π to 2π for rotation about the camera optical axis (z -axis), and $-\pi$ to π for rotations about the x - and y -axes. These two axes were more limited since a rotation with magnitude greater than π will result in the feature points lying behind the camera. The image was 512×512 pixels, and the location of points in the goal images can

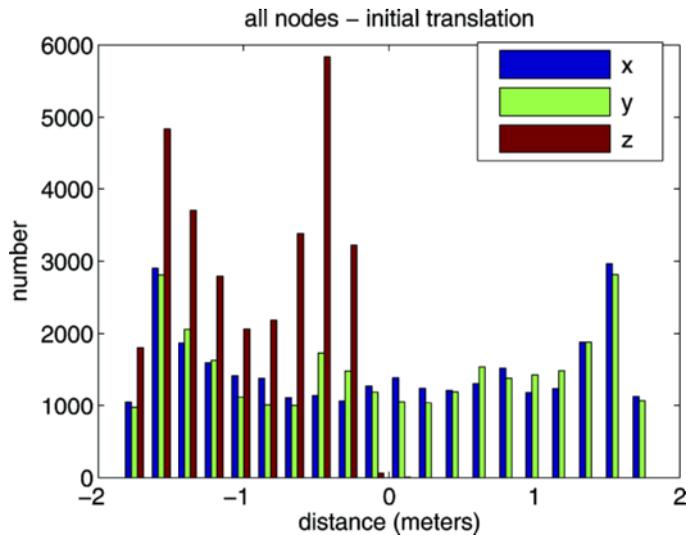


Figure 17. Histograms of the sampled initial translation for the switched system design problem.

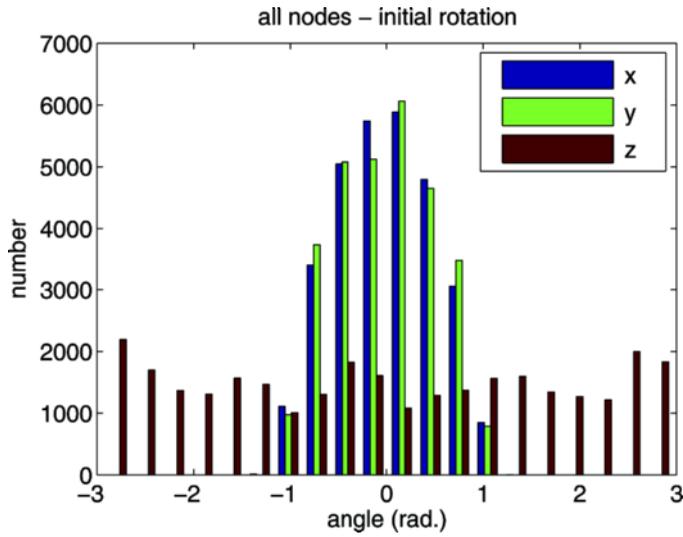


Figure 18. Histograms of the sampled initial rotation for the switched system design problem.

be seen in Figures 1, 4, and 7. Histograms of the sampled translations and rotations are seen in Figures 17 and 18, respectively. The sampling is not uniform due to the sampling method used, a rapidly-exploring random tree (RRT) (La Valle and Kuffner 2000).

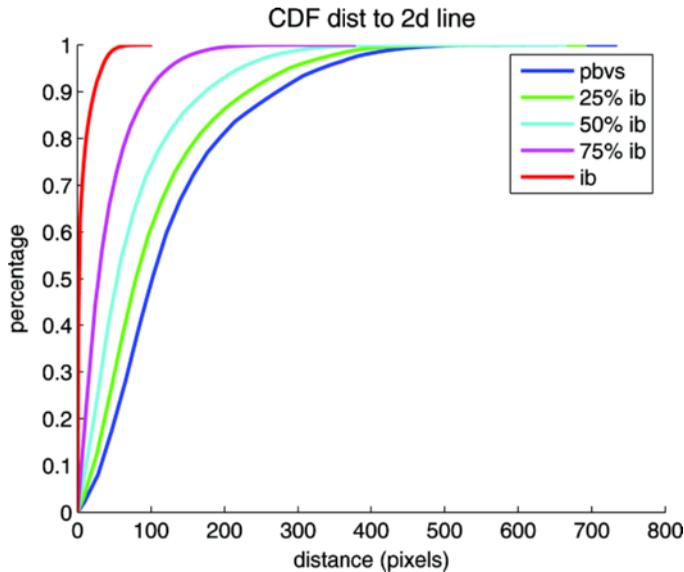


Figure 19. Cumulative distribution functions for x_{2D} for the switched system design problem.

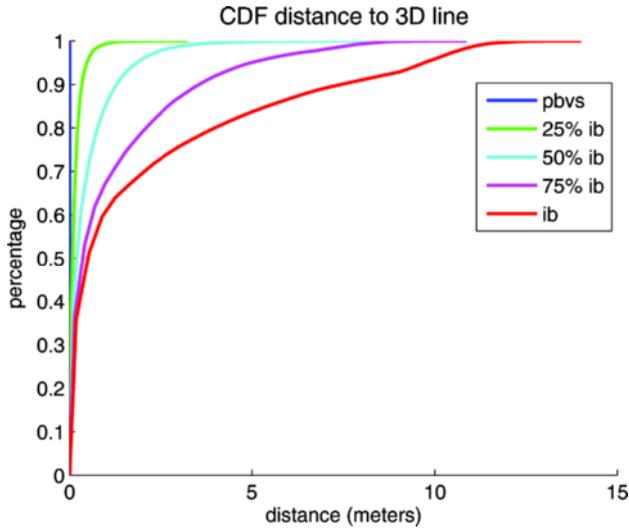


Figure 20. Cumulative distribution functions for x_{3D} for the switched system design problem.

Figures 19 and 20 show cumulative distribution functions for x_{2D} , x_{3D} , respectively. As expected, the probable x_{2D} is much smaller for IBVS than PBVS. Likewise the probable x_{3D} is much smaller for PBVS than IBVS. The hybrid systems lie in between. The failure rate is a discrete value, so there are no density functions. The failure rates are given in Table 1.

This information is used with the MAUF to evaluate the visual servoing systems and choose the system with the highest utility. This information is most easily presented in the form of a decision tree, which is presented in Figure 21. There are five chance nodes, one for each visual servo system. In order to accurately gage the performance of each system, while keeping the amount of data manageable, the chance nodes have five branches corresponding to five ranges of decreasing performance. Each branch has a weighting of 0.2 corresponding to a 20% chance of that branch being true. In effect, the first branch supposes there is a 20% chance the system will operate somewhere in the best 20% of performance for that system. The value of the attribute that corresponds to 0.2 in the CDF's are then used in the MAUF, so the utility value that results is best thought of as a lower bound. The second branch states there is a 20% chance of performance to the 20%–40% range

Table 1. Failure rates for visual servoing simulations for the switched system design problem

System	Rate (%)
IBVS	19.78
PBVS	14.23
.25 IBVS	9.11
.5 IBVS	11.48
.75 IBVS	16.39

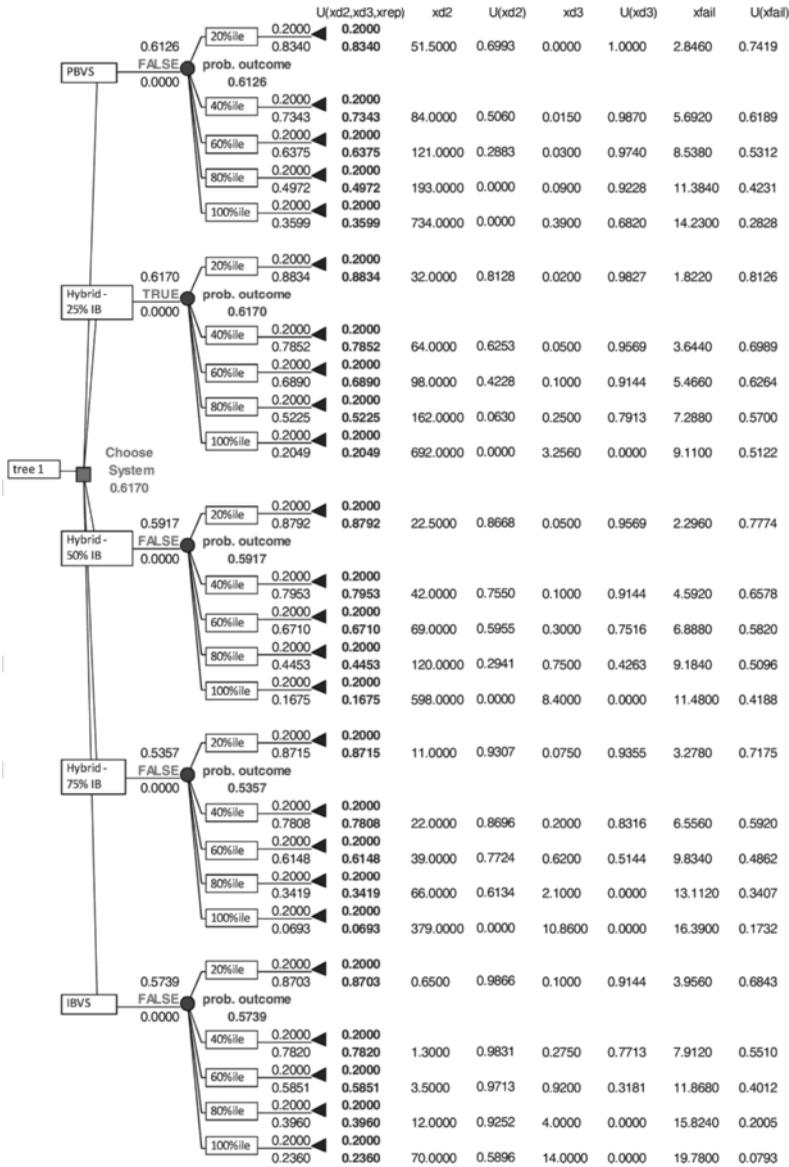


Figure 21. Decision tree for the switched system design problem.

of system performance, and gives the lower bound of utility in this range. The subsequent branches continue this analysis trend. A detailed examination of the decision tree will be presented in the section 3.1.

3.1. Results

A decision tree was created to present the results of the multi-attribute utility analysis. It is shown in Figure 21. As previously described, this tree has a decision for

each of the systems being considered. Each decision node has five branches to reflect a 20% chance of a certain performance range. At the end of each branch are seven columns. The first column is the expected utility given by the MAUF for that performance percentile and the weighting of 0.2 that each performance percentile receives. The next six columns give the value of each attribute that the system can expect at that performance level and the utility values of the individual attribute functions. The weighed sum of the MAUF's at each performance level is given in red next to the node. The node of the system that maximizes expected utility is denoted "TRUE." The maximum expected utility is reprinted at the central node.

For the utility function and weighting functions derived here, the switched system choosing IBVS with 25% probability comes out on top with an expected utility of 0.617. PBVS is second with expected utility of 0.613, followed by the neutral switched system and IBVS with expected utilities of 0.592 and 0.574, respectively. The switched system biased toward IBVS comes in last with expected utility of 0.536. These results are collected in Table 2.

It can be seen that the expected utility is a combination of the decision maker's preferences and the statistical data for each system's performance. The decision maker considered 2-D and 3-D performance to be of nearly equal importance, as reflected in the similar weights for the attributes x_{2D} and x_{3D} , and held the overall failure rate to be more important, as seen in the higher weighting of x_{fail} . Looking at the distribution functions in Figures 19 and 20, it appears that IBVS and IBVS biased switched systems have shallow distribution functions, and therefore large values of the attribute x_{3D} , as the cumulative probability approaches 1. In comparison, PBVS and PBVS biased switched systems remain more moderate in the attribute x_{2D} as the cumulative probability approaches 1.

Sensitivity analysis of the results shows the impact of the decision maker's weighting constants to the expected utilities. To examine the sensitivity of the outcome to changes in the weighting constants, vary a single constant from 0 to 1 while keeping the others the same as chosen by the decision maker. Figures 22–24 shows the resulting utilities for each system as a function of the shifting attribute. Examining the results when a gain is set to 0 will give insight into how analysis using fewer attributes will affect the outcome. For example, when $k_{fail} = 0$, IBVS is the best choice.

As seen in Figure 22, when varying $k_{d_{2D}}$, IBVS would be chosen for $k_{d_{2D}} > 0.45$. If $k_{d_{2D}} = 0$, performance in the image error is not considered and PBVS becomes the top choice. Similarly, as seen in Figure 23, when varying $k_{d_{3D}}$, PBVS becomes the preferred choice only when $k_{d_{3D}} > 0.4$. If $k_{d_{3D}} = 0$, IBVS is the top choice. Varying k_{fail} ,

Table 2. Expected utility for visual servoing systems for the switched system design problem

System	Expected utility
.25 IBVS	0.617
PBVS	0.613
.5 IBVS	0.592
IBVS	0.574
.75 IBVS	0.536

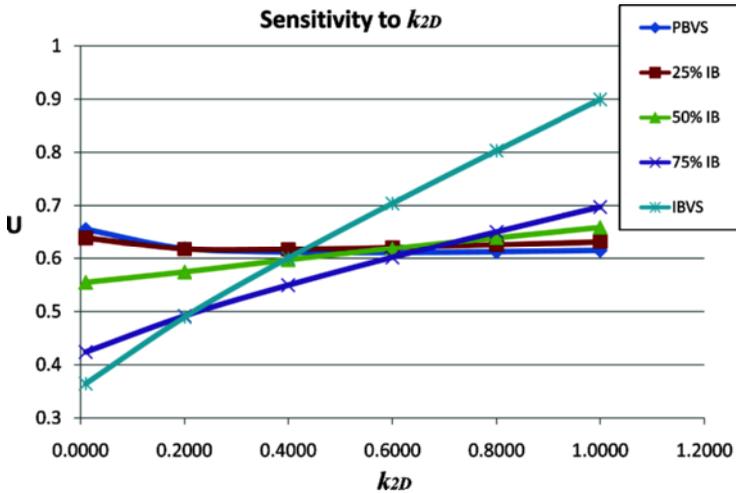


Figure 22. Analysis of sensitivity of the MAUF to k_{2D} for the switched system design problem.

it can be seen in Figure 24 that if $0 \leq k_{fail} < 0.15$, IBVS is the best choice, and PBVS becomes the preferred choice if $0.15 \leq k_{fail} < 0.34$. Looking at all sensitivity graphs, it seems that the switched system biased toward PBVS is generally preferred to the other switched systems, as the weighting constants are varied.

It is important to note that these results, like those for any MAUA, are very individualistic. They hold for a specific decision maker and for the specific set of statistical data gathered. A different decision maker may have very different utility functions and weighting constants, and varying the parameters of the simulations

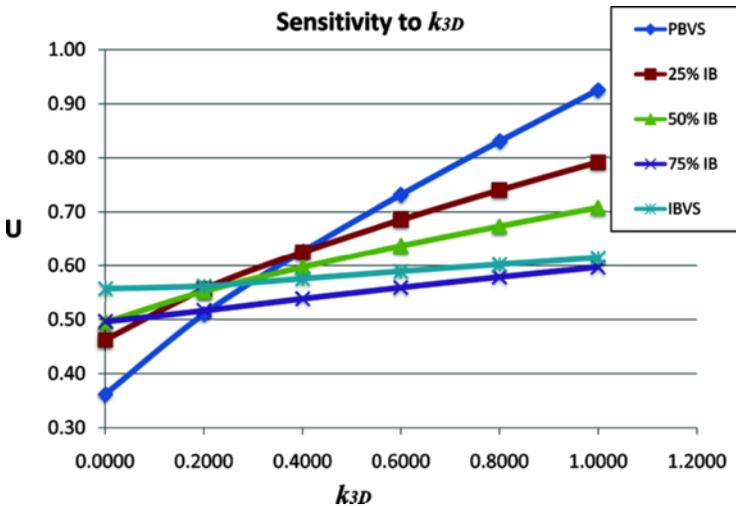


Figure 23. Analysis of sensitivity of the MAUF to k_{3D} for the switched system design problem.

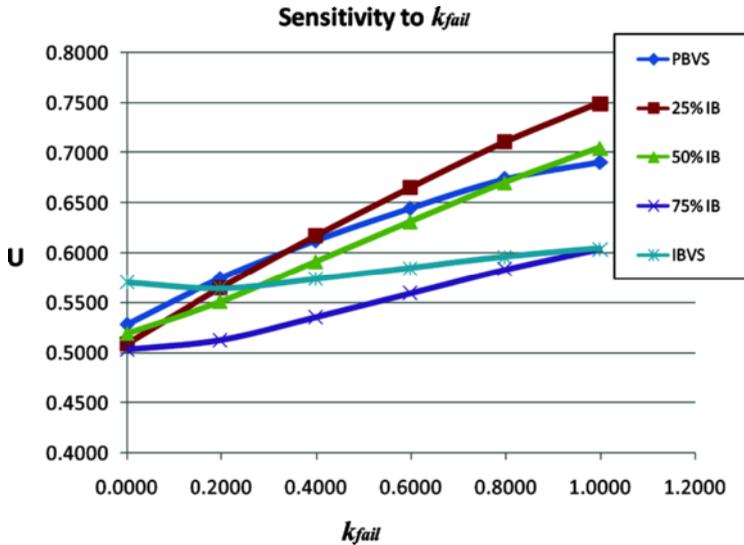


Figure 24. Analysis of sensitivity of the MAUF to k_{fail} for the switched system design problem.

could generate different statistical data. MAUA is designed to tailor a specific system to specific needs.

4. MULTI-ATTRIBUTE UTILITY ANALYSIS IN THE CHOICE OF OPTICS AND VISUAL SERVO CONTROLLER FOR A UAV REFUELING TASK

In section 3, MAUA was introduced through an example of choosing a binary switching rule for a switched-system visual servo control. In this section, MAUA is applied to choose a visual servo controller and camera optics for the task of refueling an UAV. In this scenario, a fuel tanker plane lowers a funnel-shaped fuel drogue, and the fueling UAV must insert a probe into the drogue in order to receive fuel. This is illustrated in Figure 25 for the case of two unmanned air vehicles with approximately three meter wingspan. This is a variation of a docking task.

The decision maker for this task was Stephen Korn, who was centrally involved in the design of the task scenario and Monte Carlo analysis. Mr. Korn is the Director for Interdisciplinary Research at the University of Florida Research and Engineering Educational Facility (UF-REEF) in Shalimar, FL, USA. He has more than 35 years of aerospace engineering experience, including four years in systems program management in the A-10 aircraft System Program Office at Wright-Patterson AFB, and 31 years of experience in the Air Force laboratory system developing munitions science and technology.

It was decided to test IBVS and PBVS systems using lenses with different focal lengths. Competing performance needs were chosen for this scenario. First, it is desired to have a small amount of residual pose error when the goal image has been reduced, despite the presence of image noise. This favors a narrow angle lens (i.e., long focal length). Second, it is desirable to keep the tanker plane in the field of



Figure 25. Illustration of the refueling task.

view while docking to avoid collision, which would favor a wide-angle lens (i.e., short focal length). Third, it is desirable to dock as quickly as possible. These performance needs lead to three metrics to use in MAUA: remaining pose error when visual servoing has stopped, remaining pose error when the tanker plane leaves the field of view, and time until visual servoing stops. These attributes are denoted as x_{err} , x_{FOV} and x_{time} , respectively.

The Monte Carlo analysis was set up as follows. To allow for vision-based docking control in a variety of light conditions, infrared (IR) light arrays can be attached to the drogue or the fuel boom. The light array was simulated by a $0.2\text{ m} \times 0.2\text{ m}$ square configuration of feature points. An IR array can be fixed to the tanker fuselage as well, which was simulated a $0.4\text{ m} \times 0.4\text{ m}$ square configuration of feature points. The drogue points were rigidly positioned 1m below and 1.2m behind the tanker points. The tanker flew with constant linear velocity and no angular velocity, but a random disturbance was added to each velocity element at each image sample time. This disturbance was modeled as a normally distributed random process with zero mean and linear velocity in three dimensions with standard deviation of $1\text{ mm}/0.033\text{ s}$ (i.e., 1 mm per video frame) and angular velocity of $0.0003\text{ rad}/0.033\text{ s}$.

The camera was simulated with an 1024×768 pixel array, and lenses with focal lengths of 6 mm, 12 mm, and 25 mm. Gaussian white noise with zero mean and one pixel standard deviation was added to the projected image points, which were then rounded to the nearest integer to simulate quantization noise. A goal image of the fuel drogue was taken at the docking position at 0.5m from the drogue with no noise. This goal image is suitable for IBVS and PBVS using the homography matrix (Faugeras and Lustman 1988; Zhang and Hanson 1996). IBVS and homography PBVS systems were simulated using each focal length. Ten thousand starting positions were sampled from a three dimensional sphere of radius 5 m, centered 10m from the drone, with roll, pitch and yaw angles sampled from $-\pi/4$ to $\pi/4$. Histograms of the sampled translations and rotations are seen in Figures 26 and 27,

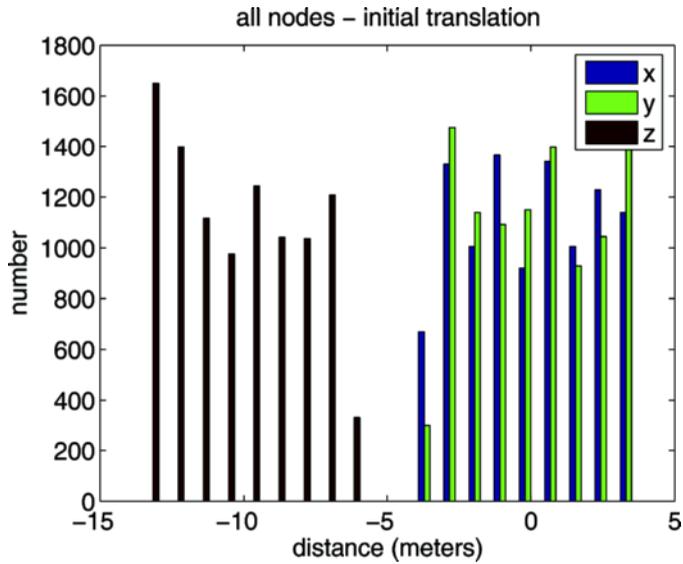


Figure 26. Histogram of the sampled initial translation for the refueling design problem.

respectively. Visual servoing was halted if the average image error was less than two pixels or if 33 s passed.

It was ensured that all start positions had both the drogue and tanker points in view. Different gains were used for each system and each focal length. Gains were chosen such that each system/lens pair could complete the first 100 tasks in the same

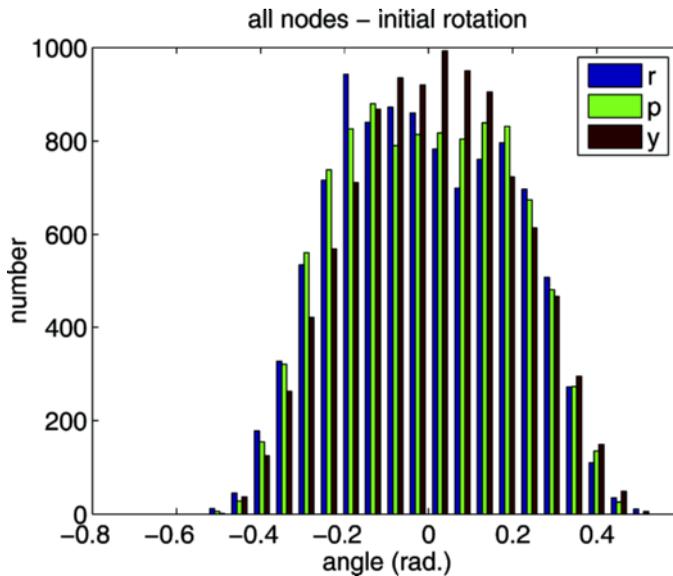


Figure 27. Histogram of the sampled initial rotation for the refueling design problem.

total amount of time if no noise added to the image or velocity. Each system was simulated from each starting pose three times, for a total of 30,000 tests for each system/lens. Figures 28–30 shows cumulative distribution functions for x_{err} , x_{FOV} , and x_{time} , respectively. IBVS and PBVS perform similarly for each focal length. As expected, a shorter focal length generally keeps the tanker in view longer, and the longer focal length has less residual error. The short focal length tends to finish quicker, as the courser resolution made it more likely that the average image error was under two pixels.

It was determined that the decision maker viewed these attributes as mutually independent. The next step was to determine individual utility functions for each attribute, and to build the multi-attribute utility function. The decision maker decided that $x_{\text{err}} = 0.1$ m (i.e., half the width of the drogue) was the maximum final pose error he would accept, as larger error would likely result in a failure to dock with the drogue. Based on initial results of the simulations, a value of 15 m was selected as the largest acceptable value for x_{FOV} and a maximum allowable run time of 30 s was selected for x_{time} . This gives us the following utilities for the extremes of the domains,

$$\begin{aligned} U_{\text{err}}(.1) &= 0, & U_{\text{FOV}}(15) &= 0, & U_{\text{time}}(30) &= 0 \\ U_{\text{err}}(0) &= 1, & U_{\text{FOV}}(0) &= 1, & U_{\text{time}}(0) &= 1 \end{aligned} \quad (15)$$

The decision maker was queried using the certainty-equivalent and probability methods to determine utility functions. The decision maker felt that the utility function for x_{time} should be linear. Points of the utility functions were determined for several values of k_{err} and k_{FOV} , and polynomial functions were line fitted to the points.

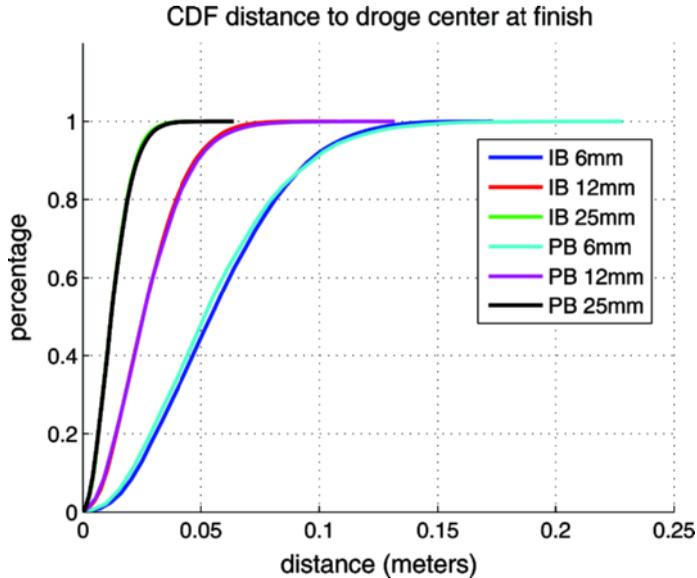


Figure 28. Cumulative distribution functions of x_{err} for the refueling design problem.

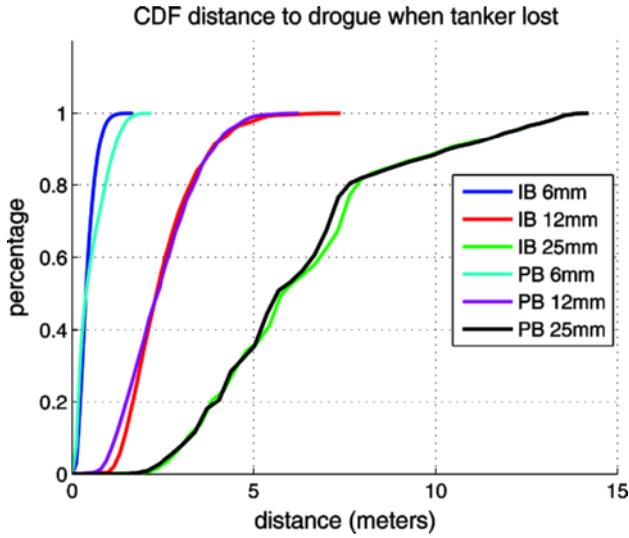


Figure 29. Cumulative distribution functions of x_{FOV} for the refueling design problem.

The resulting utility functions are as follows.

$$U_{err}(x_{err}) = -1.2 \times 10^6 x_{err}^5 + 2.4 \times 10^5 x_{err}^4 - 1.4 \times 10^4 x_{err}^3 + 1.8 \times 10^2 x_{err}^2 - 7.9 x_{err} + 1$$

$$U_{FOV}(x_{FOV}) = .0009 x_{FOV}^3 - 0.021 x_{FOV}^2 + 0.045 x_{FOV} + 1$$

$$U_{time}(x_{time}) = 1 - \frac{x_{time}}{30}$$

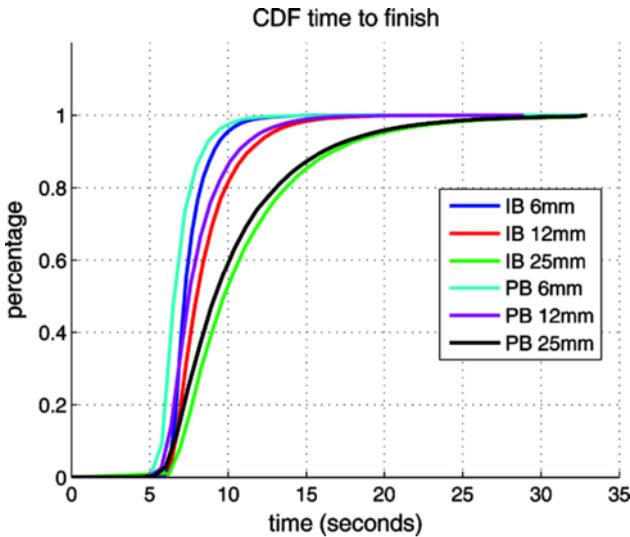


Figure 30. Cumulative distribution functions of x_{time} for the refueling design problem.

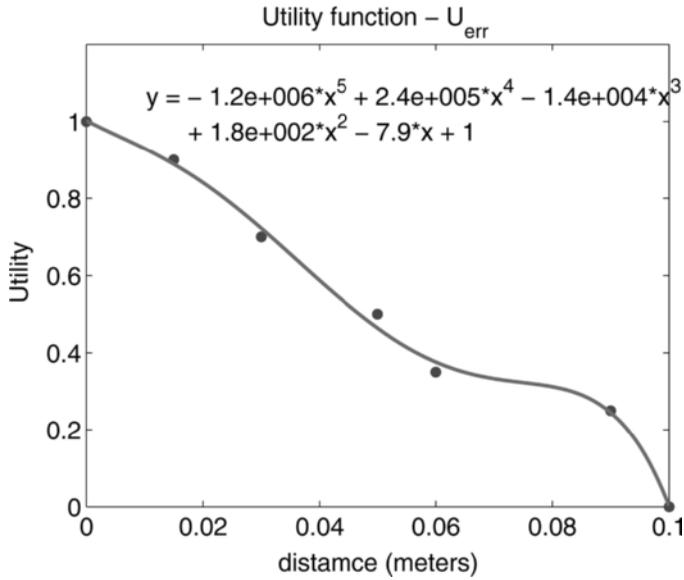


Figure 31. Utility function for k_{err} .

The functions $U_{err}(x_{err})$ and $U_{FOV}(x_{FOV})$ are plotted in Figures 31 and 32, respectively. It can be seen that the fitted function for $U_{FOV}(x_{FOV})$ is not quite monotonic, but it is the function with the smallest residual error to the sample points and deviates from monotonicity by at most 0.08.

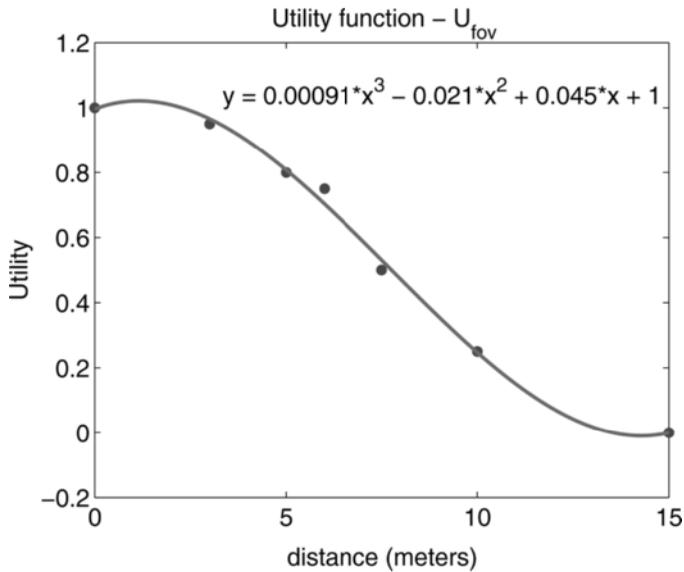


Figure 32. Utility function for k_{FOV} .

Table 3. Expected utility for visual servoing systems and lens choice for the refueling design problem

System	Expected utility
PBVS 6 mm	0.912
IBVS 6 mm	0.906
IBVS 12 mm	0.882
IBVS 25 mm	0.782
PBVS 25 mm	0.781
PBVS 12 mm	0.778

Attribute gains were chosen as described in section 3. The gains were determined to be $k_{FOV} = 0.8$, $k_{err} = 0.65$, and $k_{time} = 0.3$. This matches the decision maker’s feelings that losing sight of the tanker plane is potentially dangerous scenario and should be avoided. He felt that a failure to dock and was not as serious a problem, as the plane could fall back and make a second attempt. As stated, K is the non-zero solution to Eq. (14) and was determined to be $K = -0.925$. This gives us enough information to use the MAUF $U(x_{err}, x_{FOV}, x_{time})$ as in Eq. (13).

With the gains determined, MAUA was performed for all systems and lens combinations. PBVS using 6 mm lens came out on top with an expected utility of 0.912. IBVS/6 mm was a close second with expected utility of 0.906. Clearly the strong weighting $k_{FOV} = 0.8$ leads to choosing a wide-angle lens. All of the results are collected in Table 3. The results of the analysis are presented in a decision tree shown in Figure 34.

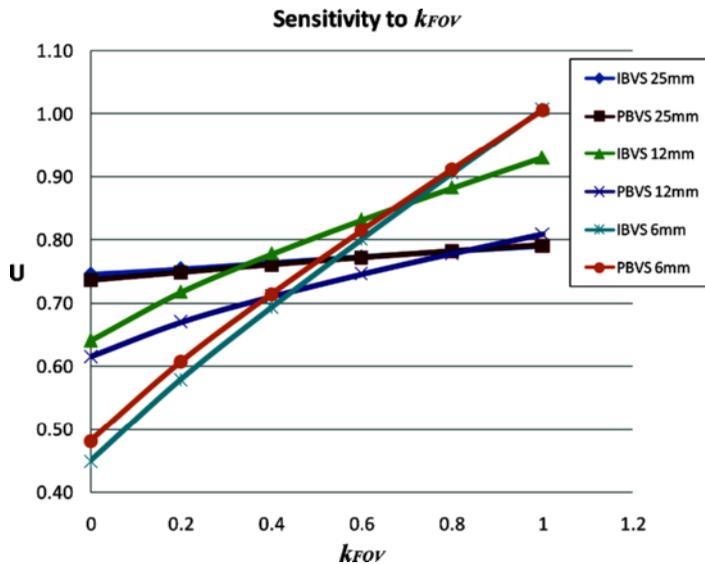


Figure 33. Analysis of sensitivity of MAUF to k_{FOV} for the refueling design problem.

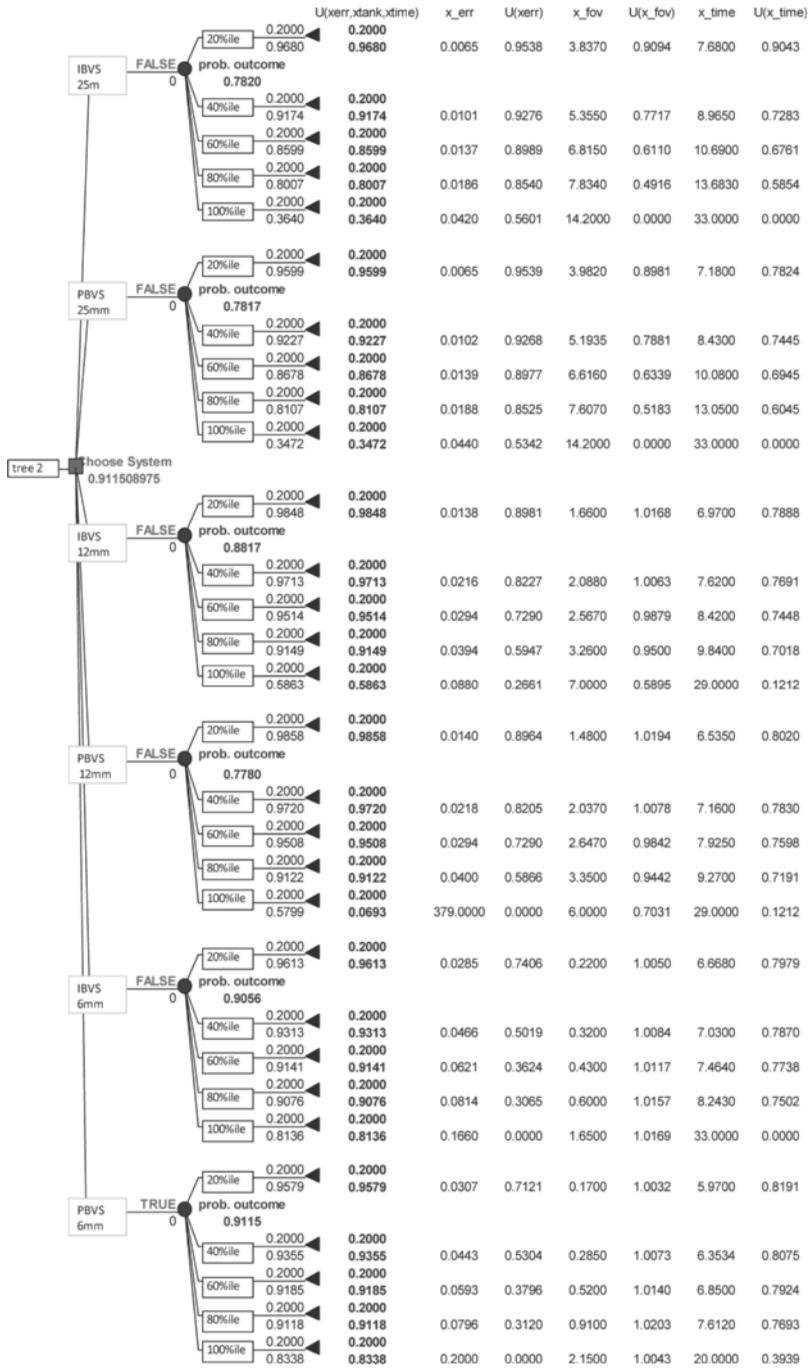


Figure 34. Decision tree for the refueling design problem.

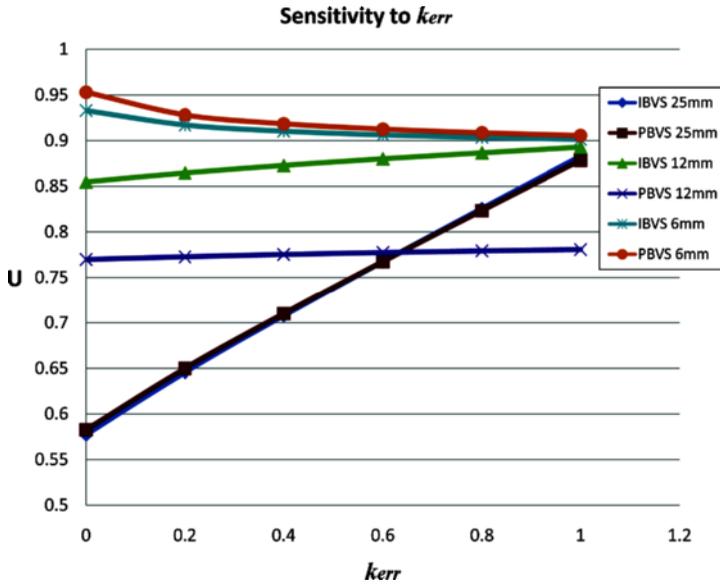


Figure 35. Analysis of sensitivity of MAUF to k_{err} for the refueling design problem.

Sensitivity analysis was performed by varying the values of the gains k_{err} , k_{FOV} and k_{time} . As seen in Figure 32, if $k_{FOV} < 0.325$ keeping the tanker in the field of view is not important, and IBVS with a 25 mm lens is the preferred method, with

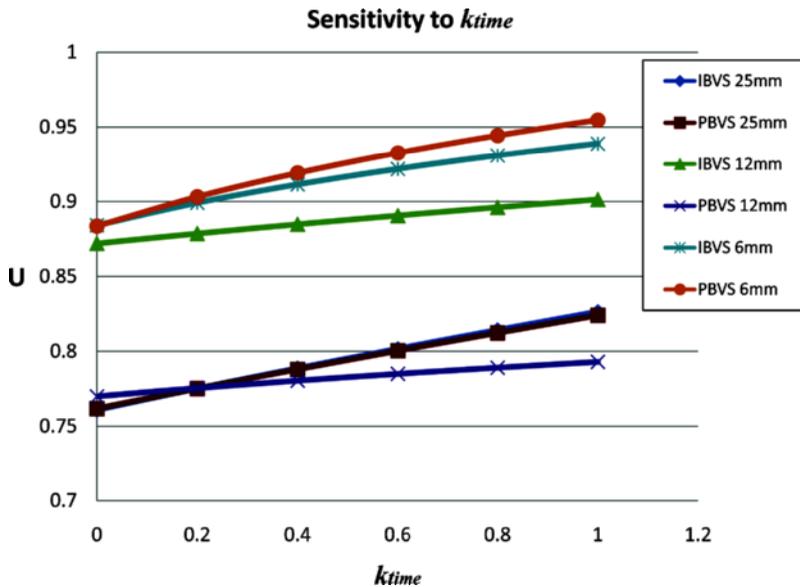


Figure 36. Analysis of sensitivity of MAUF to k_{time} for the refueling design problem.

PBVS/25 mm lens a very close second. If $0.325 < k_{\text{FOV}} < 0.7$, balanced performance is sought and IBVS with a 12 mm lens becomes the best choice. Varying k_{err} or k_{time} while keeping $k_{\text{FOV}} = 0.8$ did not affect the choice of PBVS/6 mm as giving the highest expected utility, which can be seen in Figures 35 and 36.

5. CONCLUSION

We have performed multi-attribute utility analysis in an effort to choose a vision based control system for two different tasks. The first task involved the choice of a visual servo controller for a robot manipulator. Two general systems exist, image-based and position-based. Each one offers strengths and weaknesses, but recent analysis showed that a hybrid system that switched between them showed promise. By biasing the switching toward image-based or position-based, a large number of potential controllers are available, with a wide spectrum of performance. The choice of a switching rule should allow the system to perform according to a system designer's preferences, and MAUA is a natural method to choose the rule.

IBVS is known to perform best in terms of feature point motion in the image, and PBVS is known to perform best in terms of distance-optimal robot motion. Indeed, both of these systems are rated above any hybrid method when the feature point motion or robot motion are the preferred performance metric to improve. The hybrid systems have lower failure rates than IBVS or PBVS, and if the failure rate is the attribute most heavily weighted, they are chosen. PBVS shows better performance (e.g., lower failure rate, less extreme bad performance) than IBVS in the simulations, so was generally preferred to IBVS, and a switched system biased toward PBVS were preferred to other switched systems.

The second task involves a UAV refueling by docking with a fuel drogue attached to a tanker plane. In addition to successfully docking the fuel probe with the drogue as quickly as possible, it is important to keep the tanker and drogue in the camera field of view to avoid collisions. Thus, final position error, time to docking and distance to the goal when the tanker leaves the field of view were chosen as attributes. The choice of camera lens will affect pose error and the camera field of view, so MAUA was run for IBVS and PBVS with the choice of three focal length lenses. The decision maker preferred to keep the tanker in the field of view over the other two attributes, thus PBVS with a wide-angle lens was the preferred method. Sensitivity analysis was performed on both set of results to study the effect of preferences for one attribute over the others.

REFERENCES

- Branicky, M., V. Borkar, and S. Mitter. 1998. A unified framework for hybrid control: Model and optimal control theory. *IEEE Trans. on Automatic Control* 43(1):31–45.
- Brockett, R. W. 1993. Hybrid Models for Motion Control Systems. In *Essays on Control-perspectives in the Theory and its Applications*, eds. L. Trentelman and J. C. Willems, Birkhauser, Boston, pp. 29–53.
- Chaumette, F. 1998. Potential problems stability and convergence in image-based and position-based visual servoing. *Notes in Control and Information Sciences* 237:66–78.

- Chaumette, F. and S. Hutchinson. 2006a. Visual servo control part I: Basic approaches. *IEEE Robotics and Automation Mag.* 13(4):82–90.
- Chaumette, F. and S. Hutchinson. 2006b. Visual servo control part II: Advanced approaches. *IEEE Robotics and Automation Mag.* 14(1):109–118.
- Corke, P. and S. Hutchinson. 2001. A new partitioned approach to image-based visual servo control. *IEEE Trans. Robot. Automat.* 17(4):507–515.
- Deguchi, K. 1998. Optimal motion control for image-based visual servoing by decoupling translation and rotation. *Proceedings of IEEE/RSJ Int. Conf. Intelligent Robots and Systems* 2:705–711.
- DeMenthon, D. and L. S. Davis. 1995. Model-based object pose in 25 lines code. *International Journal of Computer Vision* 15(1–2):123–141.
- Deng, L., F. Janabi-Sharifi, and W. Wilson. 2002. Stability and robustness of visual servoing methods. *Proceedings of IEEE Int. Conf. Robotics and Automation* 2:1604–1609.
- Diller, N. and J. Warmkessel. 2001. Applying multi-attribute utility analysis to architecture research for the terrestrial observer swarm. *Conf. Digital Avionics Systems* 1, 4B6/1–4B6/9.
- Espiau, B., F. Chaumette, and P. Rives. 1992. A new approach to visual servoing in robotics. *IEEE Trans. Robot. Automat.* 8(3):313–326.
- Fang, Y., W. E. Dixon, D. M. Dawson, and P. Chawda. 2005. Homography-based visual servo regulation of mobile robots. *IEEE Trans. Syst. Man, Cybern.* 35(5):1041–1050.
- Faugeras, O. D. and F. Lustman. 1988. Motion and structure from motion in a piecewise planar environment. *Int. J. Pattern Recog. and Artificial Intell.* 2(3):485–508.
- Feddema, J. and O. Mitchell. 1989. Vision-guided servoing with feature-based trajectory generation. *IEEE Trans. Robot. Automat.* 5(5):691–700.
- Gans, N. R. and S. A. Hutchinson. 2002. A switching approach to visual servo control. *Proceedings of IEEE Int. Symp. on Intelligence and Control* 770–776.
- Gans, N. R. and S. A. Hutchinson. 2003. An experimental study of hybrid switched system approaches to visual servoing. *Proceedings of IEEE Int. Conf. Robotics and Automation* 3:3061–3068.
- Gans, N. R. and S. A. Hutchinson. 2007. Stable visual servoing through hybrid switched-system control. *IEEE Trans. Robotics* 23(3):530–540.
- Hafez, A. A., E. Cervera, and C. Jawahar. 2007. Optimizing image and camera trajectories in robot vision control using on-line boosting. *Proceedings of IEEE/RSJ Int. Conf. Intelligent Robots and Systems* 352–357.
- Huang, T. and O. Faugeras. 1989. Some properties of the E matrix in two-view motion estimation. *IEEE Trans. Pattern Anal. Machine Intell.* 11(12):1310–1312.
- Hutchinson, S., G. Hager, and P. Corke. 1996. A tutorial on visual servo control. *IEEE Trans. Robot. Automat.* 12(5):651–670.
- Keeney, R. and H. Raiffa. 1993. *Decisions with multiple objectives*. Cambridge, UK: Cambridge University Press.
- Kelly, R., R. Carelli, O. Nasisi, B. Kuchen, and F. Reyes. 2000. Stable visual servoing camera-in-hand robotic systems. *IEEE/ASME Trans. Mechatron.* 5(1):39–48.
- LaValle, S. M. and J. J. Kuffner. 2000. Rapidly-exploring random trees: Progress and prospects. *Proceedings of Int. Workshop on the Algorithmic Foundations of Robotics* 293–308.
- Liberzon, D. 2003. *Switching in systems and control*. New York: Springer.
- Liberzon, D. and A. Morse. 1999. Basic problems in stability and design of switched systems. *IEEE Control Systems Magazine* 19(5):59–70.
- Longuet-Higgins, H. 1981. A computer algorithm for reconstructing a scene from two projections. *Nature* 293:133–135.
- Malis, E., F. Chaumette, and S. Boudet. 1999. 2-1/2-D visual servoing. *IEEE Trans. Robot. Automat.* 15(2):238–250.

- Martinet, P., J. Gallice, and D. Khadraoui. 1996. Vision-based control law using 3-D visual features. *Proceeding of WAC 96* 3:497–502.
- Merkhofer, M. and R. Keeney. 1987. A multiattribute utility analysis of alternative sites for the disposal of nuclear waste. *Risk Analysis* 7(2):173–194.
- Thurston, D. 1990. Multi-attribute utility analysis in design management. *IEEE Trans. Engineering Management* 37(4):296–301.
- Thurston, D. 1991. A formal method for subjective design evaluation with multiple attributes. *Research in Engineering Design* 3(2):105–122.
- Weiss, L. E., A. C. Sanderson, and C. P. Neuman. 1987. Dynamic sensor-based control robots with visual feedback. *IEEE Trans. Robot. Automat.* 3(5):404–417.
- Zhang, Z. and A. Hanson. 1996. 3-D reconstruction based on homography mapping. *Proceedings of ARPA Image Understanding Workshop Palm Springs CA* 1007–1012.