Chapter 2
Empirical Characterization of Convergence Properties for Kernel-based Visual Servoing

John P. Swensen, Vinutha Kallem, and Noah J. Cowan

Abstract. Visual servoing typically involves separate feature tracking and control processes. Feature tracking remains an art, and is generally treated as independent of the underlying controller. Kernel-based visual servoing (KBVS) is a categorically different approach that eliminates explicit feature tracking. This chapter presents an experimental assessment of the convergence properties (domain of attraction and steady-state error) of the proposed approach. Using smooth weighting functions (the kernels) and Lyapunov theory, we analyze the controllers as they act on images acquired in controlled environments. We ascertain the domain of attraction by finding the largest positive invariant set of the Lyapunov function, inside which its time derivative is negative definite. Our experiments show that KBVS attains a maximum pixel error of one pixel and is commonly on the order of one tenth of a pixel.

2.1 Featureless Visual Servoing

Typically, visual servoing involves tracking image features and controlling a robot based on the motions of these image features. Usually this involves tuning the feature tracking algorithm and controller independently, with no clear notion of how to co-optimize tracking and control. KBVS is distinguished from traditional visual servoing techniques in two respects, namely the lack of explicit feature tracking at each image frame and the inherent combination of tracking and control. These

John P. Swensen and Noah J. Cowan
Department of Mechanical Engineering, Johns Hopkins University,
3400 North Charles Street, Baltimore, MD 21218, USA
e-mail: {jpswensen,ncowan}@jhu.edu

Vinutha Kallem
GRASP Laboratory, University of Pennsylvania, 3330 Walnut Street,
Philadelphia, PA 19104, USA
e-mail: vkallem@seas.upenn.edu

G. Chesi & K. Hashimoto (Eds.): Visual Servoing via Advan. Numer. Methods, LNCIS 401, pp. 23–38.
springerlink.com

characteristics may provide a benefit over traditional position-based and image-based visual servoing [10], 2 1/2 D visual servoing [2], and other advanced visual servo algorithms [1].

"Featureless" visual servoing methods including KBVS take advantage of the rich set of visual data present in images without reducing the image to feature points. This reduces computation because extracting image features usually requires more computation than the image measurements used in featureless methods. Also, this confers robustness to region or feature occlusions in the images.

Nayar *et al.* [15] present one of the earliest works in featureless visual servoing. The authors select patches of the goal image that form a high dimensional measurement. They then generate a training data set of images acquired at known robot poses. The leading principal components of the training images form a low dimensional approximation of the so-called appearance manifold. Images during control are projected onto this manifold and control is performed on this low dimensional space. The sampling of the camera workspace becomes prohibitively time consuming as the number of degrees of freedom increases and has to be evaluated for each scene. Deguchi [6] extends the method to 6 degrees of freedom and automates patch selection to improve the Jacobian. These papers do not address formal guarantees of convergence.

More recently, Tahri and Chaumette [16] use moments for determining camera motions with respect to a goal. They compute the Jacobian relating changes in low-order image moments to the velocity of the moving camera. The image moments provide a similar measurement as the KBVS sampling kernel in that they provide a scalar measurement of the entire image for each type of moment calculation. Formally, with an appropriately designed family of kernels, it may be possible to conceive of moment-based visual servoing as a special case of KBVS. The advantages of KBVS are that selection of kernels with compact (finite) support will minimize edge effects, and moreover the KBVS approach embraces gray scale images. However, the image moment solution provides for 6 degree of freedom visual servoing, a problem not yet solved for KBVS.

A related result by Collewet *et al.* [3] notes that for a static scene, the time-varying image is simply a transformed version of the original image, assuming no occlusions. They develop the Jacobian relating the motion of every pixel in the image to the motion of the camera, thus allowing derivatives of the image signal to be calculated directly. They then use the time derivative of the image to minimize a cost function, which in similar to the Lyapunov function as discussed later in this chapter, if the image is a continuous signal. This approach is analogous to placing a kernel at every pixel.

KBVS presents a new kind of featureless visual servoing and this chapter presents the first in-depth empirical evaluation of KBVS. Section 2.2 presents the KBVS algorithm and its conceptual and theoretical underpinnings. Section 2.3 describes the experiments conducted to evaluate the method and characterize the convergence properties for several degrees of freedom. We conclude by discussing the outstanding issues that need to be addressed to make KBVS robust and to rationalize kernel selection.

2.2 Kernel-based Visual Servoing

To aid the reader in understanding the basics of KBVS, we describe the details of the algorithm in a simple two dimensional case [12] and refer the reader to Kallem *et al.* [13] for the details of the $\mathrm{SO}(3)$ case, roll about the optical axis, translation along the optical axis, and combined motion derivations.

The kernel-projected value in KBVS, derived from the kernel-based tracking literature [4, 9, 8, 7, 17], is a weighted measurement of an image based on the a sampling kernel. A Lyapunov function is formed from the vector of kernel-projected values and the KBVS control input ensures Lyapunov stability, as shown below. A key concept in the demonstration of stability is the equivalence of the kernel-projected value under a change of coordinates: ideally there is no difference between the kernel-projected value under a transformation of the image or the inverse transformation of the kernel. In the two dimensional case discussed below, this transformation is simply translation parallel to the image plane of the camera.

Throughout the analytical derivation of KBVS, several assumptions are made about the scene and the signals that appear in the computation. First, we assume that the image plane is continuous (rather than discrete) and infinite, namely the image plane is a copy of $\mathbb{R}^2$. Second, we assume the scene to be planar. Finally, we require that pixels are constantly illuminated across all image frames.

We assume the robot, as seen in Figure 2.1, is the kinematic plant

$$\dot{\mathbf{x}} = \mathbf{u} \tag{2.1}$$

where the configuration of the robot is described as $\mathbf{x} = [x, y]^T$ and $\mathbf{u} \in \mathbb{R}^2$ drives the robot parallel to the image plane of the camera. Let the image be represented as a

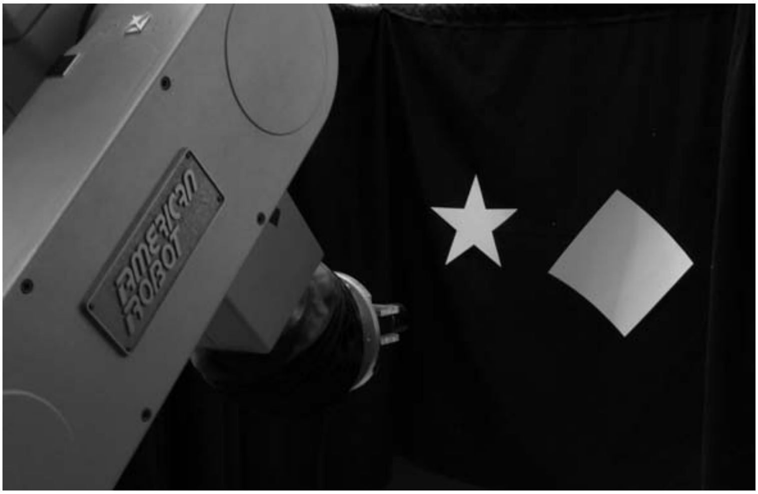


Fig. 2.1 The robot used for the experiments done in this chapter

signal $I(\mathbf{w},\mathbf{x}(t)) \in \mathbb{R}$. The spatial indexing parameter of the 2D image is $\mathbf{w} \in \mathscr{I} = \mathbb{R}^2$. The signal is represented as a function of the spatial indexing and the position which is a function of time. Although one might expect to see the signal written only as a function of time, $I(\mathbf{w},t)$, we will be more explicit and write it as a function of the robot configuration which is in turn a function of time, $I(\mathbf{w},\mathbf{x}(t))$. For the sake of notational simplicity and without loss of generality we assume that the image plane is a unit distance away from the scene.

Using the modified notation, the signal at any translated position is related to the signal at the goal

$$I(\mathbf{w},\mathbf{x}(t)) = I(\mathbf{w}-\mathbf{x}(t),\mathbf{0}) = I_0(\mathbf{w}-\mathbf{x}(t)). \tag{2.2}$$

In the two dimensional case, the kernel-projected value is a function $\xi : \mathscr{I} \to \mathbb{R}^n$ where

$$\begin{aligned}\xi &= \int_{\mathscr{I}} K(\mathbf{w}) I(\mathbf{w},t) d\mathbf{w} \\ &= \int_{\mathscr{I}} K(\mathbf{w}) I_0(\mathbf{w}-\mathbf{x}(t)) d\mathbf{w},\end{aligned} \tag{2.3}$$

and the kernel-projected value at the goal is $\xi_0 = \xi$ at x_0. The dimension n refers to the number of kernels being used such that the kernels represent a function $K : \mathbb{R}^2 \to \mathbb{R}^n$. Note that using the change of coordinates $\bar{\mathbf{w}} = \mathbf{w} - \mathbf{x}(t)$, (2.3) can be written as

$$\xi = \int_{\mathscr{I}} K(\bar{\mathbf{w}} + \mathbf{x}(t)) I_0(\bar{\mathbf{w}}) d\bar{\mathbf{w}}. \tag{2.4}$$

(2.3) and (2.4) show that the the kernel-projected value is the same for the fixed kernel with an image taken after the camera has been moved and a kernel shifted in the opposite direction with the goal image. This can be seen pictorially in Figure 2.2 for a one-dimensional signal and kernel. The change of coordinates in (2.4) will be used later to allow us to take time derivatives of the kernel-projected value: we can easily design kernels with known and analytically determined derivatives, whereas the time derivatives of the images are unknown.

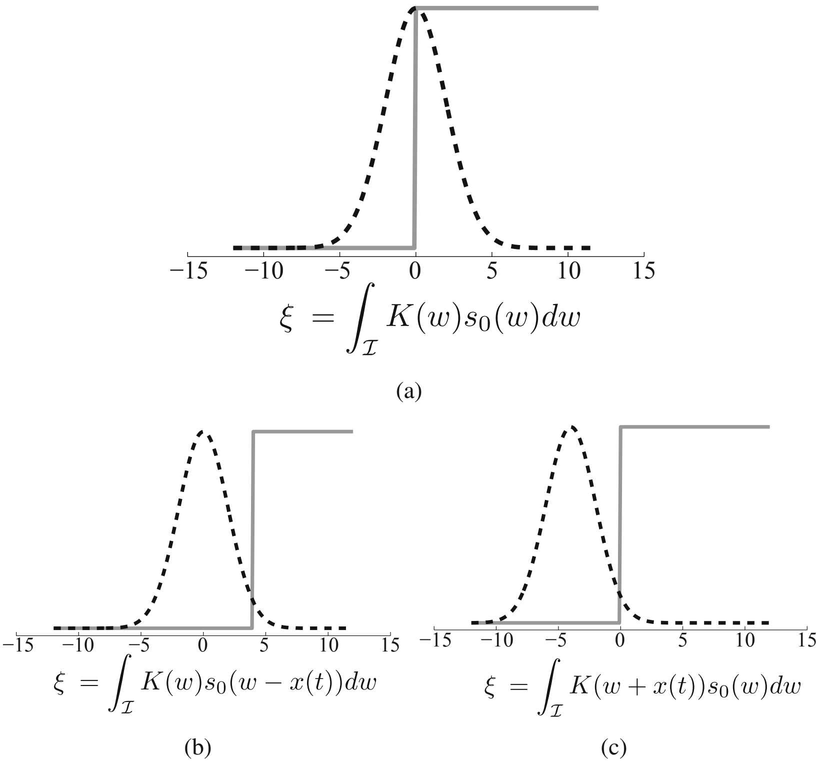
The goal of the KBVS method is to find a control input $\mathbf{u}$, a function of ξ, to move the robot such that

$$\lim_{t\to\infty} \mathbf{x}(t) = \mathbf{x}_0.$$

To find such a control input, we consider

$$V = \frac{1}{2}(\xi - \xi_0)^T P (\xi - \xi_0) \tag{2.5}$$

as a Lyapunov function candidate, where P is any positive $n \times n$ matrix. To choose the control input, $\mathbf{u}$, we analyze the time derivative of the Lyapunov function using the shifted kernel representation of the kernel-projected value from (2.4):



(a)

(b)

(c)

Fig. 2.2 Demonstration of the equivalence of the kernel-projected value whether moving the camera or moving the kernel: (a) a static kernel and the signal at the goal; (b) a static kernel and a shifted signal; and (c) a shifted kernel and the signal at the goal

$$\begin{aligned}\dot{V} &= (\xi - \xi_0)^T P \dot{\xi} \\ &= (\xi - \xi_0)^T P \frac{\partial \xi}{\partial \mathbf{x}} \dot{\mathbf{x}} \\ &= (\xi - \xi_0)^T P \left[\underbrace{\int_{\mathscr{I}} \frac{\partial K(\bar{\mathbf{w}} + \mathbf{x}(t))}{\partial \mathbf{x}} I_0(\bar{\mathbf{w}}, \mathbf{0}) d\mathbf{w}}_{\text{Jacobian } J} \right] \mathbf{u}.\end{aligned} \tag{2.6}$$

After reversing the change of coordinates, (2.6) can be written as

$$\begin{aligned}\dot{V} &= (\xi - \xi_0)^T P \left[\int_{\mathscr{I}} \frac{\partial K(\mathbf{w})}{\partial \mathbf{x}} I_0(\mathbf{w} - \mathbf{x}(t), \mathbf{0}) d\mathbf{w} \right] \mathbf{u} \\ &= (\xi - \xi_0)^T P \left[\int_{\mathscr{I}} \frac{\partial K(\mathbf{w})}{\partial \mathbf{x}} I(\mathbf{w}, \mathbf{x}(t)) d\mathbf{w} \right] \mathbf{u}.\end{aligned} \tag{2.7}$$

If we choose the control input

$$\mathbf{u} = -\left[\int_{\mathscr{I}} \frac{\partial K(\mathbf{w})}{\partial \mathbf{x}} I(\mathbf{w}, \mathbf{x}(t)) d\mathbf{w}\right]^T P^T (\xi - \xi_0) \tag{2.8}$$

the time derivative of V is given by

$$\dot{V} = -\left\| (\xi - \xi_0)^T P \left[\int_{\mathscr{I}} \frac{\partial K(\mathbf{w})}{\partial \mathbf{x}} I(\mathbf{w}, \mathbf{x}(t)) d\mathbf{w}\right] \right\|^2 \leq 0. \tag{2.9}$$

We now have a Lyapunov function that is positive definite in ξ by construction and have shown a control input $\mathbf{u}$ that ensures its time derivative is negative semidefinite in ξ. Asymptotic stability to the goal configuration, however, must be shown in $\mathbf{x}(t)$ (assuming without loss of generality that $\mathbf{x}_0 = 0$). That is, we must show that both V and $\dot{V}$ are positive definite and negative semidefinite in $\mathbf{x}(t)$, respectively. To see this locally, we look at the first order Taylor series expansion of ξ about the point ξ_0, as done for classical visual servoing in [5]:

$$\begin{aligned} \xi &= \xi_0 + \frac{\partial \xi}{\partial \mathbf{x}} \mathbf{x}(t) + O(\mathbf{x}^2) \\ \xi - \xi_0 &= \frac{\partial \xi}{\partial \mathbf{x}} \mathbf{x}(t) + O(\mathbf{x}^2) \\ \xi - \xi_0 &= J\mathbf{x}(t) + O(\mathbf{x}^2). \end{aligned} \tag{2.10}$$

Then, inserting (2.10) into (2.5) and (2.6) we achieve the desired definiteness in $\mathbf{x}(t)$ under certain conditions on the Jacobian matrix J:

$$V = \frac{1}{2}\mathbf{x}(t)^T J^T P J \mathbf{x}(t) + O(\mathbf{x}^3), \quad Q = J^T P J, \tag{2.11}$$

and

$$\dot{V} = -\mathbf{x}(t)^T Q Q^T \mathbf{x}(t) + O(\mathbf{x}^3). \tag{2.12}$$

If the Jacobian matrix $J \in \mathbb{R}^{n \times p}$ is full column rank, where n is the number of kernels and p is the dimension of $\mathbf{x}(t)$, then Q will be a full rank $p \times p$ matrix. From (2.11), (2.12), and a full rank assumption for Q, we can conclude that V is positive definite and $\dot{V}$ is negative definite in some neighborhood of the goal with $V = 0$ and $\dot{V} = 0$ at the goal. The rank condition on Q also clearly sets forth a necessary condition on the number of kernels required; namely that there must be at least as many kernel measurements as there are degrees of freedom.

Although we have only shown the method for the simple 2D case, the reader can infer how it extends to other motions given an appropriate change of coordinates based on the relationship between camera motion and image transformation. Several motions are worked out in great detail in our previous work [12, 11, 13]. These include 2D translation, depth, roll, rigid body rotations, and some combinations of these.

Considering the computation and assumption from above, to practically implement a KBVS controller we must relax several of the assumptions made about the scene and signals. First, the image plane is not continuous. Our kernel-projected value then becomes a discretized summation, rather than a continuous integral:

$$\xi = \int_{\mathscr{I}} K(\mathbf{w}) I(\mathbf{w}(t)) d\mathbf{w} \approx \sum_{w} \bar{K}[w] \bar{I}[w(t)]. \tag{2.13}$$

Additionally, the image plane in not infinite. In (2.13), the domain of the integration would be infinite whereas the domain of the summation extends to the boundaries of the image. We work around this problem by choosing kernels with finite support. That is, the kernels are selected such that the weighting is zero at the boundaries of the image. This allows us to truncate the integration, as the integral over the domain outside the image boundaries will essentially be zero.

Another limitation is the assumption of a planar scene. This is necessary to avoid issues of parallax between near and far objects in the scene. Scene parallax could significantly affect the size of the domain of attraction. In our experimental configurations we have constructed a planar scene. In real world implementations, one can ensure that the kernels and their finite support are wholly contained within an approximately planar object.

Because we are making a weighted measurement on the intensities of each pixel individually, we need to assume that each point in space is providing the same measurement as it moves across the pixels of the camera. This is also known as the brightness constancy constraint. To avoid problems of varying lighting, we normalize the image at each time step to the maximum pixel value.

The first aim of the following sections is to verify the KBVS method empirically, taking into consideration the fact that we are violating the assumptions of the analytical solution. During experimentation, we are using the workarounds for each of the nonideal issues as described above. Additionally, the analysis above only determined that the KBVS control input produces an asymptotically stable system in a neighborhood of the goal configuration, but gave no insight into the size of the neighborhood. The second aim of the empirical validation is to characterize the domain of attraction for each of the degrees of freedom discussed above.

2.3 Empirical Validation

Empirical validation of KBVS is a two step process. The first is to characterize the domain of attraction for a kernel and scene combination. As discussed during the analytical derivation, there are scenes and/or kernel combinations that could result in either a poor domain of attraction or instability. After finding a good set of kernels for the scene, per the Jacobian rank condition, and having ascertained the expected domain of attraction, the second step is to use the expected domain of attraction for initial conditions to a set of experiments.

Subsection 2.3.1 discusses the domain of attraction for several of the combinations of degrees of freedom for which we have an analytical solution as indicated in

Section 2.1 and calculated in [12, 13]. In order to visualize the domain of attraction, we show the level sets of V in most cases and a level surface plus three orthogonal slices of the level surface that pass through the origin for the $\mathrm{SO}(3)$ case. In each figure we also plot the sign of $\dot{V}$ at each point.

Subsection 2.3.2 then uses the results of Section 2.3.1 to show experimental results for a variety of the combinations of the degrees of freedom in the controlled environments. We conduct 50 trials for each type of robot motion by going to the edge of the domain of attraction as determined in Section 2.3.1.

2.3.1 Analysis of the Domain of Attraction

To find the domain of attraction experimentally, we perform the following sequence for each type of robot motion:

1. gather images on an evenly space grid of the camera workspace;
2. compute the value of the Lyapunov function, V, at each position;
3. compute the control input, $\mathbf{u}$, at each position as described in Section 2.2;
4. compute the gradient of the Lyapunov function, ∇V, using a finite central difference approximation at each position;
5. compute the time derivative of the Lyapunov function using the chain rule

$$\dot{V} = \nabla V \dot{\mathbf{x}} = \nabla V \mathbf{u}; \quad \text{and} \tag{2.14}$$

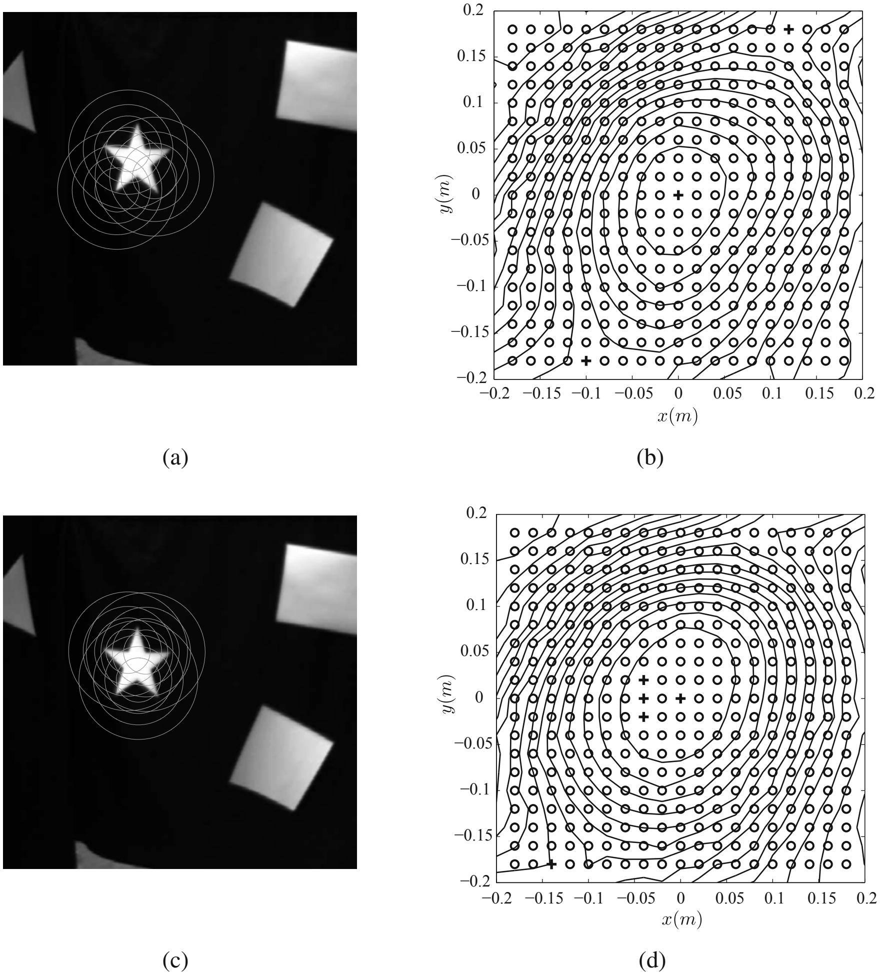
6. search for the level set of V which is homeomorphic to S^n with the largest interior and for which $\dot{V} < 0$ for every point on the interior except the goal position.

By the theory of Lyapunov stability[14], we can conclude that the volume described by the interior of the level set of the Lyapunov function as described above is the domain of attraction for the asymptotically stable system.

2.3.1.1 2D

Figure 3(b) shows the analysis of the domain of attraction for the goal scene and kernels shown in Figure 3(a). The chosen kernels are placed around the star object and provide for a domain of attraction that approaches the joint limits of the robot for planar motions. Although not seen in this figure, kernel selection often results in a Lyapunov bowl that is not symmetrically shaped. The positive definite matrix P from (2.5) can be engineered to change the shape of the Lyapunov function and provide faster convergence in all directions.

In contrast, Figures 3(c) and 3(d) show how a poor selection of kernels unintentionally resulted in a second subset of the domain of the Lyapunov function to which the system would converge. Similar to the successful set of kernels, these kernels were placed at different locations around the star object near the center of the goal image. The combination of scene and kernels resulted in an invariant region for which the robot motion was stable to some point, but not asymptotically stable



(a)

(b)

(c)

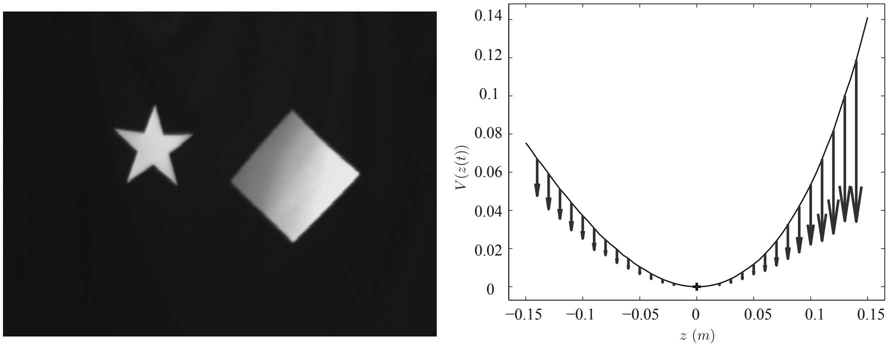
(d)

Fig. 2.3 An analysis of the domain of attraction for good and bad kernel selection: (a) the goal image and the contours of the good kernels; (b) the level curves of V for the good kernels. The $+$ symbols indicate locations at which $\dot{V} \geq 0$ and the $\circ$ symbols indicate locations where $\dot{V} < 0$; (c) the goal image and the contours of the bad kernels; and (d) the level curves of V for the bad kernel with the associated marker for $\dot{V}$

to the goal for all the initial condition within the invariant region. This illustrates the need for a principled approach to kernel selection.

2.3.1.2 Depth

The domain of attraction for the depth motion was as large as the joint limits of the robot allowed. That is, the Lyapunov function consisted of a convex quadratic



(a) (b)

Fig. 2.4 An analysis of the domain of attraction for depth motion: (a) the goal image and the contours of the kernels used; and (b) the plot of V. The arrow indicate the value of $\dot{V}$ at each depth

function of the camera depth along the optical axis for which $\dot{V}$ was negative everywhere except the goal location and zero at the goal, as seen in Figure 2.4.

2.3.1.3 SO(3)

Figure 2.5 shows the analysis of the domain of attraction for the set of kernels shown in Figure 2.6. To determine the domain of attraction, we determined all the orientations of the robot for which our empirical computation of the time derivative of the Lyapunov function was positive. We then start at the level surface $V = c$ for c very large and slowly decrease c until arriving at a level surface where $\dot{V} < 0$ for every point in its interior except at the goal. The volume defined by the interior of the level surface defines the domain of attraction for this 3 degrees of freedom system. Figure 5(a) shows this level surface and nearby points where $\dot{V} > 0$. Figures 5(b), 5(c) and 5(d) show orthogonal slices of the Lyapunov function through the origin. It becomes much easier to visualize the magnitude and shape of the domain of attraction using these slices, taking into account that these may give a inaccurate representation for strangely shaped level surfaces of V.

Figure 2.6 shows the kernels used for determining the domain of attraction and for the experiments shown later. It is often useful to show the kernel overlayed on top of the image at the goal position to give an idea of how the kernel-projected values are generated from the image at the goal. It is important to note that a change in the kernels, whether location or covariance for our Gaussian kernels, carries with it an implicit change in the domain of attraction.

2.3.2 Experimental Results

To validate the domains of attraction determined above, we conducted 50 trials for each of the degrees of freedom analyzed in Section 2.3.1. We randomly set the

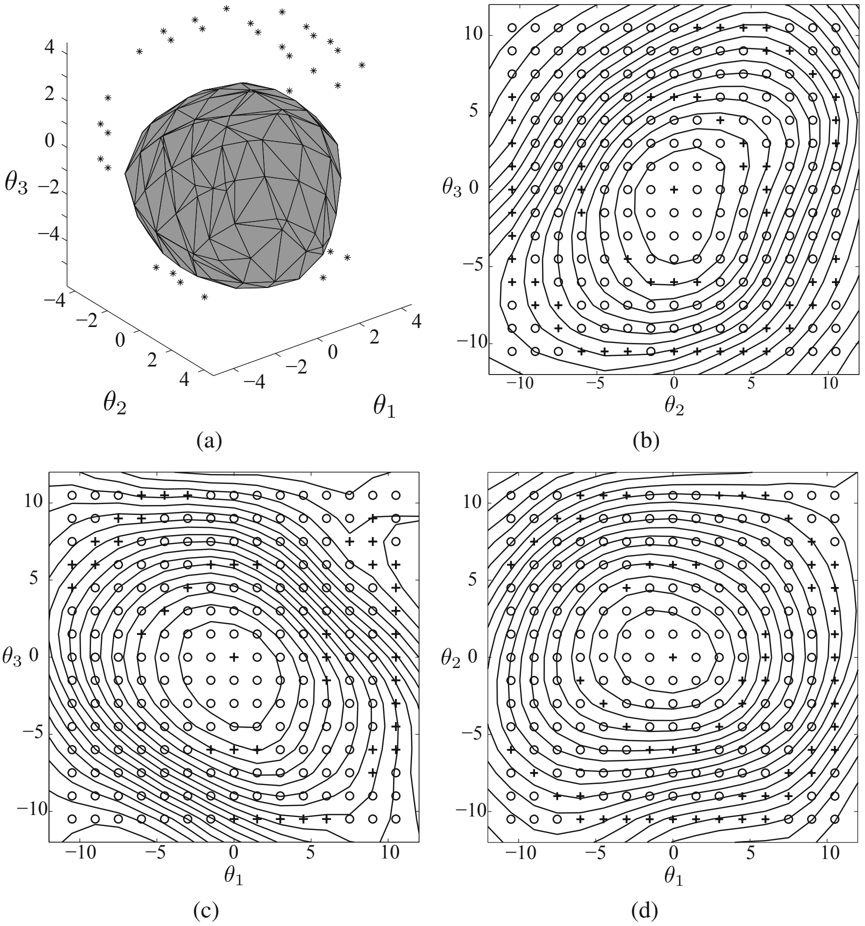
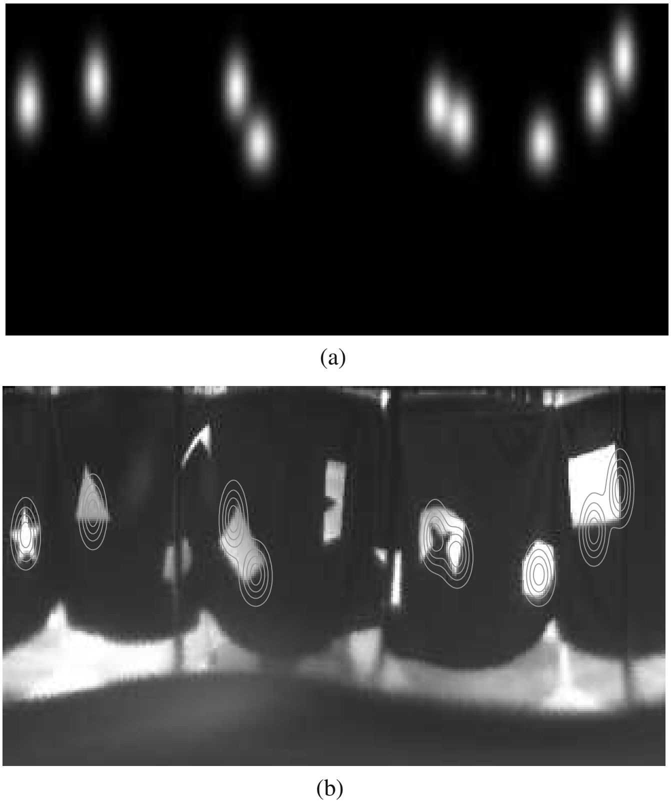


Fig. 2.5 The domain of attraction for rigid body rotations: (a) the volume representing the domain of attraction with nearby points where $\dot{V} > 0$; (b) the slice through the level sets of V through the origin and aligned with the θ_1 axis; (c) the slice aligned with the θ_2 axis; and (d) the slice aligned with the θ_3 axis. The $+$ symbols indicate locations at which $\dot{V} \geq 0$ and the $\circ$ symbols indicate locations where $\dot{V} < 0$

initial displacement of the robot to be near the empirically determined limits of the domain of attraction, except for the case of the rigid body rotations as discussed later. Error measurements were computed using the forward kinematics of the robot to determine the goal pose of the robot, the initial displacement, and the final pose of the robot.

Figure 2.7 shows the camera trajectories of the robot in the (x,y) plane for the 50 experimental trials of 2D motion parallel to the image plane. The statistics concerning average error, standard deviation of error, and maximum and minimum errors is given in Table 2.1. A point of interest is the minimum final error of 90 microns.


(a)

(b)

Fig. 2.6 Useful visualizations for placing kernels. These provide insight into how the kernels interact with an image: (a) the sum of the kernels; and (b) the kernels overlayed on the image at the goal location

This represents an error of 2-5 encoder counts for each of the joint variables of the robot. While the maximum error was 1.5 orders of magnitude larger, there were many trials that were near this minimum error.

Figure 2.8 shows the camera trajectories of the robot along the optical axis for the 50 experimental trials of depth motion. The corresponding error statistics are shown in Table 2.2.

Figure 2.9 shows the rotations about the focus of the parabolic mirror for the 50 experimental trials of rigid body rotations. The error statistics for the angles are found in Table 2.3. The determination of the domain of attraction for the rigid body rotations indicated that the maximum allowable move would be approximately ±5° in each of the angles. However, experimentation showed that the system rarely failed for initial displacements of approximately ±9°. Upon closer inspection of Figure 2.5, it is evident that there is a region of the volume near the 5° radius and the 9° radius where $\dot{V}$ is positive. Figure 2.9 shows one failed trial whose initial

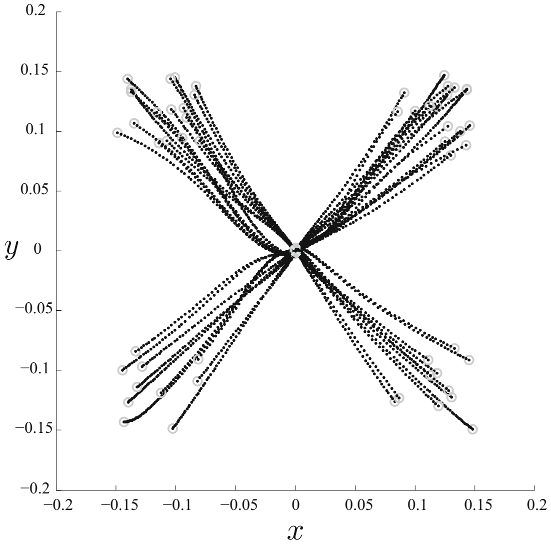


Fig. 2.7 Convergence results for the 50 trials of 2D translations orthogonal to the optical axis

Table 2.1 Error measurements for the fifty 2D experiments

	Average Error	σ Error	Max Error	Min Error
x	0.3347 mm	0.2472 mm	1.0527 mm	4.989×10^{-3} mm
y	0.6857 mm	0.5792 mm	2.4527 mm	9.6609×10^{-3} mm
$\sqrt{x^2+y^2}$	0.9439 mm	0.5267 mm	2.4598 mm	89.5382×10^{-3} mm

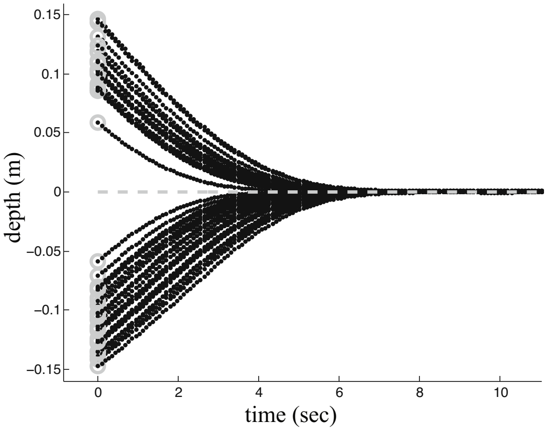


Fig. 2.8 Convergence results for the 50 trials of translation parallel to the optical axis

displacement was at the 9° radius, which was not included in the calculation of error statistics.

2.3.3 Final Error in Pixels

Although the final errors in configuration space given in Section 2.3.2 seem more that adequate, a more insightful measure of efficacy is the final error in pixels. Because perspective cameras were used, the distance of the objects from the camera play a significant role in how the objects move, and consequently, the magnitude of change in the kernel projected value under camera motions. The number of pixels per unit of motion was determined using the set of collected images from Section 2.3.1 by following a point during motion in each of the degrees of freedom.

For the 2D experiments, the plane containing the object was at a distance such that the relationship between robot motion and image motion was approximately 2 mm/pixel. From Table 2.1 we see that the maximum error was on the order of a pixel and the average error was approximately a half of a pixel.

For the depth experiments, the plane containing the object was at a distance such that the relationship between the robot depth motion and the image motions was approximately 4 mm/pixel. From Table 2.2 we see that the maximum error was about one-third of a pixel and the average error was approximately one-tenth of a pixel.

For the rigid body rotation experiments, the results were very promising in terms of pixel error. Because the image is the result of a spherical projections, the pixel error changes depending on whether it is measured near the center of the image or

Table 2.2 Error measurements for the fifty depth experiments

	Average Error	σ Error	Max Error	Min Error
z	0.3708 mm	0.2864 mm	1.2163 mm	0.0203 mm

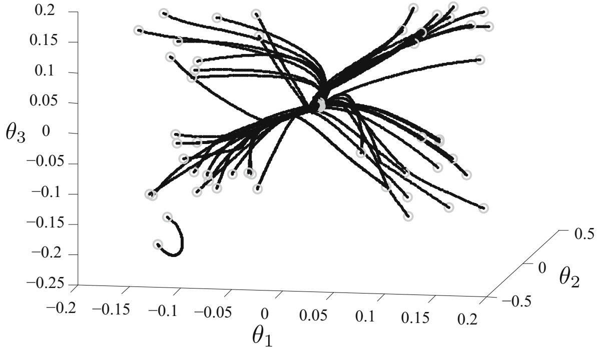


Fig. 2.9 Convergence results for the 50 trials of rigid body rotations

Table 2.3 Error measurements for the fifty rigid body rotation experiments

	Average Error	σ Error	Max Error	Min Error
θ_1	0.0391°	0.0257°	0.1277°	2.2192×10^{-6} °
θ_2	0.0417°	0.0314°	0.1175°	6.18×10^{-4} °
θ_3	0.0474°	0.0463°	0.2649°	0.0017°
$\sqrt{\theta_1^2+\theta_2^2+\theta_3^2}$	0.0853°	0.0446°	0.2781°	0.0151°

near the periphery. Since most of our objects and kernels were between these two extremes, the pixel error was calculated based on pixel motions in this intermediate region. The structure containing the objects was at a distance such that the relationship between the rotations about the focus of the parabolic mirror and the image motions was approximately 0.6°/pixel. Referring to Table 2.3 we see that the maximum error was about one-third of a pixel and the average error was about one-fifth of a pixel.

2.4 Conclusions

The results of the analysis in Section 2.3.1 and experimentation in Section 2.3.2 indicate that relaxing the assumptions required for the analytical treatment of KBVS does not hamper experimental performance. We achieved adequate convergence rate and domain of attraction. Moreover, the KBVS results in very low steady-state error, despite our current *ad hoc* approach to kernel selection.

Robust kernel selection for unstructured environments is an area for future research. Figure 2.3 demonstrated how slight variations in kernel selection could cause the robot to fail to converge to the goal image. Even though both sets of kernel satisfied the rank requirement for the Jacobian between $\dot{\xi}$ and $\dot{\mathbf{x}}$, one choice is clearly superior with respect to the size of the domain of attraction. The ability to optimize kernel selection to maximize the domain of attraction using only the image at the goal is the natural next step.

References

[1] Chaumette, F., Hutchinson, S.: Visual servo control, part ii: Advanced approaches. IEEE Robotics and Automation Magazine 14(1), 109–118 (2007)

[2] Chaumette, F., Malis, E.: 2 1/2 D visual servoing: a possible solution to improving image-based and position based visual servoings. In: IEEE International Conference on Robotics and Automation (2000)

[3] Collewet, C., Marchand, E., Chaumette, F.: Visual servoing set free from image processing. In: IEEE International Conference on Robotics and Automation, pp. 81–86 (2008)

[4] Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(5), 564–575 (2003)

[5] Cowan, N.J.: Vision-based control via navigation functions. Ph.D. thesis, University of Michigan (2001)

[6] Deguchi, K.: A direct interpretation of dynamic images with camera and object motions for vision guided robot control. International Journal of Computer Vision 37(1), 7–20 (2000)

[7] Dewan, M., Hager, G.: Towards optimal kernel-based tracking. In: Computer Vision and Pattern Recognition, vol. 1, pp. 618–625 (2006)

[8] Fan, Z., Wu, Y., Yang, M.: Multiple collaborative kernel tracking. In: Computer Vision and Pattern Recognition (2005)

[9] Hager, G.D., Dewan, M., Stewart, C.V.: Multiple kernel tracking with SSD. In: Computer Vision and Pattern Recognition, vol. 1, pp. 790–797 (2004)

[10] Hutchinson, S., Hager, G.D., Corke, P.I.: A tutorial on visual servo control. IEEE Transactions on Robotics and Automation 12(5), 651–670 (1996)

[11] Kallem, V.: Vision-based control on lie groups with application to needle steering. Ph.D. thesis. Johns Hopkins University (2008)

[12] Kallem, V., Dewan, M., Swensen, J.P., Hager, G.D., Cowan, N.J.: Kernel-based visual servoing. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, San Diego, CA, USA, pp. 1975–1980 (2007)

[13] Kallem, V., Dewan, M., Swensen, J.P., Hager, G.D., Cowan, N.J.: Kernel-based visual servoing: featureless control using spatial sampling functions (in preparation)

[14] Khalil, H.K.: Nonlinear Systems, 2nd edn. Prentice Hall, Englewood Cliffs (2002)

[15] Nayar, S.K., Nene, S.A., Murase, H.: Subspace methods for robot vision. IEEE Transactions on Robotics and Automation 12(5), 750–758 (1996)

[16] Tahri, O., Chaumette, F.: Point-based and region-based image moments for visual servoing of planar objects. IEEE Transactions on Robotics 21(6), 1116–1127 (2005)

[17] Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys 38(4) (2006)