Visual Servoing in Unconventional Environments

Undergraduate Thesis

Bachelor of Technology (with Honors)
in
Computer Science & Engineering

by

Santosh Kumar Divvala
200201024
santosh@students.iiit.ac.in

Adviser: C V Jawahar

Center for Visual Information Technology
International Institute of Information Technology
Hyderabad, India
July 2006

Contents

Chapter Page

1 Visual Servoing in Robotics . . . 1
1.1 Problem of Visual Servoing . . . 2
1.2 Motivation and Problem Statement . . . 2
1.3 Contributions and Overview . . . 3

2 Background and Related Work . . . 4
2.1 Computer Vision fundamentals . . . 4
2.2 Types of Visual Control . . . 5
2.2.1 Image-based Visual Servo Control . . . 5
2.2.2 Position-based Visual Servo Control . . . 6
2.2.3 Hybrid Visual Servo Control . . . 6
2.3 Non-Rigid Object Motion . . . 7

3 Homography-based Visual Servoing . . . 8
3.1 Introduction . . . 8
3.2 Homography-based Visual Control . . . 9
3.2.1 Degenerate Configurations and the Use of Multiple Planes . . . 11
3.3 Visually-guided Optimal Camera Positioning in Piecewise Planar Environments . . . 12
3.3.1 Robust Homography Computation . . . 12
3.3.2 Assignment of weights . . . 13
3.4 Modified Control Design . . . 14
3.4.1 Proposed Control . . . 14
3.5 Experimental Results . . . 16
3.5.1 Analysis of Control Law . . . 17
3.5.2 Application to Planar Scene Reconstruction . . . 17
3.6 Conclusion . . . 18

4 Non-Rigid Visual Servoing . . . 19
4.1 Introduction . . . 19
4.2 Servoing in Presence of Deformations . . . 20
4.2.1 Projection of Images . . . 21
4.3 Non-Rigid Manipulation . . . 22
4.3.1 Manifold Model . . . 22
4.3.2 Projection, Prediction & Reconstruction of appearance using the Manifold . . . 23
4.3.3 Motion Canceling Algorithm . . . 24

4.4 Visual Servoing using Curves 24
4.5 Results and Discussions 25
4.5.1 Analysis of Servoing Algorithm 26
4.5.2 Stability and Robustness of the Servoing Algorithm 27
4.5.3 Analysis of Motion Cancelling Algorithm 28
4.6 Concluding Remarks 29

5 Conclusion 30

Bibliography 32

Abstract

Classical robotic algorithms have been proposed by envisaging robots operating in highly structured environments and cannot be applied to real-life scenarios. Our aim is to develop vision strategies and algorithms that can perform optimally even in unconventional environments. This thesis analyzes two principle problems. Firstly, we study the problem of visual servoing in presence of non-rigid objects and analyze the design of servoing strategies needed to perform optimally even in non-rigid environments. Non-rigid objects undergo a persistent change in their structure which prevents them from being easily manipulated. We also propose a servoing algorithm that is robust to non-rigidity. The algorithm extracts invariant features of the non-rigid object and uses these features in the servoing process. Computational efficiency is achieved by predictively calculating the features rather than computing them at every instant. Next, we develop a new vision-based control for positioning a camera with respect to an unknown non-planar object. We introduce a novel homography-based approach that integrates information from multiple planes to estimate the relative displacement of the camera. This approach is robust to image measurement errors and provides a stable estimate of the camera motion that is free from degeneracies in the entire task space. Another contribution of this work is the development of a new control formulation that guarantees the contradictory requirements of producing an optimal camera trajectory and ensuring object visibility by using the homography relating the two views. Experimental results validate the efficiency and robustness of our approaches and demonstrate their applicability.

Chapter 1

Visual Servoing in Robotics

Robot Motion and Robotic Vision are the challenging problems in Robotics research. The task in Robotic Vision is to analyze the images of the scene taken by the cameras attached to the robot and obtain a physical interpretation of the world around it. The motion problem is more complex. Given the position of an object, how does the robot navigate to reach the object. Traditionally, the visual sensing and the navigation problems have been combined in an open-loop fashion *i.e.*, 'looking' and then 'moving' (See Fig. 1.1). The accuracy of the resulting operation then depends directly on the accuracy of the visual sensor and the robot manipulator. An alternative to increase the accuracy of these subsystems is to use a 'visual-feedback' control loop that will improve the overall accuracy of the system. The use of computer vision techniques to control robotic systems has received great

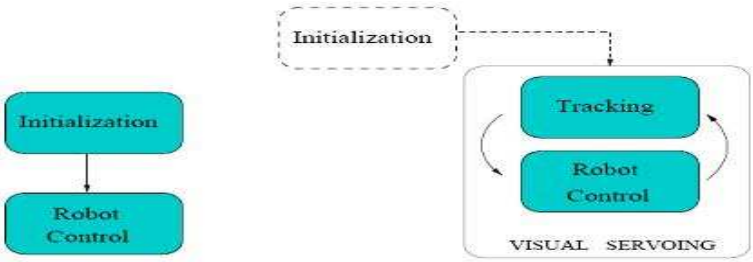

Figure 1.1 Visual Servoing - Feedback-based Systems

popularity in recent times. Images captured by cameras attached to a robot provide ample information about its surroundings that assists it in efficiently navigating the environment. This field, known as 'Visual Servoing', has gained recent prominence due to the widespread availability of high quality cameras and low cost microprocessors. In addition to robotics, visual servoing algorithms also find interesting applications for interactive vision systems such as video conferencing, tracking, active vision, augmented reality etc. The concept of visual servoing has been studied in various forms for more than three decades starting from simple pick-and-place tasks to todays real-time, advanced manipulation of objects. Further, it has has great potential in performing non-repetitive tasks in hazardous environments such as nuclear plants, space, off-shore, underwater.

1.1 Problem of Visual Servoing

The aim of visual servoing is to control a robot using the information provided by a vision system. More precisely, the basic problem to position the robot at a desired position relative to a target object or a set of target features (See Fig. 1.2). It involves the use of one or more cameras and a computer vision system to control the position of the robot's end-effector as required by the task. Many servoing

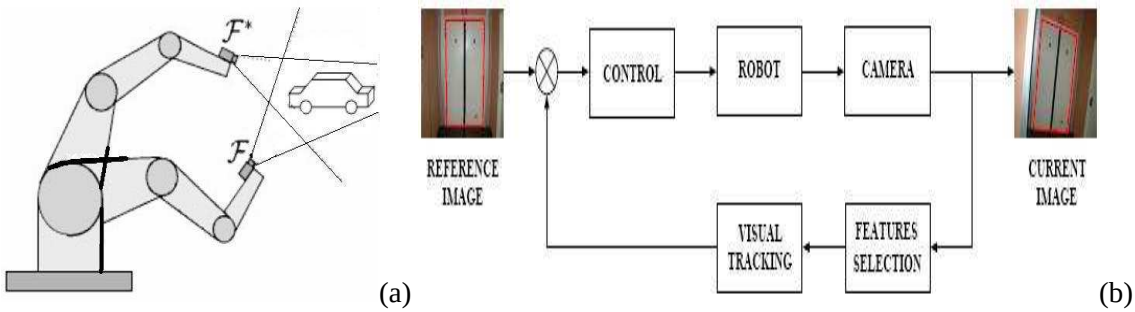

Figure 1.2 Task in visual servoing is to move the robot from its current to a desired position

techniques have been proposed and extensively studied in literature. They can be, in general classified into two categories. Algorithms that use optical flow along with Jacobian-based control to control the camera position constitute the class of *Image-based Visual Servoing* algorithms [7] while techniques that employ $3D$ information to regulate the error in camera pose pertain to *Position-based Visual Servoing* [30]. For the relative merits and demerits of the above techniques, the reader may refer to [15]. Recently, a new group of algorithms have been proposed in [20, 23, 29] that exploit a combination of the above methods to estimate the camera displacement between the desired and the current pose. They combine the traditional Jacobian-based control with other techniques to form the class of *Hybrid Visual Servoing* algorithms. These methods yield a decoupled, optimal camera trajectory and possess a large singularity-free task space.

1.2 Motivation and Problem Statement

It may seem very difficult and sophisticated for robots to assist in tasks such as brain surgery; however in these cases, the task is carefully cut out and mainly formed in a way that it is largely positioning and high-precision application for very specialized tools. Whereas in a household, the problem is basically that there is no high precision around and no information about the objects location. Further there will be other complex objects lying around and objects could be deformable such as carpets, cushions etc. To deal with such variable environments poses a great challenge to robotics.

This thesis proposes the utility of visual servoing techniques to solve many of the problems that currently hinder the applications of robots in 'advanced' applications. The precise research question being addressed is:

Can machine vision be used to control robot manipulators in unconventional environments by providing end-point relative sensing?

'Unconventional environments' in the current context refers to the scenarios which occur in our day-to-day life unlike highly structured industrial settings.

The first problem that we tackle is to perform the servoing task in presence of non-rigid objects. Though several visual servoing algorithms [1, 3, 14, 19] have been proposed, much of the research until now focuses only on rigid objects. There have been few attempts towards non-rigid object tracking [8, 17]. As far as we are aware, the problem of visual servoing in presence of non-rigid objects has not been addressed in the literature. The challenge with non-rigidity is that the motion planned based on the image-clues extracted at current time instant might not be valid in the next time instant as the object undergoes a change in its pose. For a deforming object, each feature point transforms differently and thus generates different control signals at the same time instant, thereby destabilizing the motion planning process.

Another equally challenging problem is to estimate the relative robot displacement between its current and desired pose in unknown and non-planar scenes. Existing algorithms suffer with the problems of degeneracies such as planarity of the scene, occlusion etc. Further, they fail to provide the requirements of generating a straight-line trajectory and ensuring object visibility.

1.3 Contributions and Overview

The two major contributions of this work are

- Designing a servoing algorithm that performs the positioning task optimally in presence of non-rigid objects
- Developing a visual control that robustly estimates the relative displacement in non-planar environments

The rest of the thesis is organized as follows. Chapter 2 explains the basic concepts that are required to follow the material presented in later chapters. In chapter 3, the robust homography based control for camera positioning in non-planar environments is presented. Chapter 4 analyzes the non-rigid visual servoing problem and presents the various strategies devised to tackle this problem. Chapter 5 concludes the thesis and discusses the future work to be pursued.

Chapter 2

Background and Related Work

2.1 Computer Vision fundamentals

To avoid the intrusiveness of mechanical and magnetic sensors, it is desirable to use 'touch free' computer vision-based systems. In context of visual systems, the primary concept that needs to be understood is the geometry of the scene being analyzed by the camera. Geometry is used to explain the relation between robot motion in the world and related 'image' motion. In general, three coordinate frames are used to describe the scene viz., task space (on robot), sensor space (on camera) and the $3D$ world. The transformations across frames can be computed by estimating the displacement between the frames.

The position and orientation of the camera in the $3D$ world is given by its 'pose', which is calculated during the 'calibration' step (See Fig. 2.1). It is given by a 11-parameter model consisting of the camera intrinsic and extrinsic parameters. Given the camera pose, the projection of the $3D$ features on to the $2D$ image plane is given by the projection equation as $p = MP$. The parameters extracted from the image features gives the 'feature vector'. Most vision-based control approaches generally use points as visual features. Other features (straight lines, ellipses, contours, etc.) can be also extracted from the images. A

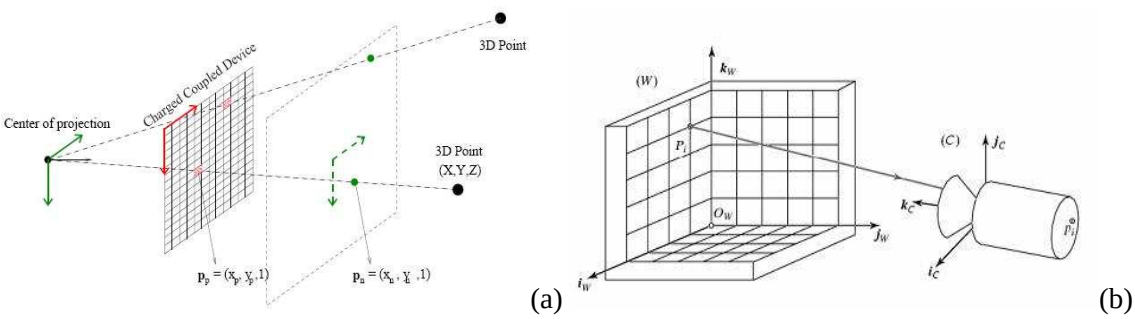

Figure 2.1 Camera projection and calibration

fundamental yet unsolved problem in computer vision is to estimate correspondences between features across images taken by a camera in different views. Fig. 2.2(a) shows the corresponding features of two

scene views. This problem is particularly difficult when the displacement of the camera between the

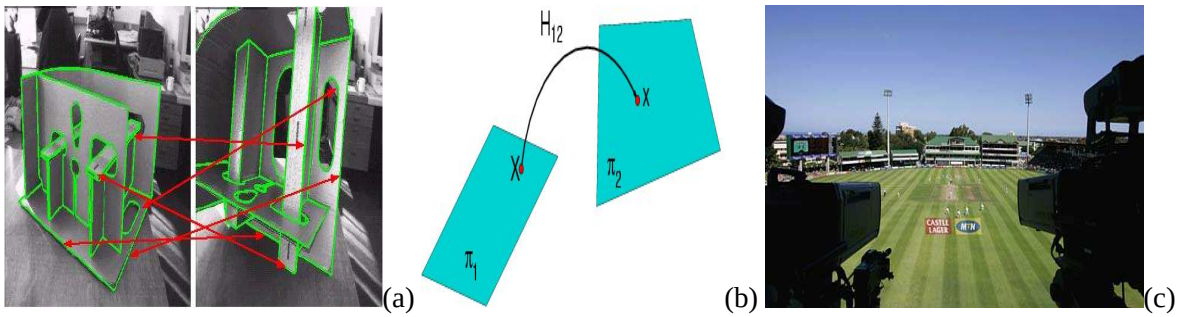

Figure 2.2 Correspondence and Homography estimation

two images is big and when lighting conditions change. Tracking features is similar to the matching problem, however in this case the displacement between the two images is generally smaller. If all the $3D$ points lie on a plane, then the transformation relating two corresponding images is defined by a homography (Fig. 2.2(b)). A good example illustrating the concept of homography is the appearance of advertisements on playgrounds (Fig. 2.2(c)).

2.2 Types of Visual Control

Visual servoing systems are generally classified depending on the number of cameras used (monocular, stereo etc) or on the positions of the camera with respect to the robot (eye-in-hand or eye-to-hand etc) or based on the use of the target knowledge or on the design of the error function to minimize in order to reposition the robot. The general paradigm consists a combination of off-line and on-line steps. The off-line consists of camera/robot calibration, computation of the target features (image set-point), matching target features with observed features while the on-line steps include tracking of image features over time, computation of the error function between current and target location and the computation of the kinematic screw for moving the robot.

Major classification of the controls is based on the use of features in defining the task function. Based on the visual information being utilized, the servoing algorithms can be classified into three categories: position-based ($3D$), image-based ($2D$), or hybrid ($2\frac{1}{2}D$) visual servo control.

2.2.1 Image-based Visual Servo Control

Formally, in Image-based visual servo control, the error signal e is defined directly in terms of image feature parameters as $e(S) = S - S^*$, where S and S^* are the current and the desired feature parameters respectively. The set S may be composed of the coordinates of the points that belong to the object. Other kinds of geometric features like straight line segments, or angles can also be used. By differentiating

this error function with respect to time, we get:

$$\frac{de}{dt} = \frac{dS}{dt} = (\frac{\partial S}{\partial r})\frac{dr}{dt} = L_S V, \tag{2.1}$$

where λ is a gain factor that depends on the kinematic constraints of the robot end-effector and L_S^+ is the pseudo-inverse of the interaction matrix. It relates the motion of features in the image space to the camera motion in the Cartesian space. Assuming a perspective projection model with unit focal length, the interaction matrix L_{S_i} for each point feature (u_i, v_i) in the image is given by [15]

$$\begin{bmatrix} -\frac{1}{Z_i} & 0 & \frac{u_i}{Z} & u_i v_i & -(1+{u_i}^2) & v_i \\ 0 & -\frac{1}{Z_i} & \frac{v_i}{Z_i} & 1+{v_i}^2 & -u_i v_i & -u_i \end{bmatrix}, \tag{2.2}$$

where $i = 1, ..., N$, and Z_i is the depth of the point in the camera coordinate frame. The interaction matrix L_S for the complete set of N points is

$$L_S = \begin{bmatrix} L_{S_1} \\ \vdots \\ L_{S_N} \end{bmatrix}, \tag{2.3}$$

where L_{S_i} is the interaction matrix given by (2.2). Note that in case of a moving object, the expression for V will get modified as

$$V = -\lambda L_S^+ e(S) - L_S^+ \widehat{\frac{\partial e}{\partial t}}, \tag{2.4}$$

where $\widehat{\frac{\partial e}{\partial t}}$ represents the object motion model (see [15]). The final motion instruction to the robot joint controller is obtained by transforming the velocity command from the Cartesian space to the joint space using the robot Jacobian [28].

2.2.2 Position-based Visual Servo Control

The $3D$ visual servoing is also called position-based visual-servoing [30] since the control law uses directly the error on the position of the camera. Depending on the number of visual features available in the image one can compute the position error using or not the model of the target. The main advantage of this approach is that it directly controls the camera trajectory in Cartesian space. However, since there is no control in the image, the image features used in the pose estimation may leave the image (especially if the robot or the camera are coarsely calibrated), which thus leads to servoing failure. Also note that, if the camera is coarse calibrated, or if errors exist in the 3D model of the target, the current and desired camera poses will not be accurately estimated.

2.2.3 Hybrid Visual Servo Control

Recently, Malis et al. [21] proposed a hybrid control scheme (called $2\frac{1}{2}D$ visual servoing) that avoids the respective drawbacks of $2D$ and $3D$ controls. Contrary to position-based visual control, the hybrid

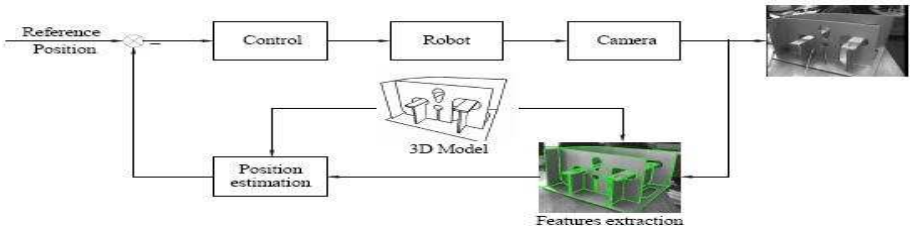

Figure 2.3 Position-based Visual Servo Control

control does not need any geometric 3D model of the object. Furthermore and contrary to image-based visual control, the approach ensures the convergence of the control law in the whole task space. This control is based on the estimation of the partial camera displacement from the current to the desired camera poses at each iteration of the control law. Visual features and data extracted from the partial displacement are used to design a decoupled control law controlling the six camera dof [21].

2.3 Non-Rigid Object Motion

Non-rigid motion encompasses wide range of possible motions (See Fig. 2.4); hence, a 'general' representation for all kinds of motion is difficult and impossible. In literature, most of the methods analyzing non-rigid objects assume the availability of explicit models such as parametric (snakes, super quadrics etc) or physically-based (dynamics represented using set of Lagrangian equations) or hybrid models. In Fig. 2.4, 'elastic' motion represents the most general class of non-rigid motion. It is a

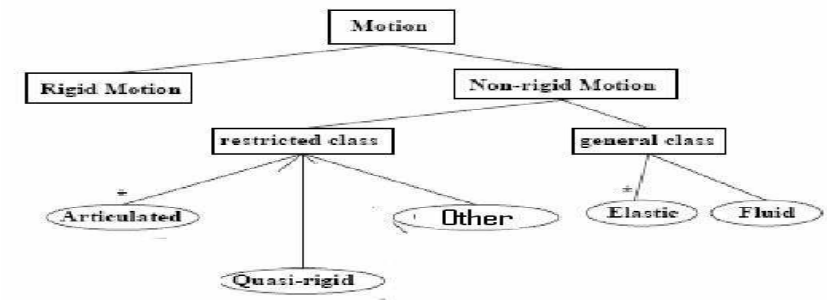

Figure 2.4 Non-Rigid Objects

challenging problem to handle. 'Other' non-rigid motion pertains to motion such as conformal and isometric motions that are in general impractical in nature. Though there have been a few attempts to tackle 'articulated' and 'quasi-rigid' motions, the elastic motion has not been studied in literature.

Chapter 3

Homography-based Visual Servoing

This chapter presents a new vision-based control for positioning a camera with respect to an unknown piecewise planar object. We introduce a novel homography-based approach that integrates information from multiple planes to estimate the relative displacement of the camera. This approach is robust to image measurement errors and provides a stable estimate of the camera motion that is free from degeneracies in the entire task space. We also develop a new control formulation that guarantees the contradictory requirements of producing an optimal camera trajectory and ensuring object visibility by using the homography relating the two views. Experimental results validate the efficiency and robustness of our approach. Further as a consequence of the positioning task, the $3D$ reconstruction of the scene is also obtained thus increasing the application and versatility of this approach.

3.1 Introduction

Hybrid algorithms (Sect. 2.2.3) can essentially be classified into two primary categories. Algorithms pertaining to the first category are generally based on the computation of the essential matrix relating the two camera views [4,23]. Although the relative camera displacement can be obtained even for unknown (non-planar) scenes, a serious problem with epipolar geometry is that, it degenerates in certain critical cases (for example, when the target is planar or when the relative displacement is a pure rotation) and hence is unsuitable for servoing. Alternatively, one can also determine the relative displacement by computing the homography induced by a scene plane relating the two camera views. However, a major drawback of these methods is the implicit assumption of the planarity of the scene, which prevents their application to real world scenarios as the world is often made up of non-planar regions. It must be emphasized that in either cases the degeneracies critically affect the convergence of the system resulting in an unpredictable behavior of the overall control scheme. Thus dealing with such degeneracies is of vital importance in the design of a stable system.

In summary, the desirable characteristics of a *hybrid* visual-control algorithm are

- Absence of degeneracies in its entire task space
- Applicability to both planar and non-planar environments

- Robustness to image measurement errors
- Continuity in velocity instruction and perfect convergence behavior
- Independence from prior knowledge of the object model
- Independence from initialization of parameters

In this chapter, we propose a new hybrid homography-based servoing algorithm that achieves the above features. Our method integrates homographies induced by multiple scene planes using geometric and subspace constraints to efficiently estimate the motion and structure parameters (Fig. 3.1). Another contribution is the development of a modified control law that provides the complementary characteristics of producing an optimal camera trajectory and ensuring object visibility by only using the homography transformation relating the two camera poses. Finally, we demonstrate application of our approach to planar scene reconstruction where the $3D$ structure is recovered directly using its images.

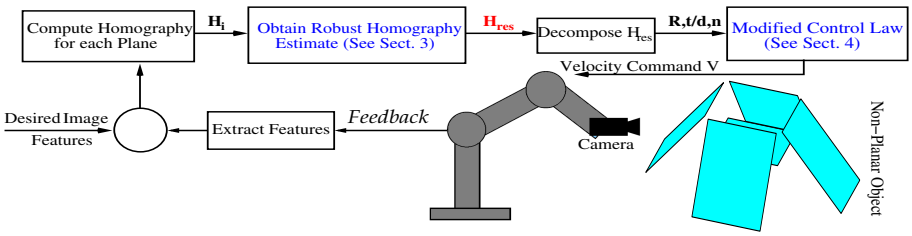

Figure 3.1 Visual-feedback control: Multiple homographies are integrated to obtain a robust homography, which is used in the modified control law to gain optimal performance

The rest of the chapter is outlined as follows. After reviewing the necessary background, we proceed to Section 3.4 where a robust control for single plane scenarios is developed. It is concluded by highlighting the severity of the degenerate configurations and the need for using information from multiple planes. Section 3.3 describes the proposed approach for obtaining a robust estimate of the motion and structure parameters and also validates the claims of the algorithm. Experimental results displaying the robustness of our method and comparisons with existing approaches are reported in Section 3.5. Section 3.6 concludes the paper.

3.2 Homography-based Visual Control

A visual servo control compares the current image of a target with the desired image and the difference (or 'error') is used to drive the camera towards the goal position. Very often, system defects such as calibration errors etc. affect the control performance. Hence the demand is for a robust control that is tolerant to such errors. However, often the task is not just to regulate the image error but also to ensure a realizable camera trajectory. In such scenarios, homography-based control acts as a convenient

option as it regulates the error in camera pose by estimating the $3D$ motion parameters only using image information.

If all the object points lie on a $3D$ plane, their coordinates in the current image I and the goal image I^* are related by a 'collineation' [13]. Assume that a point P lies on a plane whose normal vector is n as shown in Fig. 3.2. The point expressed in current camera frame $\mathcal{F}$ is related to goal camera frame $\mathcal{F}^*$

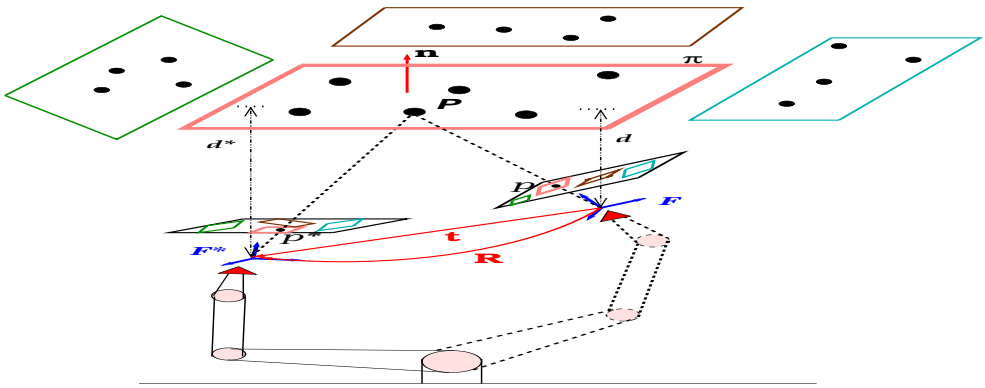

Figure 3.2 Homography-based Visual Servoing

by a rotation matrix R and translation vector t as

$$P^* = R\,P + t = (R + t\frac{n^T}{d})P, \tag{3.1}$$

where $d = n^T P$ is the distance of the plane from the current camera center. Assuming the camera intrinsic parameters are known, the image coordinates of the $3D$ points are given by $p = \frac{P}{Z}$ and $p^* = \frac{P^*}{Z^*}$ respectively. This transforms (3.1) to

$$\frac{Z^*}{Z}p^* = (R + t\frac{n^T}{d})p, \tag{3.2}$$

which can be rewritten as $\alpha p^* = Hp$ where $H_{3\times3} = R + t\frac{n^T}{d}$ is called the 'homography' matrix up to a scale factor α [13]. Given four or more feature correspondences, the matrix H for the normalized α can be determined.

The recovered homography can be decomposed to obtain the rotation matrix R, the translation vector $\frac{t}{d}$ and the plane normal n using the procedure described in [10]. Unfortunately, in the most general case the decomposition of H yields four different solutions (two of them being the 'opposites' of the other). They can be reduced to two solutions by applying the visibility constraint (*i.e.*, all the points must lie in front of the camera). Further ambiguity can be resolved by decomposing an additional homography induced by another scene plane. Two pairs of solutions (S_1, S_2) and $(S_1^{'}, S_2^{'})$ are obtained respectively and a compatible pair $(S_i, S_j^{'})$ among them is found, *i.e.*, a pair with common motion $(R, \frac{t}{d})$. In general, there is only one compatible pair, and hence the unique solution can be obtained. Thus

using information from multiple planes, H can be decomposed unambiguously to obtain the motion and structure parameters. These parameters are used in the control law to generate the optimal velocity instruction.

3.2.1 Degenerate Configurations and the Use of Multiple Planes

Some of the limitations of the existing *hybrid* techniques to estimate relative camera displacement were reviewed in Sect. 3.1. Recently, another method was proposed by Malis *et al.* [20] to estimate the relative orientation between the two camera views for a non-planar object using the concept of 'virtual parallax' [5]. By defining a plane using three arbitrary points on the object, they estimate the homography using this virtual plane and perform the positioning task.

A critical problem with the use of arbitrary planes is that the control is affected by degenerate configurations. Degeneracies in the task space can result either due to occlusion of the feature points, the camera center approaching the world (virtual) plane, the camera centers and the feature points arriving in a singular configuration [13] or due to singular homographies. In either of the cases, when a degeneracy occurs, the plane in consideration is switched *i.e.*, the points used to define the virtual plane are changed and a new plane using three different points is defined. This switching causes a discontinuity in the velocity command and leads to the instability of the control system. In Fig. 3.3, the effect of switching is demonstrated. A positioning task with respect to a non-planar object was simulated. In cases when a degeneracy occurred (occlusion and singular homographies were considered), the virtual plane was switched resulting in a discontinuity.

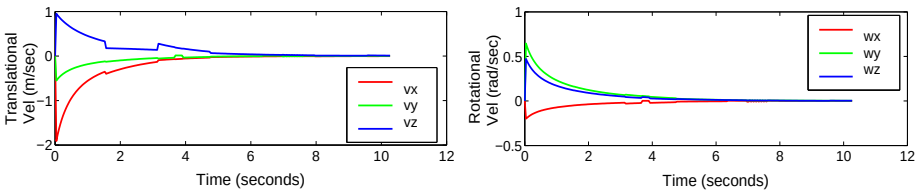

Figure 3.3 Velocity Screw using virtual parallax algorithm: Discontinuities are due to switching of planes during a degeneracy

The other drawbacks in defining a non-planar object using arbitrary planes include

- **Effect of image measurement errors** Homography estimation is highly sensitive to measurement errors in the correspondences. By choosing a different set of points (that are error-free) to define the world (virtual) plane, one can always obtain better results.

- **Unfavorable for planar scenes** The methods using virtual parallax are theoretically inefficient to deal with planar objects as the epipolar geometry degenerates in this case [20].

- **Initialization of plane parameters** In order to resolve the ambiguity in homography decomposition, *apriori* information about the normal vector of the virtual plane is required.

- **Assumption of point features** Point correspondences are not available in many situations or could be noisy. Since the virtual plane is defined explicitly using the non-coplanar points on the object, these methods may not be applicable when such features are not available.

It must be emphasized that the above limitations are caused by the fact that only information from a single plane is being utilized to perform the positioning task. The bottleneck has been the fact that there exists no single homography relating the two camera views that can be absolutely relied upon. Nevertheless, by selectively exploiting the information available from multiple planes, one can avoid all the above drawbacks and achieve superior performance.

3.3 Visually-guided Optimal Camera Positioning in Piecewise Planar Environments

The objective of the servoing task is to drive the disparity between the current and the desired camera configurations to zero. The homographies relating the two camera poses induced by different planar regions are used to guide the positioning task.

Our approach proceeds initially by partially tessellating the non-planar scene into piecewise planar patches. This is done by a simple partitioning of the image features into homogeneous planar regions (See Fig. 3.6(a)). Interest regions are detected and the regions subject to planarity constraint form a set of matching regions [11]. The seed regions act as a 'driver' to guide the evolution of planar patches in the image. Any interest region detector with the ability to detect robust and stable regions can be employed here. For each pair of matching regions, a plane-induced homography is calculated.

Even though a single homography is sufficient to determine the motion parameters (*rigidity constraint*), information from multiple homographies can be combined to obtain a reliable estimate of the camera displacement. However, to avoid the estimation of multiple homographies at each instant, the constraints on homographies can be exploited to reduce the computations. Recall from (3.2) that any H induced by a $3D$ scene plane is described by $H_{3\times3} \approx R + tn^T$. Given a homography matrix H_π of some $3D$ plane π, all other homographies H_i can be described as $\lambda_i H_\pi + tn_i^T$ for a fixed pair of cameras [13]. This observation results from the fact that all the homographies differ only in their scale λ_i and plane n_i parameters. Consider k homography matrices $H_1, H_2, \ldots, H_k$, each expressed as a column vector in a $9 \times k$ matrix. The rank of this matrix is known to be utmost four [25]. Hence the space of all homographies between two fixed camera views is embedded in a 4-dimensional linear subspace of $\Re^9$. This observation follows the fundamental fact that multiple planar patches in the scene share the common global camera geometry (*i.e.*, R, t).

3.3.1 Robust Homography Computation

Given the rank-4 constraint, any new homography can be computed as a weighted combination of four linearly independent homographies. The four homographies are in general selected such that they

are induced by planes that possess largest area and best visibility since they are the most reliable (If the centroid of the features in a planar region is within a threshold distance from the nearest image boundary, then it satisfies the visibility constraint). If the selected four homographies are not linearly independent, then other planes are chosen so that the rank constraint is preserved. This selection is to be made only once initially and at times a new homography has to be chosen *i.e.*, when one of them degenerates.

The resultant homography H_{res} is defined as

$$H_{res} = \lambda_1 H_1 + \lambda_2 H_2 + \lambda_3 H_3 + \lambda_4 H_4, \tag{3.3}$$

where the weights λ_i are assigned such that good homographies receive higher weights while the degenerate or errored estimates are given low priority. By appropriately choosing the λ_i's, the best homography can be deduced. Recall that, in general, any homography in the subspace can be expressed as a linear combination of four base homographies. In our case, H_{res} is one such 'valid' homography possessing certain desired characteristics. It can be visualized as a homography induced by a virtual plane in the scene (analogous to Virtual Parallax [5]).

The principle behind the weight assignment is to prefer valid homographies and reject singular ones in order to prevent abrupt switching of planes during a degeneracy. It must be emphasized that most of the degeneracies are not arbitrary changes and in general, can be predicted in advance. For instance, distance between a camera and a (virtual) plane gradually regresses to zero. Likewise, occlusion of planes can be anticipated by the persistent decrease in area of the planar region (or the number of features). Other degenerate cases can also be predicted in a similar manner and thus homographies that are likely to confront a degenerate configuration can be rejected.

3.3.2 Assignment of weights

Let us define the constraints to assign the weights and hence the parameter λ_i that is used in the computation of the optimal homography.

- **Re-projection Error** This constraint measures the accuracy of the estimated homography. A high error in re-projection indicates a poor estimate and such H should receive less weight as parameters obtained from them will be quite unreliable. Thus the weights are set inversely proportional to the re-projection error. This ensures that planar regions that are affected by the cumulative tracking errors (*'drift error'*) are avoided and thereby guaranteeing the robustness of H_{res} to image measurement errors. The exact weight λ_i^e is defined by first calculating the reprojection error *i.e.*, $e = \sum_k d(p_k^*, Hp_k) = \sum_k ||\frac{p_k^*}{||p_k^*||} - \frac{H*p_k}{||H*p_k||}||$ and then assigning it using a one-sided normal distribution $N(e_{thres}, \sigma_e)$ where e_{thres} is the tolerable reprojection error and σ_e is the variance.

- **Homography Determinant** This quantity signifies the 'goodness' of a homography estimate. If determinant D is tending toward zero, it suggests the arrival of a degeneracy and hence such a homography should acquire low weight. Therefore the weights are set directly proportional to the value of the determinant. This constraint ascertains the resultant homography to be free

of degeneracies. Here again, the weights λ_i^D are set using a one-sided normal density function $N(D_{thres}, \sigma_D)$ where D_{thres} is the minimum acceptable determinant.

- **Area of the Plane** Occlusion of a plane can be detected by measuring the gradient of the plane area dA. If the area of the planar region is decreasing drastically, then it indicates a possible occlusion of this plane in the near future. Thus the λ_i's are to be set inversely proportional to the value of dA. More precisely, the weight λ_i^{dA} is set using a Gaussian distribution $N(dA_{thres}, \sigma_{dA})$ where dA_{thres} is the minimum acceptable gradient.

The resultant weight is computed as a function of the above constraints *i.e.,* $\lambda_i = f(\lambda_i^e, \lambda_i^D, \lambda_i^{dA})$ and the final expression for H_{res} is calculated as

$$H_{res} = \sum_{i=1}^{4} \lambda_i H_i, \quad where \sum_i \lambda_i = 1.$$

Hence a judicious assignment of weights using the above constraints helps in deducing a 'virtual' homography with the ideal characteristics. A change of bases might be required in case one of the H_i degenerate. However, the degenerate homography would automatically procure a low λ value and its replacement does not affect the stability of the system. The parameters obtained from decomposition of H_{res} are used in the modified control law to compute the optimal camera trajectory.

3.4 Modified Control Design

Given the stable estimate of the motion and structure parameters, our focus is to design a robust control that not only produces an optimal camera trajectory but also guarantees feature visibility. Classical approaches such as the $3D$ control algorithms compute an optimal camera trajectory but very often violate the visibility criteria, whereas $2D$ controls ensure the features to remain in the camera field of view, although they suffer from non-optimal trajectory, computational complexity of calculating the Jacobian pseudo-inverse and the demand for $3D$ depth estimates. Note that providing the contradictory requirements of either controls pose a daunting challenge in designing an optimum control scheme. Though a few attempts in this direction have been made [9, 20, 29], the devised controls do not satisfy all of the above requirements.

Much of the information that is required for performing the positioning task is readily available from the homography transformation. The presence of multiple planes in the scene further compliments this fact. We exploit this result to fulfill the requirements of the desired optimal control.

3.4.1 Proposed Control

We first introduce the Cartesian ($3D$) control law and then proceed to derive the robust control. Given the parameters obtained from homography decomposition, the translational velocity to go directly to the goal is determined as $\nu = -\lambda_v t = -\lambda_v(\frac{t}{\hat{d}})\ \hat{d}$, where λ_v is a gain factor and $\hat{d}$ is an estimate of

the distance to the plane (See Fig. 3.2). The rotational velocity is computed as $\omega = -\lambda_\omega \mathbf{u}\theta$, where λ_ω is again a gain factor and u, θ denote the rotation axis and angle that are obtained using the Rodriguez formula for the rotation matrix R as $\theta = \arccos(\frac{1}{2}(tr(R) - 1))$ and $[\mathbf{u}]_\times = \frac{R - R^T}{2\ \mathsf{sinc}(\theta)}$ [30].

However, a direct control in the Cartesian space might result in the features leaving the camera field of view. To enforce the visibility constraint, we use a single image point to control two axes of rotation (around x and y) and the final axis of rotation is controlled directly using the rotation matrix. This is done as follows: We know from the image-based visual servoing control [15]

$$\begin{bmatrix} u - u^* \\ v - v^* \end{bmatrix}_{2\times 1} = \underbrace{\begin{bmatrix} -\frac{1}{Z} & 0 & \frac{u}{Z} \\ 0 & -\frac{1}{Z} & \frac{v}{Z} \end{bmatrix}}_{L_\nu} \underbrace{\begin{matrix} uv & -(1+u^2) \\ 1+v^2 & -uv \end{matrix}}_{L_{\omega_{xy}}} \underbrace{\begin{matrix} v \\ -u \end{matrix}}_{L_{\omega_z}} \Bigg]_{2\times 6} \begin{bmatrix} \nu_{3\times 1} \\ \omega_{3\times 1} \end{bmatrix}_{6\times 1}, \quad (3.4)$$

where $p = [u\ v\ 1]^T = [x\ \ 1]^T$, $p^* = [u^*\ v^*\ 1]^T = [x^*\ \ 1]^T$, $Z = \mathbf{Z}(P)$ (See Fig. 3.2) and $[\nu\ \ \omega]^T$ denotes the camera velocity. (3.4) relates the motion of image features *i.e.*, $x - x^*$ to the camera motion using the 2×6 Jacobian matrix L. It can be rewritten as $x - x^* = [L_\nu\ L_{\omega_{xy}}\ L_{\omega_z}][\nu\ \omega_{xy}\ \omega_z]^T$. Observe that a simple rearrangement of terms yields

$$\omega_{xy} = L_{\omega_{xy}}^{-1}[(x - x^*) - L_\nu \nu - L_{\omega_z}\omega_z], \quad (3.5)$$

where $\nu = (\frac{t}{d})\hat{d}$ and $\omega_z = \mathbf{u_z}\theta$. The rotational motion ω_{xy} is controlled not only to minimize the differences between the current and the goal image features but also to compensate the effects caused by translation on the image. This ensures a straight-line feature trajectory in the image and thereby guarantees object visibility. Estimates of the values Z and d are required in (3.4) that can be obtained as follows: Firstly, observe that

$$det(H) = det(R + \frac{tn^T}{d}) = det(R + \frac{t(n^{*T}R)}{d}) \quad (3.6)$$

$$= det(I + \frac{tn^{*T}}{d})det(R) = \frac{d + n^{*T}t}{d} \quad (3.7)$$

where (3.6) uses the fact that $n^* = Rn$ (See Fig. 3.2). Equation (3.7) can be further simplified using the result $d^* - d = n^{*T}P^* - n^T P = n^{*T}(P^* - RP) = n^{*T}t$. Hence we have $\hat{d} = \frac{\hat{d}^*}{det(H)}$. Using (3.7), Z can be calculated as

$$\frac{Z}{d^*} = \frac{Z}{d^*}\frac{d}{n^T P} = \frac{1}{n^T p}\frac{1}{det(H)}. \quad (3.8)$$

Thus we have $\hat{Z} = \frac{\hat{d}^*}{n^T p}\frac{1}{det(H)}$, where $\hat{d}^*$ is an estimate of the constant distance to the plane in the desired camera frame. In general, this quantity is considered as a gain ratio [20] and a coarse estimate obtained from a simple stereo technique is adequate. Consequently, all the parameters required for the control are now available directly from the homography decomposition.

In summary, the resultant expression for the velocity $\mathbf{v}$ is given as

$$\mathbf{v} = \begin{bmatrix} -\lambda_v I_{3\times 3} & 0_{3\times 2} & 0_{3\times 1} \\ 0_{2\times 3} & -\lambda_{\omega_{xy}} I_{2\times 2} & 0_{2\times 1} \\ 0_{1\times 3} & 0_{1\times 2} & -\lambda_{\omega_z} I_{1\times 1} \end{bmatrix} \begin{bmatrix} \nu \\ \omega_{xy} \\ \omega_z \end{bmatrix} \left(= \begin{bmatrix} (\frac{t}{d})\ \hat{d} \\ L_{\omega_{xy}}^{-1}[(x - x^*) - L_\nu \nu - L_{\omega_z}\omega_z] \\ \mathbf{u_z}\theta \end{bmatrix} \right)$$

The only singularity that occurs is at $Z = 0$ (See expression for L_ν). However, as discussed in the earlier section, this degenerate configuration is avoided by the robust homography computation algorithm. Thus by incorporating image features into the $3D$ control, we have developed an ideal control that offers the complimentary features of object visibility and optimal trajectory.

3.5 Experimental Results

In our experiments, we constructed an arbitrary configuration of planes as shown in Fig. 3.1. The projection of points onto the images were considered as features. A perspective camera projection model was assumed. The basic implementation of the proposed algorithm given below was used to perform the positioning task.

1. Extract features from the current image and partition them into piecewise planar regions
2. Compute homography H_i induced by each region
3. Select four independent homographies induced by the regions that have the largest areas and best visibility (Only the selected regions need to be tracked in the successive iterations)
4. Determine the weights using the geometric constraints and compute the normalized weight λ_i for the selected homographies (Sect. 3.3.1)
5. Determine the optimal homography H_{res} using (3.3)
6. Decompose H_{res} to obtain the motion and structure parameters (Resolve ambiguity using an additional homography)
7. Use the control law to obtain the velocity instruction $\mathbf{v}$ (3.9)
8. Repeat above steps until convergence

We analyzed the performance of our algorithm by generating several random initial camera configurations and then moving the camera to a fixed desired pose in a multi-plane scenario as shown in Fig. 3.1. Observe that a camera can frequently encounter degenerate cases during the positioning task in such a scene. However, in almost all the cases, the proposed algorithm was uninfluenced by degeneracies. In Fig. 3.4, the velocity command generated by the proposed approach for the particular scenario as tested in Fig. 3.3 is shown. It is evident from the continuous velocity screw that presence of degeneracies does not affect the convergence of the algorithm. In addition, this method was unaffected by the errors in image measurements as homographies estimated from noisy correspondences were rejected. Note that the expression for Z in (3.8) requires at least one feature p belonging to the planar region. However as a 'virtual' homography is used in our case, it does not refer to any physical plane in the scene. In our method, we obtained this feature by finding the intersection of the virtual plane with other scene planes [16].

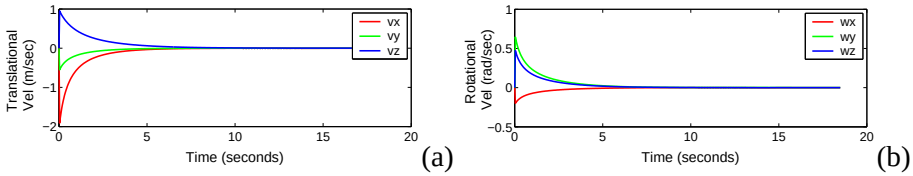

Figure 3.4 Velocity Screw using proposed algorithm: Smooth convergence behavior even in presence of degeneracies (unlike Fig. 3.3)

3.5.1 Analysis of Control Law

We analyzed the performance of the control law in simulation. Fig. 3.5(a) and 3.5(b) show the velocity screw (*i.e.*, the translational and rotational components) and the feature trajectory obtained during a positioning task using the proposed control. The velocity screw obtained in this case is very similar to the one obtained using 3D control (Fig. 3.5(c)). Further, the feature trajectory almost follows a straight line (unlike in Fig. 3.5(d)). These two observations validate our claims of decoupled straight-line camera trajectory and object visibility using the proposed control.

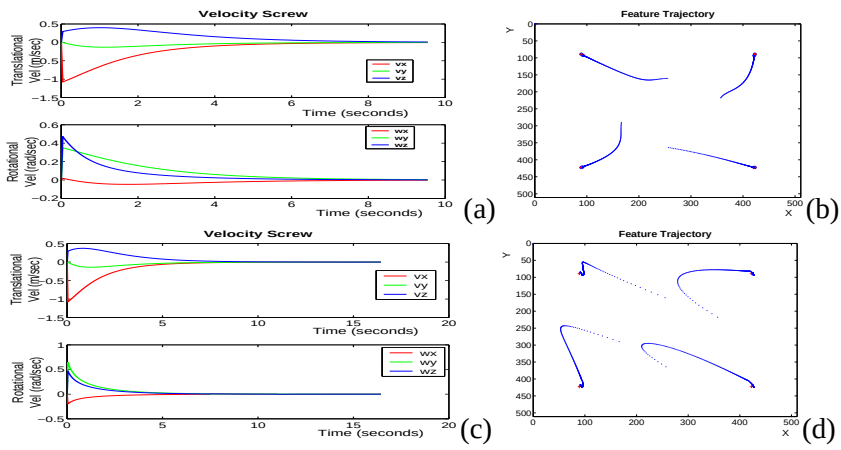

Figure 3.5 Analysis of proposed (a,b) and $3D$ (c,d) control: Similarity of velocity screws in (a) & (c) demonstrates optimal trajectory behavior while near straight-line trajectory in (b) confirms object visibility by the proposed control

3.5.2 Application to Planar Scene Reconstruction

Inter-image homographies are an interesting tool for reconstruction of planar surfaces. By utilizing the availability of motion and structure parameters from the homography decomposition, one can not only compute the desired camera trajectory but also obtain the $3D$ plane parameters required to reconstruct the scene. This task is similar to the visual SLAM and Structure from Motion problems where the concept is to not only obtain the camera motion but also simultaneously build the $3D$ model.

Recall that during the servoing process, the interest regions are detected and are grown in the successive iterations. This refinement step not only improves the current estimate of the planar regions

but also the associated homographies. Using the completely grown regions, the final homographies are computed and decomposed to recover the structure parameters (*i.e.*, the plane normals) that are required to perform the reconstruction. By considering a common feature belonging to two planes n_i and n, a relation could be derived between their distances using (3.8) as

$$Z = \frac{d_i}{n_i^T p}\frac{1}{det(H)} = \frac{d}{n^T p}\frac{1}{det(H)} \quad i.e., \quad d_i = \frac{n_i^T p}{n^T p} d, \tag{3.9}$$

where p denotes the common image feature. Thus given the plane normals n_i and the relative distances d_i, the $3D$ scene could be reconstructed upto a scale factor. Given an estimate of d, the exact scene can be reconstructed. In Fig. 3.6(b), the result of this method applied to the scene in Fig. 3.6(a) is presented. A similar technique to perform planar scene reconstruction is discussed in [16].

(a)

(b)

Figure 3.6 Planar scene reconstruction using inter-image homographies

Our motivation to pursue this work is the curiosity to extend multi-plane homography-based methods into a unified framework wherein the camera motion and scene structure could be learned overtime and be utilized with minimal apriori knowledge for improving the overall performance of the servoing system. Further, by utilizing the additional information available about the camera motion during the positioning task, we plan to improve the reconstruction performance.

3.6 Conclusion

A novel homography-based control capable of positioning a camera even in presence of non-planar objects has been developed for the first time. An optimal homography was efficiently computed using the multiple homographies by employing geometric and subspace constraints. This homography estimate was used in a modified control law to compute the optimal camera trajectory. The method performed better in comparison to existing servoing algorithms and avoided their critical drawbacks. In future, we plan to investigate further the utility of multi-plane homography-based formulations for efficiently solving other classical computer vision problems.

Chapter 4

Non-Rigid Visual Servoing

4.1 Introduction

Many of the living forms that we come across everyday are non-rigid [2] in nature. For example, the articulated human motion, the deforming nature of flying birds and moving aquatic animals, the near-rigid nature of the human face etc. If robots are to be used in presence of such objects, we need visual servoing algorithms that are adaptable to non-rigidity. Apart from applications in *surgical robotics*, such algorithms are also needed in the field of *deep-sea robotics*. During underwater exploration, robots need to calibrate themselves with respect to aquatic animals and plants, which undergo deformations in their structure due to the quasi-periodic wave motion. The work done by [14] and [19] deals with only moving object visual servoing. But to achieve positioning tasks in above situations, we need algorithms that can not only tackle global motion but are also robust to deformations. Recently in [6], Cavusoglu *et al.* proposed a method to handle non-rigidness by canceling the relative motion between the non-rigid object (heart) and the camera by using a model predictive controller so as to facilitate easy manipulation of the object. In this paper, we address the issue of positioning of the end-effector of a robot with respect to a non-rigid object. Often, the positioning task is a pre-requisite for the manipulation task.

The problem of non-rigid visual servoing is different from that of servoing with respect to a moving object, where all the features that are being tracked undergo the same or predictable transformations in the image space. Whereas, for a deforming object, each feature point transforms differently and thus generates different control signals at the same time instant, thereby destabilizing the motion planning process.

As the non-rigid object undergoes a continuous change in pose, it is preferable to extract certain pose-invariant features and use them in the servoing process. These invariant features describe the overall stationary characteristics of the non-rigid object. By using such features, we reduce the non-rigid object to a rigid object, which enables us to use the conventional servoing algorithms even in this case .As a first attempt towards solving this problem, we propose a simple yet an effective visual servoing algorithm to handle deformations. Our method is based on a not so strict assumption that any non-rigid motion can be approximated to a generalized quasi-periodic motion. And for a quasi-periodic motion, gross measures such as the moving average are sufficient to reveal its general stationary characteristics

provided that the change in pose can be compensated. By using the recent results in multiple-view geometry, we efficiently extract such static features for the non-rigid object, which enable us to use the conventional visual servoing techniques even in this case.

Though non-rigid objects have not been studied in the visual servoing framework, they have been well analyzed in computer vision in the context of non-rigid tracking and $3D$ reconstruction problems. Most of these algorithms employ models like the *linear subspace model (manifolds)* [26], *kinematic chains* [8] etc in order to maintain a representation of the non-rigid object and also to infer its internal motion model. Such models allow prediction of the future states of the object, given its current state. Prediction is necessary in case of manipulation tasks where the end-effector needs to interact with the deforming object. For the positioning task, only an approximate pose of the object is sufficient to align the camera with respect to the object. Any error resulting due to this approximation can be corrected by using the visual feedback.

The task of servoing in presence of non-rigid motion can be classified into two primary sub-tasks. The first task involves moving the camera (attached to the robot) from an initial position to a desired position with respect to a non-rigid object. Next, the camera needs to be predictively positioned with respect to the deforming object so that the robot end-effector can easily interact with it. We refer to the former task as *positioning* while the latter as *manipulation*.

The rest of the chapter is outlined as follows. The proposed algorithm for robot positioning is described in Section 2. In Sect. 3, the algorithm for canceling the relative motion between the non-rigid object and the robot is presented. Experimental results are reported in Section 4 and Section 5 concludes the chapter.

4.2 Servoing in Presence of Deformations

Our problem of visual servoing is to position the end-effector of the robot with respect to a deforming object. The algorithm, which computes an optimal path for the robot to traverse from the initial position to the desired position, can be divided into two basic steps. After deciding on an appropriate feature to represent the non-rigid object, the following steps need to be repeated until the desired position is reached.

- Compute the error estimate between the current and the desired feature vector i.e., $S - S^*$
- Using a proportional control law (see Sect. 2.2.1), move in the direction that minimizes the above error

It is crucial to have a good feature vector S to represent the object state and thereby to perform the servoing task. However for non-rigid objects as the object pose continuously changes, we cannot obtain a feature vector from a single image that characterizes its state as it might lead to an unstable end-effector trajectory. For instance, a surgical robot positioning itself only using the current image of the heart will oscillate corresponding to the systole and diastole states of the heart during its motion.

Moreover, if only a single image is used to describe the desired position of the end-effector (S^*), it will lead to oscillations of the manipulator even after the desired position is reached. One possible solution to circumvent this problem would be to use the sequence of images representing all states of the object to perform the servoing task. This will generate better control signals as it takes into account the object deformations. However, an efficient solution is to extract certain invariant features from those sequence of images by exploiting the stationary signal property of the quasi-periodic deformations of the non-rigid object. The invariant features provide a lower dimensional representation of the non-rigid object which uniquely defines the object state. Such features have been used in the domain of object recognition e.g., cross ratios, eigen features etc. Such features describe the general class of non-rigid motion where the transformation is due to shifting and scaling.

At each pose, a sequence of images $(I_1, I_2, ..., I_n)$ are acquired by the camera, capturing the deformations of the non-rigid object (where n depends on the periodicity of the deformations). Then, m point features are extracted from each image I_i to obtain a feature vector $F_i^t = [f_1 f_2 ... f_m]^T$, where each feature f_k corresponds to the projection of a $3D$ point on the surface of the non-rigid object at time instant t. The invariant feature G^t corresponding to the image sequence is computed as a function of all the feature vectors.

$$G^t = f(F_1^t, F_2^t, ..., F_n^t) \tag{4.1}$$

However, the camera can use its previous knowledge and predict the next gross appearance of the deforming object. Thus at each position, only a single image is captured by the camera. The rest of the deformations of the object are obtained by projecting the past images into the new pose. Hence, the gross feature at time t can be predictively calculated as a function of the features extracted from previous images and the current image as

$$G^t = f(g(F^{t-n}, ..., F^{t-2}, F^{t-1}), F^t), \tag{4.2}$$

where F^{t-i} denotes the feature vector obtained from the camera pose r_{t-i}, F^t denotes the feature vector extracted from the current image and $g(\cdot)$ is a function that projects the past image features into the current view as explained below.

4.2.1 Projection of Images

Given two camera matrices M_1 and M_2 and an image I_1 taken from the pose corresponding to M_1, the task is to obtain the image I_2 as seen from the pose corresponding to M_2 using a partial 3D model of the world that is assumed to be available in the image-based visual servoing process [15]. Let $M_1 = K_1[R_1\ t_1]$ and $M_2 = K_2[R_2\ t_2]$ where K_i denotes the intrinsic parameter matrix and $[R_i\ t_i]$ denotes the pose (rotation and translation). Let p_1 represent a point in I_1 and Z_1 be an estimate of the depth of the corresponding $3D$ point P_1 in the camera frame given by the $3D$ model. The coordinates of point P_1 are given by $P_1 = K_1^{-1} Z_1 p_1$ (see [13]).

Now to obtain the 3D coordinate of this point in the other camera frame, we have

$$P_2 = [R_2\ t_2][R_1\ t_1]^{-1} P_1. \tag{4.3}$$

Finally, the image coordinates of the point with respect to the second camera are given by $p_2 = \frac{1}{Z_2} K_2 P_2$. Thus by projecting the past image features into the new view, all the deformations of the object viewed from the current pose of the camera can be obtained. These features are used to compute the representative feature of the non-rigid object.

4.3 Non-Rigid Manipulation

Non-rigid objects [2] undergoes a persistent change in their structure which prevents them from being easily manipulated. To employ robots to interact with such objects, we need to design algorithms that can anticipate the future deformations of the object and position the robot appropriately. Such scenarios are encountered in non-rigid object grasping [18], manipulation [27], motion following etc. However, if the relative motion between the object and the camera can be nullified, then the manipulation task could be easily performed. In fact, in most of the above scenarios, performing the manipulation task essentially boils down to canceling the relative motion between the non-rigid object and the robot which facilitates better interaction between them.

Our algorithm proceeds by constructing a model of the non-rigid object over time and uses this model to predict the future appearances of the object. Given the current and the future appearance, an image-based visual servoing algorithm [15] is employed to move the camera from its current pose to the desired pose with respect to the object. Any resultant error during this step is corrected by the visual feedback. By repeatedly performing the above steps, the camera is predictively positioned such that it observes the same appearance of the object over time. This effectively cancels the relative motion between the robot and the non-rigid object.

4.3.1 Manifold Model

The first stage of the algorithm is to build a model of the non-rigid object. This is a crucial step as a model is needed not only to represent the object deformations but also to infer its internal motion. Here we use PCA for modeling the subspace of appearances of the object. Appearance, in the current context, refers to a configuration of point features on the target extracted from the image (See Fig. 4.1).

The manifold model was first proposed by Nayar et al. in [22]. Here, we consider a modified formulation of their method. The first step of our method involves computation of the eigen features from the given image features. Given a set of features extracted from images taken from some camera pose r (where each image represent a deformations of the object), we rearrange them in a vector form $x_1^{'}, x_2^{'}, ..., x_M^{'}$ to get $2N$ dimensional vectors (where N denotes the number of point features and M denotes the number of images). The vectors are normalized and then stacked as columns of a $2N$ x M data matrix $A = [x_1, x_2, ..., x_M]$. The eigen values of the covariance matrix AA^T are computed and the eigen vectors (e_i) corresponding to the top k $(\frac{\Sigma_{i=1}^{k}\lambda_i}{\Sigma_{i=1}^{M}\lambda_i} > \Theta)$ eigen values are chosen as the basis of the linear subspace.

Each feature vector x_i is then projected on to the eigen-space by first normalizing it and then finding its dot product with each of the eigenvectors (basis) of the subspace. By projecting all the feature vectors, we obtain a set of discrete points in the eigen-space. Since consecutive images are correlated, their projections in the eigen space are close to one another. The discrete points $g(p)$ describe a smoothly varying manifold as shown in Fig. 4.1. Note that the above procedure generates a discrete manifold. To obtain a continuous one, we interpolate between the points using a standard cubic-spline algorithm.

4.3.2 Projection, Prediction & Reconstruction of appearance using the Manifold

Given a test image, the first task is to identify its resemblance in the manifold. This is done by extracting the features from the image and then projecting it onto the manifold. Due to image noise, z (the corresponding eigen feature of the input image) may not lie exactly on the manifold. Therefore, we find a point p' that gives the minimum distance d between the manifold $g(p)$ and z i.e., $d(p') = min||z - g(p')||$. If $d(p')$ is within some threshold, we conclude that the image is of the point p' (See Fig. 4.1). Once the point p' corresponding to the test image is determined, we can predict the next

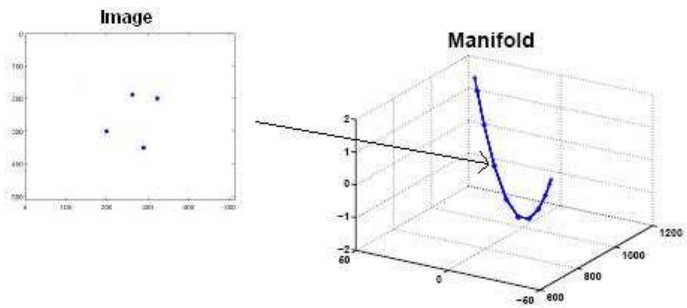

Figure 4.1 Projection of image onto the manifold (a) Four points describe the image appearance (b) Manifold constructed for the non-rigid object

appearance of the object using the knowledge of the manifold. As the appearance of the object varies smoothly, consecutive images will be projected close to one another on the manifold. Hence the point adjacent to the projected point will correspond to the next appearance of the object (p_{new}). By using the basis vectors of the eigen-space, the image corresponding to the predicted point can be reconstructed as

$$I = \sum_{i=1}^{k} p_{new}(k) * e_k. \tag{4.4}$$

If the image is taken from a different camera pose, then the eigen feature z (corresponding to this test image) would not get projected near to its corresponding point on the manifold as the manifold was constructed using images taken from a different camera view. To circumvent this problem, we construct the manifold from a single reference camera pose and then map images from any arbitrary camera pose

to this reference pose. Once such a mapping is defined, the transformed image can be projected onto the manifold and the next appearance of the object could be obtained. One possible mapping function is discussed in 4.2.1. Using this function, we can transform the test image taken from the current camera pose to the reference pose. This transformed image can now be projected onto the manifold and its next image can be predicted and reconstructed. The reconstructed image is then transformed back to the current camera pose using the similar procedure as described above. Thus, we obtain the future appearance of the object given the current image captured from an arbitrary camera pose without explicitly maintaining the manifolds for every view.

4.3.3 Motion Canceling Algorithm

Our aim is to cancel the relative motion between the non-rigid object and the robot end-effector so as to facilitate better manipulation of the object by the robot. We assume that the non-rigid object exhibits some pattern in its deformations and does not change its structure arbitrarily. This is a mild assumption as most of the real world non-rigid objects exhibit some periodicity in their motion. Hence, by observing an object for a finite amount of time, all its deformations can be captured and can be modeled using a manifold.

In order to cancel the relative motion, we need to predictively position the camera such that it observes the same appearance of the object in every frame. At the end of the positioning task [24], the camera reaches the desired pose corresponding to the object appearance that has to be maintained during the manipulation task. The image observed at this camera pose is captured and projected onto the manifold. The next appearance is predicted and then reconstructed as described in section 4.3.2. Given the current image and the predicted image, the image-based visual servoing algorithm is employed to move the camera from its current position to the desired position from where the camera would observe the same appearance of the object at the next instant. Thus, by repeatedly performing the above steps, the desired appearance of the object can be maintained over time. Any error that results, either during the prediction of the next appearance of the object (section 4.3.2) or during the positioning of the camera (section 2.2.1), can be corrected by using the visual feedback.

4.4 Visual Servoing using Curves

A better solution to solve the positioning problem is to model the motion of the features. Consider a configuration of n points. These points can be considered as the ‘interest’ points located in the image where the object changes two-dimensionally. Each point is moving independently along a (complex) curve. The term ‘independently’ implies that every point has a different trajectory. These curves are projected onto the image (at the current and the desired camera pose) i.e., trajectory of each point (3D motion) now resembles a curve or contour in the image. Note that this contour describes the object motion rather than object shape.

Let the contour be described by parameters α (similar to a point being described by its coordinates x, y). These parameters are used to define the feature vector for performing the servoing task. The challenge now is to come up with a good representation/parameterization for the curves. A good parameterization better describes the curve and gives precise information about the object motion. Further, these parameters should be obtained predictively i.e., given only the current (single) frame and the past information, the new contour parameters should be obtained. For achieving this, we use a Kalman filter to predict the curve parameters in the new image. Prediction is necessary in the current scenario as it not only helps in learning about the motion over time but also makes the approach robust to noise. Further, it avoids the critical problem of establishing non-rigid feature correspondences.

An important point to note is that the type of motion is determined by the constraints on the object motion e.g., articulated, quasi-rigid, elastic etc. If no constraints are enforced, then there is no distinction between different types of non-rigid motion. For instance, articulated object has some stationary features on it which can be identified by observing points having similar curve/contour parameters; similarly, the complexity of the curve/contour for elastic objects is simpler when compared to fluid motion.

We first looked at the problem from a Jacobian-based control perspective. Though different features can be used to 'represent' the nonrigid motion, a new 'representation' demands deriving the new Jacobian relating the camera motion to the feature motion. Some literature is available for deriving the Jacobian in case of image contours or curves. However in the current case, the contour represents the object motion rather than its shape. Deriving the Jacobian for such features is a tedious task. A better and a more efficient method is to use the homography-based approach.

In this approach, the motion of features is modeled as planar curves. A well-known theorem of Helmoltz has shown that the most general motion of a sufficiently small element of a deformable object can be represented as a sum of translation, rotation and an extension (or contraction) in three mutually orthogonal directions. Further, in presence of opaque objects, the visible deformations are those occurring at the object surface. Thus, these deformations can be assumed to occur locally on the plane tangent to the surface. Thus, each point on the object moves along a planar curve though different points move along different planes [12]. This is a very important result as this facilitates us to reduce the non-rigid object to a set of planar motion surfaces. Thus, the homography estimate relating the current and desired views of the curves gives us the relative camera displacement. Further, by using multiple curves, we obtain multiple homography estimates and a robust homography can be computed using these homographies as described in Chap. 3. We are currently investigating this approach.

4.5 Results and Discussions

In our experiments, we consider a set of four $3D$ points on the surface of a non-rigid object moving with a linear velocity of 2 cm/s. A perspective camera projection model is assumed. The projection of the points onto the image (of size 512×512) are considered as features. The deformations of the non-rigid object are simulated by using a sinusoidal motion perturbed by a white Gaussian noise of SNR

$10dB$. This makes the object undergo non-uniform quasi-periodic deformations. The task of positioning a camera with respect to the non-rigid object is studied.

4.5.1 Analysis of Servoing Algorithm

The basic implementation of the proposed algorithm can be summarized into the following steps.

1. In an offline step, acquire images from the desired final position. Extract feature S^* from these images using (4.1)
2. Take a sequence of images from the current camera pose and calculate S, again using (4.1)
3. Compute the velocity screw V using (2.4)
4. Using V, move the end-effector to the new pose and acquire a new image
5. Project the old images into the new pose as discussed in Section 4.2
6. Using the old images and the current image, extract S using (4.2)
7. Repeat steps 3 to 6 until $e(S) < \theta$

We demonstrate the results of the proposed algorithm using two kinds of features namely the mean and the eigen feature. In the first method, all the feature vectors $F_i, i = 1, 2, ..., n$ obtained from the n images are averaged to get the mean feature vector M as $\sum_{i=1}^{n} \frac{F_i}{n}$. In this context, the mean operator corresponds to the function $f(\cdot)$ and M represents the invariant feature G^t. This feature is used as the input S to the servoing algorithm. The camera velocity V is estimated as given in (2.4). Using this estimate of V, the camera moves to the new position and again grabs the new image. This process is repeated until the desired pose (as specified by S^*) is reached. As explained in Section 4.2, the new mean is predictively calculated using the past image features as

$$M^t = \frac{(g(M^{t-1}) * n - g(F^{t-n})) + F^t}{n}. \tag{4.5}$$

In the second method, the eigen feature of the sequence of image features is computed using the PCA formulation [26]. Normalized feature vectors $F_i^{'}, i = 1, 2, ..., n$ are stacked as columns into a data matrix i.e., $A = [F_1^{'} F_2^{'} ... F_n^{'}]$ and the principle eigen vector of the covariance matrix (AA^T) is computed. This vector is used as the input feature to the servoing algorithm. Using the velocity screw V estimated from (2.4), the camera moves to the new pose. This process is repeated until the camera reaches the desired pose. Here again the features F_i are predictively calculated rather than extracting them from each image.

In comparison to classical image-based visual servoing, the use of above features gave a stable camera trajectory. Fig 4.2a shows the velocity screw obtained in the case of the classical servoing algorithm. As only a single image is used, the generated control instructions resulted in an oscillatory

camera trajectory (see the jagged behavior of the velocity screw in Fig. 4.2a as compared to that of Fig. 4.2b), affecting the stability of the robot. Also the system settled in a critically stable condition as the oscillations of the end-effector continued even after the desired position was reached.

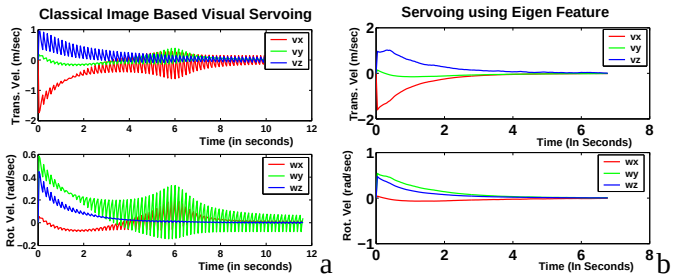

Figure 4.2 Velocity Screw computed (a) only using single image (b) using eigen feature

We also tested the use of image sequence as an input feature to the servoing algorithm. In this case, the feature S was the entire sequence of features extracted from each image in the sequence i.e., $S = [F_1, F_2, ..., F_n]^T$. As a large set of features are considered, minimizing the error difference between the current and the desired feature vector consumed more time and led to the slower convergence of the error function (compare the error norm and the time taken in Fig. 4.3a to that of Fig. 4.3b). Moreover, this method was very sensitive to noise. A small error in the image measurements drastically degraded the performance of the algorithm, hence preventing its use in real life applications.

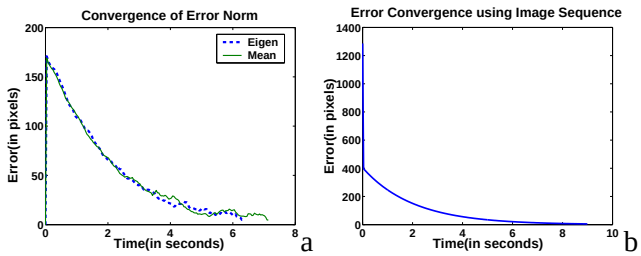

Figure 4.3 Norm of the error of point features Vs Time (a) using eigen and mean feature (b) using image sequence

4.5.2 Stability and Robustness of the Servoing Algorithm

We give an intuitive explanation of the robustness of our approach to calibration errors. Often due to system calibration errors, the obtained image measurements are corrupted. The current image will be corrupted by the error in the current camera pose whereas the projected images will be corrupted by the error in the previous camera poses. This leads to the incorrect computation of the error function as $e^{'}(S) = e(S) + \Delta e$ where $e(S)$ is as defined in Section 2.2.1 and Δe is the measurement error

obtained due to the corresponding error in pose. As a result, the velocity screw changes to $V' = V + \Delta V$ where V is as given in (2.4) and ΔV is the velocity command generated to compensate the corresponding measurement error Δe. This ΔV accommodates the uncertainty in pose and thus ensures the convergence of error to zero. The stability of the algorithm with respect to calibration errors was tested by introducing noise into the camera extrinsic parameters. The noise level was varied and the performance of the algorithm was studied. In Fig. 4.4a, it is observed that in spite of the calibration errors (a 10% gaussain noise was added), the convergence of velocity screw was achieved. This supports our analysis that uncertainty in pose is corrected by the visual feedback rather than being propagated due to the prediction step. However as the error was increased, the trajectory became unstable and the algorithm took longer time to converge.

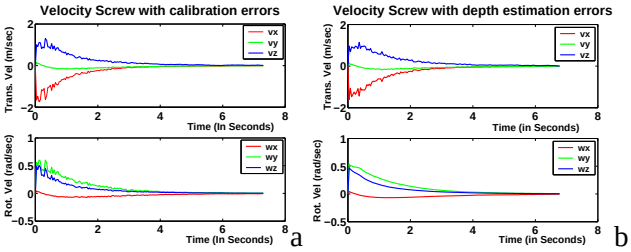

Figure 4.4 Velocity Screw in presence of (a) calibration errors (b) depth estimation errors

One of the crucial steps in the algorithm is the prediction step that assumes the depth of the feature points. The robustness of the proposed algorithm to depth estimation errors was tested by introducing noise into the depth values. Figure 4.4b shows the velocity screw obtained when the estimated depth was corrupted by 15% gaussian noise. This experiment showed that small errors in the depth values can be accommodated by the servoing algorithm.

4.5.3 Analysis of Motion Cancelling Algorithm

Two kinds of motion were studied, namely oscillatory motion and throbbing motion. It must be emphasized that these motions are very common in real-life scenarios. Examples of oscillatory motion include waving of the hand, swaying of trees etc while the motion of heart and other living organs pertain to throbbing motion. Fig. 4.1 shows the manifold obtained in case of throbbing motion. The reference position was chosen such that all the deformations of the object were visible from that camera pose. The first three eigen vectors were selected as the basis vectors.

Next an image from the reference pose was taken and projected onto the eigen-space. Its future image was predicted and reconstructed using the manifold. The error between the reconstructed image and the actual next image was computed. This error should be small which signifies that the eigen-basis capture the important image information that is sufficient to construct the predicted image. In the current case, the error was negligible as the first three eigen values contained 99% of the information. Results

were also analyzed by introducing noise into the image measurments. There was a graceful degradation in the performance as the noise was increased. The above experiments were repeated for the oscillatory motion also. Similar analysis was conducted and the results were studied. The results were very similar in this case also.

4.6 Concluding Remarks

A novel image-based visual servoing algorithm capable of positioning the end-effector in presence of deformable objects has been proposed in this chapter. The method involved extraction of gross features from sequence of images and used them in the servoing process. This algorithm generated a stable camera trajectory unlike the classical image-based servoing algorithm. we have also proposed a novel non-rigid motion canceling algorithm to cancel the relative motion between a non-rigid object and the robot end-effector. By building a linear subspace model, we maintain a lower dimensional representation of the object deformations and use this model to predict the future appearance of the object. A visual servoing algorithm is employed to move the camera from its current position to the desired position. In future, we plan to improve the features that are being used in the servoing algorithm, by considering the appearances of the object rather than point features as it is difficult to extract point features in real life applications. Another challenge is to handle the moving non-rigid object case, where the object has both local and global motion.

Chapter 5

Conclusion

In this thesis, we have presented novel vision-based strategies for performing tasks in unconventional environments. A novel homography-based control capable of positioning a camera even in presence of non-planar objects has been developed for the first time. An optimal homography was efficiently computed using the multiple homographies by employing geometric and subspace constraints. This homography estimate was used in a modified control law to compute the optimal camera trajectory. The method performed better in comparison to existing servoing algorithms and avoided their critical drawbacks. Secondly, a novel image-based visual servoing algorithm capable of positioning the end-effector in presence of deformable objects has been proposed in this chapter. The method involved extraction of gross features from sequence of images and used them in the servoing process. This algorithm generated a stable camera trajectory unlike the classical image-based servoing algorithm. we have also proposed a novel non-rigid motion canceling algorithm to cancel the relative motion between a non-rigid object and the robot end-effector. By building a linear subspace model, we maintain a lower dimensional representation of the object deformations and use this model to predict the future appearance of the object. A visual servoing algorithm is employed to move the camera from its current position to the desired position. Experimental results confirm the effectiveness of our algorithms.

We believe that these approaches will open new research directions in the field of vision-based robot control for domestic environments. In future, we plan to further pursue our work in this area and develop algorithms so as to facilitate the use of robots in real-life scenarios.

Related Publications

- D Santosh Kumar, C V Jawahar. Visual Servoing in Presence of Non-Rigid Motion. *Proc. 18th IEEE International Conference on Pattern Recognition (ICPR), Hong Kong, Aug 2006 (to appear)*
- D Santosh Kumar, C V Jawahar. Robust Homography-based Control for Camera Positioning in Piecewise Planar Environments. *Submitted to 5th Indian Conference on Computer Vision, Graphics and Image Processing (ICVGIP), Madurai, Dec 2006*
- D Santosh Kumar, C V Jawahar. Visual Servoing using Curves. *In Preparation for 24th IEEE International Conference on Robotics and Automation (ICRA), Rome, May 2007*

Bibliography

[1] J. Adachi and J. Sato. Uncalibrated visual servoing from projective reconstruction of control values. *International Conference on Pattern Recognition*, 4:297–300, August 2004.

[2] J. K. Aggarwal, Q. Cai, W. Liao, and B. Sabata. Nonrigid motion analysis: articulated and elastic motion. *Computer Vision and Image Understanding*, 70(2):142–156, May 1998.

[3] J. P. Barreto, J. Batista, and H. Araujo. Model predictive control to improve visual control of motion: Applications in active tracking of moving targets. *International Conference on Pattern Recognition*, 4:4732–4735, September 2000.

[4] R. Basri, E. Rivlin, and I. Shimshoni. Visual homing: surfing on the epipoles. *IEEE International Conference on Computer Vision*, pages 863–869, January 1998.

[5] B. Boufama and R. Mohr. Epipole and fundamental matrix estimation using the virtual parallax property. *IEEE International Conference on Computer Vision*, pages 1030–1036, June 1995.

[6] M. C. Cavusoglu, J. Rotella, W. S. Newman, S. Choi, J. Ustin, and S. S. Sastry. Control algorithms for active relative motion cancelling for robotic assisted off-pump coronary artery bypass graft surgery. *International Conference on Advanced Robotics*, pages 431–436, July 2005.

[7] F. Chaumette and B. Espiau. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–327, June 1992.

[8] F. Chaumette, E. Marchand, and A. Comport. Object-based visual 3d tracking of articulated objects via kinematic sets. *IEEE Workshop on Articulated and Non-Rigid Motion, CVPRW*, 1:2–9, June 2004.

[9] K. Deguchi. Optimal motion control for image-based visual servoing by decoupling translation and rotation. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2:705–711, October 1998.

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

[11] F. Fraundorfer and H. Bischof. Detecting distinguished regions by saliency. *Scandinavian Conference on Image Analysis*, pages 208–215, June 2003.

[12] A. Giachetti and V. Torre. Optical flow and deformable objects. *International conference on computer vision*, (706-711), June 1995.

[13] R. Hartely and A. Zisserman. Multiple view geometry in computer vision. *Cambridge University Press*, 2003.

[14] K. Hashimoto, K. Nagahama, T. Noritsugu, and M. Takaiawa. Visual servoing based on object motion estimation. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1:245–250, October 2000.

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

[16] B. Johansson. View synthesis and 3d reconstruction of piecewise planar scenes using intersection lines between the planes. *IEEE International Conference on Computer Vision*, 1:54–59, September 1999.

[17] T. Kanade, D. D. Morris, and J. Rehg. Ambiguities in visual tracking of articulated objects using two- and three-dimensional models. *International Journal of Robotics Research*, 22(6):393–418, June 2003.

[18] D. Kragic, A. Miller, and P. Allen. Real time tracking meets on–line grasp planning. *International Conference on Robotics and Automation*, 3:2460–2466, May 2001.

[19] E. Malis and S. Benhimane. A unified approach to visual tracking and servoing. *Robotics and Autonomous Systems*, 52(1):39–52, July 2005.

[20] E. Malis and F. Chaumette. 2 1/2d visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, 37(1):79–97, June 2000.

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

[22] S. Nayar and H. Murase. Visual learning and recognition of 3d objects from appearance. *International Journal of Computer Vision*, 14(1):5–24, January 1995.

[23] P. Rives. Visual servoing based on epipolar geometry. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1:602–607, November 2000.

[24] D. Santosh and C. V. Jawahar. Visual servoing in presence of non-rigid motion. *International Conference on Pattern Recognition*, 2006.

[25] A. Shashua and S. Avidan. The rank-4 constraint in multiple view geometry. *European Conference on Computer Vision*, 2:196–206, April 1996.

[26] A. Shashua, A. Levin, and S. Avidan. Manifold pursuit: A new approach to appearance based recognition. *International Conference of Pattern Recognition*, 3:590–594, August 2002.

[27] P. W. Smith. Image-based manipulation planning for non-rigid objects. *International Conference on Robotics and Automation*, 4:3540–3545, May 1998.

[28] M. W. Spong and M. Vidyasagar. *Robot dynamics and control*. John Wiley and Sons, 1989.

[29] C. J. Taylor, J. P. Ostrowski, and S.-H. Jung. Robust visual servoing based on relative orientation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2:574–580, 1999.

[30] W. J. Wilson, C. C. W. Hulls, and G. S. Bell. Relative end effector control using cartesian position based visual sovoing. *IEEE Transactions on Robotics and Automation*, 12(5):684–696, October 1996.