NEAR-MINIMUM TIME VISUAL SERVO CONTROL OF AN UNDERACTUATED ROBOTIC ARM

by

YACINE BENBELKACEM

Thesis submitted in fulfilment of the requirements for the degree of Master of Science

November 2013

ACKNOWLEDGEMENTS

I am grateful to the Lord of the worlds for my very existence and to my parents for their everlasting care and support without whom neither my stay in Malaysia nor the completion of the present manuscript would have been possible.

I express my sincere appreciation to my supervisor Dr. Rosmiwati Mohd-Mokhtar for her kindness, her valuable advices and encouragements throughout the course of my research at the Universiti Sains Malaysia.

I would also like to express with utmost warmheartedness, my appreciation to my dearest brother Hamoudi, my dearest friends Sana Chaker, Swee Liang and Abdul Ghani Abro for the constructive discussions we have had during my stay in Malaysia.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

f	focal length
z_c	Depth of a feature point at the current camera pose
$\tilde{\mathbf{p}}$	Vector $\mathbf{p}$ in homogeneous form
${}^{i}\tilde{\mathbf{p}}$	Vector $\mathbf{p}$ in homogeneous form expressed in frame i
$\boldsymbol{\Omega}$	Camera calibration matrix
$\boldsymbol{\Pi}$	Perspective projection matrix
${}^{i}\mathbf{H}_j$	Homogeneous transformation matrix from frame i to frame j
${}^{i}\mathbf{R}_j$	Rotation matrix from frame i to frame j
${}^{i}\mathbf{t}_j$	Translation vector from frame i to frame j
(α_x, α_y)	Scaling factors along x and y
(u_0, v_0)	Pixel coordinates of the camera optical centre
δ_r	Radial distortion
δ_t	Tangential distortion
$(\mathbf{u}^d, \mathbf{v}^d)$	Distorted pixel coordinates of a point
$\mathbf{k}_c$	Vector of distortion parameters
$\mathbf{v}_c$	Camera velocity screw
$\mathbf{v}_e$	End-Effector velocity screw
$\mathbf{s}$	Vector of current image features
$\mathbf{s}_d$	Vector of desired image features
$\dot{\mathbf{s}}$	Image features velocity vector
$\mathbf{e}$	Error vector
$\mathbf{L}_s$	Interaction matrix at the current camera pose
$\mathbf{L}_d$	Interaction matrix at the desired camera pose
$\boldsymbol{S}(\cdot)$	Skew symmetric operator
ϕ_i	Orientation vector of frame i represented by RPY Euler angles
χ_e	Pose of the end-effector
$\mathbf{q}$	Vector of robot joint angles
θ	Vector of robot joint angles including angle offsets

$\dot{\mathbf{q}}$	Vector of robot joint velocities
$\mathbf{J}$	Robot geometric jacobian
${}^{e}\mathbf{J}$	Robot geometric jacobian expressed in the end-effector frame
$\mathbf{J}_c$	Coupled Robot-Image jacobian
${}^{c}\mathbf{J}_e$	Twist transformation matrix between the end-effector and camera frames
$\mathbf{I}_j$	Identity matrix of dimension $j \times j$

LIST OF ABBREVIATIONS

DOF	Degree(s) of Freedom
IBVS	Image-Based Visual Servoing
PBVS	Position-Based Visual Servoing
HVS	Hybrid Visual Servoing
2-1/2-D	Two and a half Dimensional Visual Servoing
DH	Denavit-Hartenberg
CHT	Circular Hough Transform
NMTVS	Near-Minimum Time Visual Servoing
LMI	Linear Matrix Inequality
RPY	Roll-Pitch-Yaw angle representation of 3D rotations

KAWALAN VISUAL SERVO MASA HAMPIR-MINIMA BAGI LENGAN ROBOT DALAM GERAK

ABSTRAK

Di dalam industri robot, proses mencengkam suatu objek perlu berlaku secara pantas memandangkan kedudukan dan penghalaan suatu objek itu telah diketahui. Namun begitu, sekiranya maklumat tentang kedudukan dan penghalaan itu tidak ada dan objek-objek berada secara rawak di atas penghantar, cabaran akan timbul dalam menetapkan kemahiran dan kelajuan pelaksanaan sesuatu tugas itu. Dewasa ini, penggunaan penderia-penderia penglihatan untuk menghitung kedudukan dan penghalaan suatu objek serta mengubah semula sistem robot telah banyak digunakan. Teknologi ini secara tidak langsung telah memperkenalkan suatu perbezaan masa yang berubah-ubah bergantung kepada teknik kawalan yang dilaksanakan.

Di dalam tesis ini, penyelidikan dilakukan terhadap masa penumpuan bagi tiga pendekatan yang terkenal dalam teknologi servo visual iaitu servo visual berasaskan imej (IBVS), servo visual berasaskan kedudukan (PBVS) dan servo visual hibrid (HVS). Di samping itu, pendekatan masa hampir-minima litar buka berasaskan perancangan laluan bertemu ruang turut dicadangkan bagi meminimakan masa penumpuan. Setiap teknik kawalan ini disimulasikan ke atas robot MITSUBISHI RV-M1 yang mempunyai 5 darjah kebebasan. Keputusan simulasi menunjukkan bahawa pendekatan hampir-minima menumpu kepada masa yang paling singkat berbanding teknik yang lain. Masa penumpuan yang tercatat ialah 1.25 saat berbanding 21.20, 29.50 dan 21.32 saat bagi servo visual berasaskan imej, kedudukan dan hibrid. Teknik masa hampir-minima yang dicadangkan ini juga dilaksanakan secara eksperimen ke atas robot dan masa penumpuan sebanyak 1.49 saat diperhatikan. Keputusan menunjukkan kawalan yang dicadangkan ini berjaya mengatasai pendekatan-pendekatan litar tertutup daripada segi kelajuan.

Penggunaan pendekatan masa hampir-minima litar buka dilihat mampu memberikan impak kepada produktiviti dan kualiti penghasilan di dalam industri robot dan pembuatan. Beberapa contoh keadaan seperti aktiviti pengumpulan, pemeriksaan bahagian dan ubah semula bahagian boleh dilakukan dalam masa yang lebih singkat menggunakan pendekatan ini.

NEAR-MINIMUM TIME VISUAL SERVO CONTROL OF AN UNDERACTUATED ROBOTIC ARM

ABSTRACT

In industrial robotics, grasping an object is required to happen fast since the position and orientation of such an object is a-priori known. However, if such information about the position and orientation is unavailable and objects are spread randomly on a conveyor, it may be challenging to keep the dexterity and speed at which the task is carried out. Nowadays, the use of vision sensors to compute the position and orientation of an object and to reposition the robotic system is being used accordingly. This technology has indirectly introduced a disparity in time that varies according to the nature of the control technique.

In this thesis, an investigation of the convergence time of the three most famous approaches to visual servoing technology, namely Image-Based Visual Servoing (IBVS), Position-Based Visual Servoing (PBVS) and Hybrid Visual Servoing (HVS) is made. In addition, an open-loop near-minimum time approach based on a joint space path planning that minimizes the convergence time is also proposed. Each control technique is simulated on the 5 degrees of freedom MITSUBISHI RV-M1 robot. The simulation results show that the near-minimum time approach converges in a significantly shorter time compared to the other approaches. A convergence time of 1.25 seconds is observed compared to 21.20, 29.50 and 21.32 seconds for Image-Based, Position-Based and Hybrid Visual Servoing respectively. The proposed near-minimum time technique is also experimentally implemented on the robot and a convergence time of 1.49 seconds is observed. The results show that the proposed control outperforms the closed-loop approaches in terms of speed.

The use of the open-loop near-minimum time approach can have a significant impact on the productivity and the quality of production in industrial robotics and manufacturing. Several scenarios including assembly, part inspection and repositioning of parts can be performed in nearly the least possible time using this approach.

CHAPTER 1

INTRODUCTION

1.1 Overview

Robots, for industrial applications in particular, reached a very high level of accuracy and repeatability in the last three decades. Such robots were expected to perform repetitive tasks, times on end, in a well-structured environment, so as to increase productivity. This dramatic improvement in performance was possible only because the environment was made to suit the robot. The workplace in which the robot operates has to undertake a wearisome and expensive calibration without which there will be little use of the robot capabilities. Clearly, this imposed severe limitations on the nature of tasks these robots were assigned. There was a lack of versatility and flexibility since such robots could not operate in a poorly calibrated or unstructured environment because they were deprived of fundamental and necessary sensors, contrary to humans who can adapt quickly to a changing environment.

One of the most crucial sensory feedbacks that was missing in the daily routine of robots, and which could allow a robot to interact with the environment, as poorly structured as it might be, just as humans do, was "visual perception". Most of the limitations of conventional robotics were due to the fact that robots were "blind" and their motion were pre-programmed. The integration of vision in the control loop of robots has proved to bring considerable advantages and to alleviate most of the aforementioned limitations. In comparison to conventional "contact" feedback from force sensors for example, it takes robot perception a step further, by allowing a "non-contact" measurement of the environment (Hutchinson et al., 1996). Contrary to computer vision, vision-based control intends not just to observe the environment but also to interact with it. This is achieved by using the extracted visual information in a control loop to guide the robot in a specific task.

It is henceforth possible, with the aid of vision sensors to bypass the calibration of the workplace and use visual information to tell the robot where to go. In an industrial setup for example, the objects to be manipulated can now be randomly spread on the workspace and no pre-positioning or pre-orientation is required. Vision-based control thus, as a sight-giving technique, renders robotics more flexible, more accurate and more intelligent, contrary to “blind” pre-planned motion. Most industrial robots now embark all sorts of sensory including vision, and the paradigms of Visually Guided Robotics has been well established during the past three decades. However control problems are still to be tackled and this will be discussed in the course of this thesis.

Since its first basic formulation, visual servoing knew a modest advancement at the time, which was primarily due to the non-availability of low cost vision sensors, and the lack of computational capabilities to handle high speed image processing. Now with vision sensors becoming more affordable, as well as the dramatic increase in computational speed, more and more refinements of vision based control have been reported (Chesi and Hashimoto, 2010). Visual servoing is by now a mature research field with considerable sophistications finding its applications in a wide range of disciplines, from industrial and service robotics to space and underwater robotics. Due to its multidisciplinary nature, vision-based control is at the cross roads of different inter-dependent research areas and relies to a great extent on the advancements of each, and that demands a strong cooperative work, (See Figure 1.1).

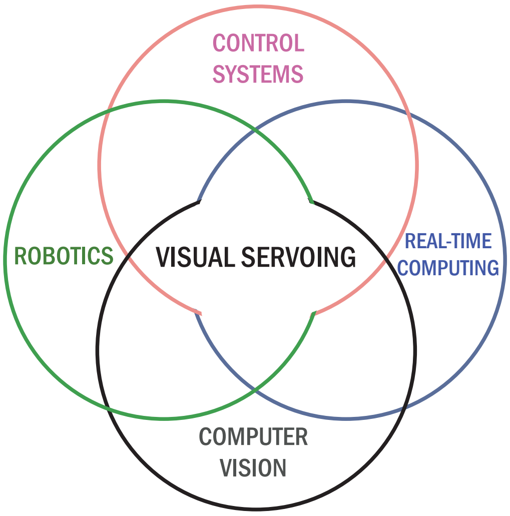


Figure 1.1: Vision-Based Control as a multidisciplinary field

1.2 Motivation of Study

Considerable effort has been made since the first visually guided robot system in the early eighties and nineties, most of which deals with closed-loop visual tracking of moving objects, with little or no reference to grasping tasks (Weiss et al., 1985; Papanikolopoulos et al., 1991; Chaumette et al., 1991; Wang and Wilson, 1992). The subject of grasping objects whether they may be moving or not in a real industrial setup is rather scarce in the literature. The robotic manipulator whose task is to grasp objects scrolling on a conveyor must reach the velocity of the conveyor before the tracking could begin (Nomura and Naito, 2000). From the beginning of the servoing to the moment of grasping, a considerable amount of time is elapsed, affecting the productivity on a large scale.

There is little reference in the literature to the problem of minimizing the time of convergence to the grasping pose which is a factor of paramount importance in productivity. In this thesis, the task of evaluating and analysing different visual control techniques on the basis of the time it takes for each of them to perform a grasping task is studied. It is shown thereafter why open-loop visual control deserves more attention when speed is considered.

Furthermore, in most of the reported simulation work (Chaumette and Hutchinson, 2006), the camera can move freely in 3D space during the servoing process, that is having a full 6DOF motion. This of course is a convenience that does not hold in a real scenario. It is set forth, through experiments conducted on an underactuated robotic arm, how constraints in 3D movement can affect the behaviour of visual information in the image space.

1.3 Objectives

The objectives of the present thesis are enunciated as follows:

1. To model the 5DOF MITSUBISHI RV-M1 robot arm and the vision sensor which consists of the Logitech c525 camera in an eye-in-hand configuration and

establish the relationship between the image space and the robot joint space.

2. To evaluate three different visual servoing techniques on the RV-M1 robot through the analysis of their behaviour both in the image space and cartesian space and their time of convergence to a given grasping pose.

3. To develop and analyse a control scheme that minimizes the time of convergence to a given 3D pose with respect to the object to be grasped.

1.4 Scope of the work

The camera is rigidly mounted on the last joint of the 5DOF MITSUBISHI RV-M1 robot. The movement of the camera is constrained by the physical limits of the robot. Therefore, this limitation has to be accounted for in the design of the control law to avoid unreachable configurations. Furthermore, not all robotic manipulators can achieve any orientation in 3D space, unless they have a minimum of 6 degrees of freedom (Corke, 2011). The RV-M1 robot used in the project has 5DOF. It is an underactuated robot. This imposes additional constraints on the camera pose and on the visual servoing system as a whole.

Furthermore, the desktop computer used to operate the robot needed to have an RS232 port. The available computer runs at 2.66 GHZ and has a 1.24 GB of RAM. It is worth noting that this configuration affects the computation time of image processing algorithms, and the results obtained with this configuration will differ when run with a different configuration. Also, both simulation and experimental setups adopt the Eye-in-Hand configuration and the object to be grasped is motionless.

1.5 Outline

This thesis is structured into five chapters covering, respectively the following themes:

After an introduction to visual perception, a thorough classification of visual servoing control techniques will be given in Chapter 2 and the difference between each of them will be highlighted. Major problems encountered by researchers in particular control schemes will be discussed and four state-of-the-art approaches to deal with these problems will be presented.

Chapter 3 will illustrate the project framework. The first part will be devoted to the modeling of the MITSUBISHI RV-M1 robot arm, with an investigation of the singularities of the robot structure and the velocity relationship between the joint space and the camera space. It is followed by the development of visual servoing control laws. Next, the eye-in-hand experimental setup is depicted.

Chapter 4 is devoted to the simulation and experiments conducted on the robot. The control techniques discussed in Chapter 3 are evaluated and compared on the basis of a particular aspect which is the time of convergence to the grasping pose. The performance of each technique is analysed in detail and conclusions are drawn as to which is more suitable for high speed grasping in an industrial setup.

Chapter 5 concludes the study and discusses the limitations, the weaknesses, and the possible improvements to be made to render the system more accurate and less sensitive to modeling uncertainties.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Robots have taken over humans in a number of repetitive tasks that require dexterity and speed. But contrary to humans, they are designed only to operate in structured and static environments that are painstakingly calibrated to suit them. Clearly, this imposes severe limitations on the use of robots, since these latter ones become clueless at the slightest change in the environment.

From the standpoint of a human being who can interact with a changing environment in real time, vision is undoubtedly the most useful sensory. Attempts to endow robots with a sense of sight to mimic human vision and overcome most of the limitations is indeed very attractive. Incorporating vision into the control loop has received a great deal of attention in the past four decades and has dramatically improved the flexibility and versatility of robotic systems.

This chapter presents the fusion of visual perception and robot motion, appropriately called in the literature visual servoing (Hutchinson et al., 1996) and provides a comprehensive classification of the existing approaches. First, an introduction on the concept of vision guided robotic systems is presented in which it is shown how to generate robot motions from visual information. In the section that follows, different techniques employed relative to the use of visual information and the nature of the induced control laws are listed. Next, the major problems that have been encountered in each technique are discussed with means to overcome them using more advanced schemes. Finally, it is gathered in an overall summary, an illustrative diagram that gives a clearer and much fuller picture of the taxonomy of visual servoing.

2.2 Visual Perception

Vision is by far the richest sensory since it provides more information about the external world than any other sensory (Spero and Jarvis, 2002). Furthermore, unlike other sensors that need a physical contact with the environment, vision allows a non-contact measurement (Corke, 1994). The overwhelming amount of information captured by a vision sensor must undergo a number of analyses and interpretations to extract the particular information that is likely to be practically useful. The science behind this process is called *Computer Vision* (Yi et al., 2005). This discipline harbours a number of aspects that are fields of their own which are roughly categorized into *Image Processing Algorithms* and *Reconstruction Algorithms*. Only the former is relevant to our work, and comprises *Detection*, *Segmentation*, *Feature Extraction* and *Matching*. Addressing these aspects in detail is well beyond the scope of the present thesis. Only essential equations of image formation and feature extraction will be discussed.

2.3 Image Formation

An image is the projection of the three dimensional external world into a two dimensional plane. This projection takes place inside a vision device. The mathematical model of this projection is not unique and depends on the geometry and the nature of the camera lens. For the sake of simplicity, a pinhole camera is considered throughout this thesis.

The principle of the pin-hole Camera was introduced by Ibn-Al-Haytham in the 10^{th} century and published in the Book of Optics (Al-Haytham, 1983). With the technological advances throughout the past centuries, adequate techniques to capture images were developed, from the earlier photo-sensitive films to the contemporary CCD/CMOS sensors. Nonetheless, the principle of image formation remained unchanged.

A typical mathematical model of a Pin-hole camera consists of a virtual optical axis perpendicularly crossing an aperture plate at the centre of which a tiny hole is made (this hole earned the camera its name). A light ray in provenance of an object point P of world coordinates $[x_w \;\; y_w \;\; z_w]$ passes through the hole and hits the sensor plane placed at a distance f called *focal length* from the aperture. The sensor element of coordinates $[u \;\; v]$ is called *pixel* and is mapped into the image plane. Figure 2.1 depicts the model

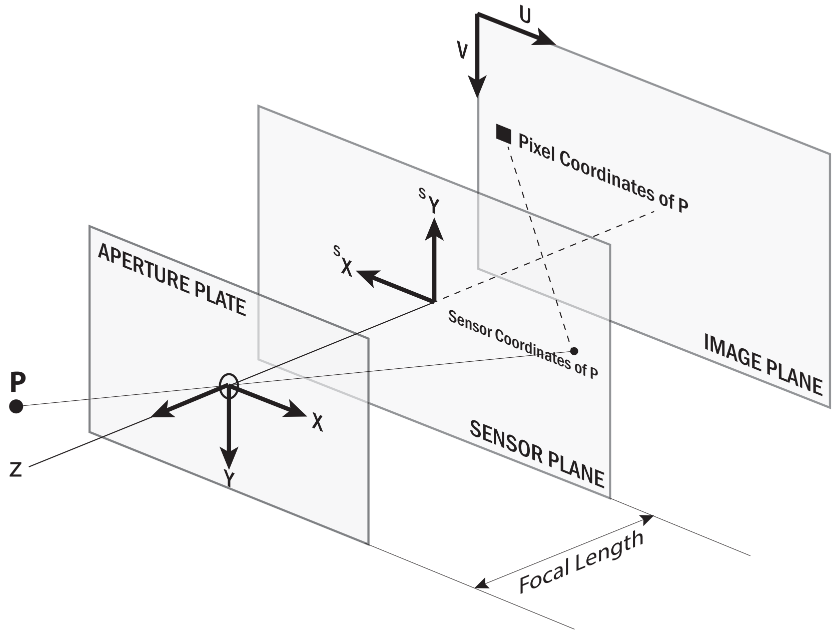


Figure 2.1: Pinhole Camera Model

of the projection process.

To formulate this projection, it is necessary to define three coordinate frames, which are denoted as {W}, {C} and {I} that stand for World, Camera and Image frame, respectively. Then, the relation between the image coordinates and world coordinates of P is given by a series of transformations between these coordinate frames, in the following form

$$z_c\tilde{\mathbf{p}} = \mathbf{\Omega}\mathbf{\Pi}(^{w}\mathbf{H}_c)^{-1}\,{}^{w}\tilde{\mathbf{P}} \tag{2.1}$$

where

z_c : is the *depth* of point P

$\tilde{\mathbf{p}}$: is the vector of pixel coordinates of point P in homogeneous form

$\mathbf{\Omega}$: is the camera *calibration matrix*

$\mathbf{\Pi}$: is the *perspective projection matrix*

$^{w}\mathbf{H}_c$: is the pose of the camera with respect to the world frame

$^{w}\tilde{\mathbf{P}}$: is the vector of world coordinates of point P in homogeneous form

Vectors $\tilde{\mathbf{p}}$ and ${}^{w}\tilde{\mathbf{P}}$ and matrices $\mathbf{\Omega}$, $\mathbf{\Pi}$ and ${}^{w}\mathbf{H}_c$ are defined as

$$\tilde{\mathbf{p}} = \begin{bmatrix} u & v & 1 \end{bmatrix}^T \tag{2.2}$$

$${}^{w}\tilde{\mathbf{P}} = \begin{bmatrix} x_w & y_w & z_w & 1 \end{bmatrix}^T \tag{2.3}$$

$$\mathbf{\Omega} = \begin{bmatrix} f\alpha_x & 0 & u_0 \\ 0 & f\alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.4}$$

$$\mathbf{\Pi} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.5}$$

$${}^{w}\mathbf{H}_c = \begin{bmatrix} {}^{w}\mathbf{R}_c & {}^{w}\mathbf{t}_c \\ 0 & 1 \end{bmatrix} \tag{2.6}$$

where α_x and α_y are two scaling factors that represent the inverse of the pixel size along the x and y axes. u_0 and v_0 are the coordinates of the optical centre relative to the image frame.

It is worth noting that the model described above in equation 2.1 is that of an *ideal* pinhole camera. Such a model is only theoretically valid, since a real lens is always subject to imperfections and distortions that affect the image quality and geometry. To derive a model that is practically valid, lens distortions need to be taken into account.

2.4 Lens Distortion

In this section, one particular type of distortions that is the most problematic in robot vision applications (Corke, 2011), called *geometric distortion* will be considered. It is responsible for aberrations in the image geometry and comprises a *radial* and *tangential* components. The radial component is the more significant of the two. It causes a translation of a point in the image towards the principal point in the radial direction. It is approximated by a polynomial of the form (Weng et al., 1992)

$$\delta r = k_1 r^3 + k_2 r^5 + k_3 r^7 + \cdots \tag{2.7}$$

where r is the distance of point P in the image from the principal point with $r^2 = \mathbf{u}^2 + \mathbf{v}^2$. Straight lines near the edge can curve inward or outward in which case it is called

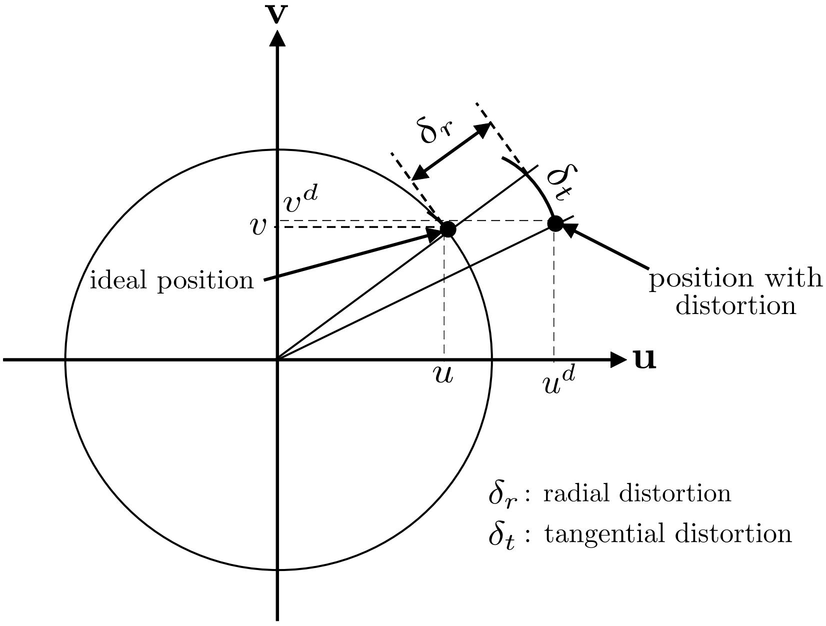


Figure 2.2: Radial and Tangential Distortions

Pincushion and *Barrel* distortion, respectively. Tangential distortion is caused by manufacturing defects, when a lens is not exactly parallel to the image plane. It is characterized by two parameters ρ_1 and ρ_2 (Bradski and Kaehler, 2008), such that

$$\delta t_u = 2\rho_1 \mathbf{v} + \rho_2(r^2 + 2\mathbf{u}^2) \tag{2.8}$$

$$\delta t_v = 2\rho_2 \mathbf{u} + \rho_1(r^2 + 2\mathbf{v}^2) \tag{2.9}$$

The coordinates of point P in the image after distortion read

$$\mathbf{u}^d = \mathbf{u} + \delta_u \tag{2.10}$$

$$\mathbf{v}^d = \mathbf{v} + \delta_v \tag{2.11}$$

where δ_u and δ_v are given by

$$\begin{bmatrix} \delta_u \\ \delta_v \end{bmatrix} = \begin{bmatrix} \mathbf{u}(k_1 r^2 + k_2 r^4 + k_3 r^6 + \cdots) \\ \mathbf{v}(k_1 r^2 + k_2 r^4 + k_3 r^6 + \cdots) \end{bmatrix} + \begin{bmatrix} 2\rho_1 \mathbf{uv} + \rho_2(r^2 + 2\mathbf{u}^2) \\ 2\rho_2 \mathbf{uv} + \rho_1(r^2 + 2\mathbf{v}^2) \end{bmatrix} \tag{2.12}$$

The distortion parameters are then gathered in a (5×1) vector for identification, which is denoted as $\mathbf{k}_c = [k_1 \quad k_2 \quad k_3 \quad \rho_1 \quad \rho_2]$. Figure 2.2 shows the effects of radial and tangential distortions on an image.

2.5 Motion Kinematics

The camera can be either fixed or moving in the environment. In either case, its location is described by some kinematic model. The camera is assumed to have 6 degrees of freedom and can virtually achieve any position and orientation in a given workspace.

Let $\{\mathbf{B}\}$ be a fixed base coordinate frame and $\{\mathbf{C}\}$ be the moving camera attached coordinate frame; and let P be a 3D point of camera coordinates ${}^{c}\mathbf{p}$ and base coordinates ${}^{b}\mathbf{p}$. Then the following relation holds

$$ {}^{b}\mathbf{p} = {}^{b}\mathbf{t}_{c} + {}^{b}\mathbf{R}_{c}{}^{c}\mathbf{p} \tag{2.13} $$

with ${}^{b}\mathbf{t}_{c}$ being the (3×1) translation vector and ${}^{b}\mathbf{R}_{c}$ the (3×3) rotation matrix from the camera frame to the base frame. This relation can be written in a compact form by using a *homogeneous representation* of $\mathbf{p}$ denoted $\tilde{\mathbf{p}} = [\mathbf{p} \quad 1]^{T}$. Equation 2.13 then becomes

$$ {}^{b}\tilde{\mathbf{p}} = {}^{b}\mathbf{H}_{c}{}^{c}\tilde{\mathbf{p}} \tag{2.14} $$

where ${}^{b}\mathbf{H}_{c}$ is given by

$$ {}^{b}\mathbf{H}_{c} = \begin{bmatrix} {}^{b}\mathbf{R}_{c} & {}^{b}\mathbf{t}_{c} \\ 0 & 1 \end{bmatrix} \tag{2.15} $$

and defines both the position and orientation of the camera with respect to the base frame simultaneously.

The movement of the camera in the workspace is supposed to be unconstrained and is described by a (6×1) absolute velocity screw vector denoted $\mathbf{v}_{c}$ composed of the linear and angular velocities, defined by

$$ \mathbf{v}_{c} = [v_{x} \quad v_{y} \quad v_{z} \quad \omega_{x} \quad \omega_{y} \quad \omega_{z}]^{T} \tag{2.16} $$

The object perceived by the camera may be fixed or in movement in which case it is described by the following relative velocity with respect to the camera

$$ {}^{c}\boldsymbol{v}_{o} = \begin{bmatrix} {}^{c}\dot{\mathbf{t}}_{o} \\ {}^{b}\mathbf{R}_{c}^{T}(\omega_{o} - \omega_{c}) \end{bmatrix} \tag{2.17} $$

where ${}^{c}\dot{\mathbf{t}}_o$ is the time derivative of ${}^{c}\mathbf{t}_o$ defined by

$$
{}^{c}\mathbf{t}_o = {}^{b}\mathbf{R}_c^T({}^{b}\mathbf{t}_o - {}^{b}\mathbf{t}_c) \tag{2.18}
$$

which represents the relative position of the origin of the object frame $\{\mathbf{O}\}$ with respect to the camera frame $\{\mathbf{C}\}$, and ω_o and ω_c are, respectively the object and camera angular velocities.

Let s be the vector of image features that characterize the object in question. The nature of the features vary from a simple point to different and more complex geometric shapes. Throughout this thesis, only point features are considered. s is written as a time varying quantity $\mathrm{s} = \mathrm{s}(t)$ due to the camera own motion and the object motion. The variation of feature points in the image are related to the object Cartesian velocity defined by

$$
\frac{\partial \mathbf{s}}{\partial t} = \mathbf{J}_s(\mathbf{s}, {}^{c}\mathbf{H}_o){}^{c}\boldsymbol{v}_o \tag{2.19}
$$

where $\mathbf{J}_s$ is the *image jacobian* mapping feature points movement in the image space to their movement in the Cartesian space.

The relation in equation 2.17 can be written as to highlight the contribution of the camera motion and the object motion by defining their respective absolute velocities $\mathbf{v}_c$ and $\mathbf{v}_o$ given by

$$
\mathbf{v}_c = \begin{bmatrix} {}^{b}\mathbf{R}_c^{T\,b}\dot{\mathbf{t}}_c \\ {}^{b}\mathbf{R}_c^T\omega_c \end{bmatrix} \tag{2.20}
$$

$$
\mathbf{v}_o = \begin{bmatrix} {}^{b}\mathbf{R}_c^{T\,b}\dot{\mathbf{t}}_o \\ {}^{b}\mathbf{R}_c^T\omega_o \end{bmatrix} \tag{2.21}
$$

then equation 2.17 becomes

$$
{}^{c}\boldsymbol{v}_o = \mathbf{v}_o + \Gamma({}^{c}\mathbf{t}_o)\mathbf{v}_c \tag{2.22}
$$

where $\Gamma({}^c\mathbf{t}_o)$ is defined as (Siciliano and Sciavicco, 2009)

$$\Gamma({}^c\mathbf{t}_o) = \begin{bmatrix} -\mathbf{I} & \boldsymbol{S}({}^c\mathbf{t}_o) \\ 0 & -\mathbf{I} \end{bmatrix} \tag{2.23}$$

with $\boldsymbol{S}({}^c\mathbf{t}_o)$ being the skew symmetric operator applied to vector ${}^c\mathbf{t}_o$. Equation 2.19 can then be rewritten as

$$\dot{\mathbf{s}} = \mathbf{J}_s\mathbf{v}_o + \mathbf{L}_s\mathbf{v}_c \tag{2.24}$$

$\mathbf{L}_s$ is called *interaction matrix* and defines a linear mapping between the camera's absolute cartesian velocity $\mathbf{v}_c$ and the features velocity in the image plane $\dot{\mathbf{s}}$ and is given by

$$\mathbf{L}_s = \mathbf{J}_s(\mathbf{s}, {}^c\mathbf{H}_o)\Gamma({}^c\mathbf{t}_o) \tag{2.25}$$

In the case where the object is motionless ($\mathbf{v}_o = 0$), the velocity relation in equation 2.24 becomes

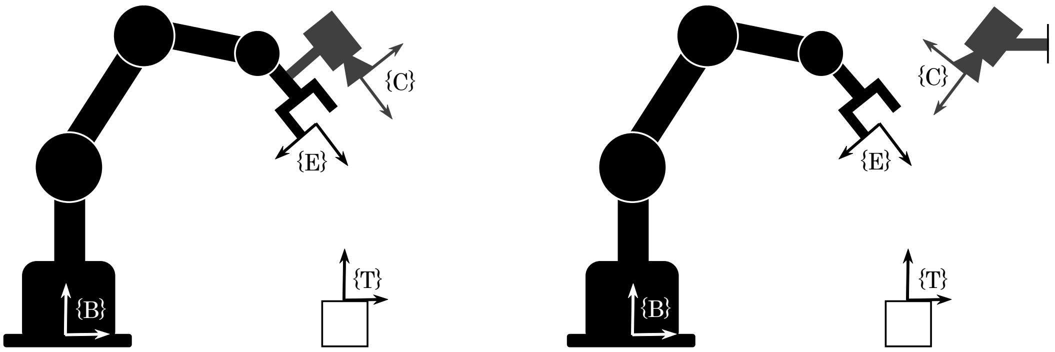
$$\dot{\mathbf{s}} = \mathbf{L}_s\mathbf{v}_c \tag{2.26}$$

The derivation of the interaction matrix for a feature point is given in Appendix A.

2.6 From Perception to Motion

The aim of combining visual perception and motion is to control the camera from an initial arbitrary pose to a final known pose with respect to a given object, using visual information. This control technique has evolved in the seventies under the name *Visual Feedback* (Shirai and Inoue, 1973). It was given, later on, the more specific name *Visual Servoing* by Hill and Park (John and Park, 1979) in 1979. The main difference between the two appellations is that presumably, the former is an open-loop *Look-then-move* control while the second is a closed-loop *Look-and-move* control (Hutchinson et al., 1996).

The camera or cameras capturing images of the scene may be either mounted on a robotic manipulator's gripper or fixed somewhere in the robot's workspace. The former configuration is commonly referred to as *Eye-in-Hand*, whereas the latter is usually



(a) Eye-in-Hand

(b) Eye-to-Hand

Figure 2.3: Camera Configurations

called *Eye-to-Hand* (Hutchinson et al., 1996) or *Standalone* (Kragic and Christensen, 2002). Figure 2.3 depicts the two vision systems using a single camera (*Monocular Vision System*).

It is easily noticeable, that Eye-in-Hand and Eye-to-Hand configurations can be used in conjunction to create a *Binocular Vision System* where two cameras are used simultaneously (Flandin et al., 2000; Lippiello et al., 2005). This leads to three possible variations: Either it be the two cameras mounted on the robot's gripper in which case it is referred to as *Binocular Eye-in-Hand*, or the two cameras fixed which is called *Binocular Eye-to-Hand*, or one camera mounted on the gripper and the other fixed which is named *Hybrid Vision System*. Other variations of the aforementioned configurations are found in the literature. As a matter of example, there are those that use more than two cameras (Paulo et al., 1998) combined in either of the two main configuration of Figure 2.3. Such vision systems are called *Redundant*. Examples where both Eye-in-Hand and Eye-to-Hand are used in a cooperative fashion can be found in (Gengenbach et al., 1996; Christian and Bernd, 1998), From this brief introduction, the Eye-in-Hand and Eye-to-Hand configurations thus, constitute a framework upon which any vision system can be built regardless of the number of cameras it uses.

A great number of the reported work in the literature adopts monocular vision for a number of reasons. One main reason is that using a single camera alleviates the computational time and burden of image interpretation and processing (Kragic and

Christensen, 2002). In addition, it is simpler to simulate since it recalls simple projective geometry. However, the main drawback of using monocular vision is that the depth information about the object is lost and cannot be precisely recovered but only estimated (Chaumette and Hutchinson, 2006; Fang and Lin, 2001; Papanikolopoulos et al., 1995). This problem is absent when using binocular vision, in which case the precise value of the depth can be obtained using epipolar geometry from two views of the object (Maru et al., 1993).

Eye-in-Hand and Eye-to-Hand configurations are to be used in particular situations where one is likely to perform better than the other. The Eye-in-Hand configuration is better adapted for tasks that require a close look at the object in which case the view of the scene is local and only a portion of the workspace is considered. This is useful when the proportion in size between the object and the workspace is small and the manipulation requires a precise sight. On the other hand, the Eye-to-Hand configuration is more useful in the opposite situation, that is, when a global sight of the scene is required, and when the robot's end-effector needs to be tracked at the same time as the object (Kim et al., 2004).

2.7 Approaches to Visual Servoing

Classifying visual servoing systems is a rather difficult task due to the non-uniqueness of the criteria used to categorize the different approaches and sometimes the non-conformal taxonomy employed by the authors.

Approaches to visual servoing can be categorized depending on different criteria:

- Depending on whether or not a geometric model of the object to be manipulated is known, *Model-Based* and *Model-Free* visual servoing are considered.

- Depending on whether or not the intrinsic/extrinsic parameters of the camera are known, *Calibrated* and *Uncalibrated* visual servoing are considered.

- Based on the control type, that is, whether a visual feedback exists or not, *Closed-loop* and *Open-loop* visual servoing are considered.

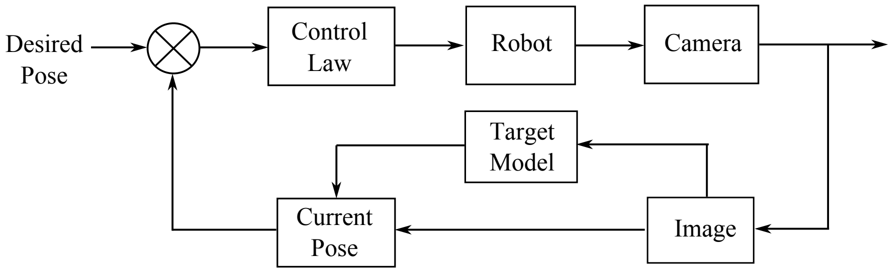


Figure 2.4: Model-Based Visual Servoing

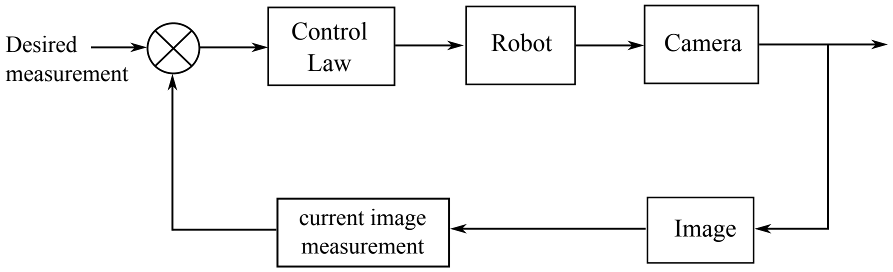


Figure 2.5: Model-Free Visual Servoing

- Based on the feedback or the nature of the error used to compute the control law, *Image-Based, Position-Based* and *Hybrid* visual servoing are considered.

In *Model-Based* visual servoing, the object's model is required with at least four feature points in addition to a calibrated camera. However, it is possible to still servo the system with an uncalibrated camera if more than four feature points are available (Malis, 2002). In the *Model-Free* visual servoing, the positioning task can be achieved without any knowledge of the object's geometric model by having recourse to a "teaching by showing" technique (Chesi and Hung, 2007). Figures 2.4 and 2.5 show the block diagrams of the Model-Based and Model-Free approaches.

In (Liu et al., 2006), it is shown how it is possible to use an uncalibrated camera to control a robot from an initial to a desired pose. This is done by deriving an error vector between the current view and desired view of the object independent of the metric coordinates of the feature points. Thus, an interaction matrix independent of the depth variable will be obtained. Further categories of visual servoing systems which are important for this research are presented in detail in the following sections.

2.7.1 Visual Servoing Based on the Control Type

A fundamental distinction in any control system consists in the Open-loop/Closed-loop approaches to the control problem. In visual servoing, this distinction is made with respect to the use of visual information.

2.7.1.1 Open-loop Control (Look-then-Move)

In the open loop approach, the visual information provided by the camera is directly used to generate a control signal that is fed to the robot (Gao et al., 2006). The robot is initially at an arbitrary pose with respect to the object. If the Eye-in-Hand configuration is considered as a matter of example, the pose of the object with respect to the camera ${}^{c}\mathbf{H}_{o}$ is estimated using a pose estimation algorithm. Then using the forward kinematics of the robot, the pose of the end-effector with respect to a fixed world frame (which is taken to be the robot's base frame) is obtained. Those two poses are combined along with the fixed and supposedly known Hand-eye transformation ${}^{e}\mathbf{H}_{c}$ (Tsai and Lenz, 1989) (i.e. the transformation from the end-effector to the centre of the camera frame) to compute an estimate of the object's pose with respect to the base frame ${}^{b}\mathbf{H}_{o}$. This in turn is used to compute the desired end-effector pose to which the robot is then steered (grasping pose).

It is important to note that in this case, the camera must be calibrated, that is, its intrinsic and extrinsic parameters are known and the geometric model of the target is available. So the open-loop approach is a model-based calibrated visual control. A limiting aspect of this approach is that the robot environment is supposed to remain static once the robot has started to move, that is, the object stands still while the robot is moving towards it. Figure 2.6. illustrates such an approach.

2.7.1.2 Closed-loop Control (Look-and-Move)

The closed-loop control differs fundamentally from the open-loop control since the pose of the object with respect to the camera is continuously updated as the robot moves. The visual information is fed back to the robot controller and image processing is performed

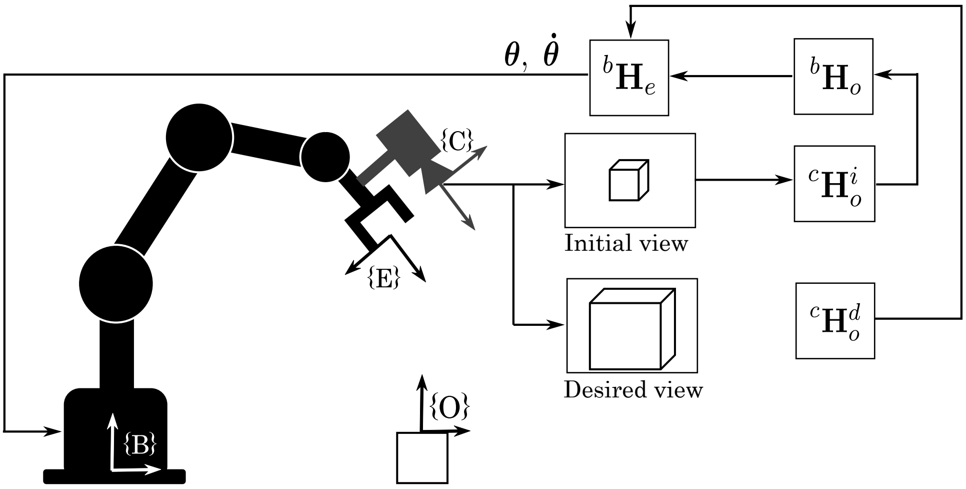


Figure 2.6: Camera Configurations

at each iteration. The environment in which the robot operates thus, does not have to remain static but may be constantly changing,. By using the visual feedback loop, the robotic system is able to track the object in question even if this latter is moving.

There are two possible ways to achieve a closed-loop visual control (Hutchinson et al., 1996). One is by considering the robot inner control loop to interpret and convert the visual control signal into a joint control signal, and the other is by eliminating the robot controller and using directly the visual control signal as input to the robot. The former control scheme is called *Indirect Visual Servoing* and the latter *Direct Visual Servoing*.

Indirect Visual Servoing This control scheme is found in the literature under the name *Dynamic-Look-and-move*, and according to (Kragic and Christensen, 2002), almost all the reported work follows this approach. The servoing task is achieved in two steps. First, the visual system issues a velocity control signal in terms of visual measurements about the object (the nature of the measurement may be 2 dimensional or 3 dimensional, which will be addressed later in this chapter). It is then sent to the robot controller, which through an inner joint feedback, transforms it into a robot joint trajectory to move the end-effector and hence the camera to its sought position and orientation.

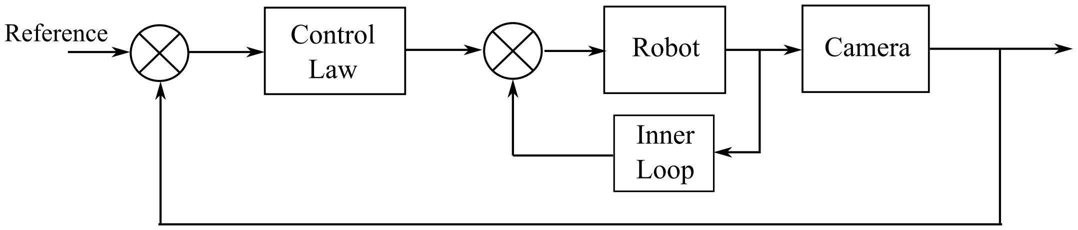


Figure 2.7: Indirect Visual Servoing

Figure 2.7 (Malis, 2002), shows the block diagram of this control method. It is important to point out as in (Corke, 1996*a*) that such a control scheme requires the robot's inner loop to be faster than the visual system's outer loop. It is also worth noting that the dynamic effects that are likely to occur during the servoing process (both the robot and the visual system dynamics) are not fully taken into account in this control scheme. Instead, they are modelled as a constant gain (Chaumette and Hutchinson, 2006). Henceforth, it is obvious that the aforementioned control method is relevant as long as the velocity at which the robot moves does not exceed a threshold, above which the dynamics of the system can no longer be ignored or ill-modelled.

Direct Visual Servoing As opposed to the precedent case, this control scheme bypasses the robot's inner joint loop, and uses instead the control signal issued by the visual controller directly to move the robot. This time, the dynamics of the system are taken into account and the sought result is that of a high performance visual servoing that can operate at high speeds (Corke, 1996*b*). The introduction of the robot dynamics makes the system relatively complex to design and model, and few systems are reported in the literature that follow this approach (Corke, 1996*b*; Weiss et al., 1985). Figure 2.8 shows the block diagram of this control scheme.

2.7.2 Visual Servoing Based on the Feedback

Technically, in a visual servoing task, the aim is ultimately to achieve a desired camera situation with respect to a given object by minimizing the error between this desired situation and the current one. The error to be minimized is formulated in terms of visual

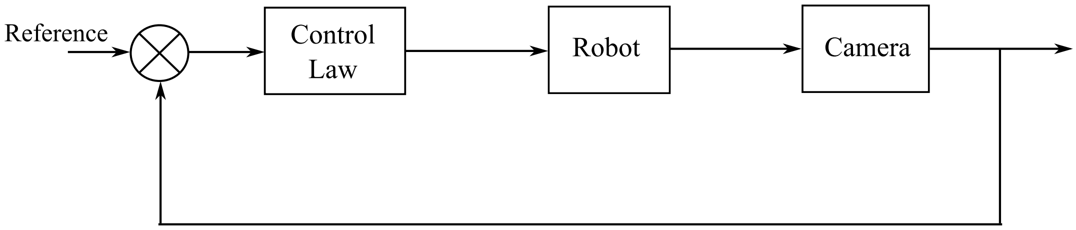


Figure 2.8: Direct Visual Servoing

measurements as follows (Chaumette and Hutchinson, 2006)

$$\mathbf{e}(t) = \mathbf{s}_d - \mathbf{s}(t) \tag{2.27}$$

The nature of the visual measurement denoted $\mathbf{s}(t)$ in the above equation can be either two dimensional or three dimensional or both. This gives rise to three different approaches to the problem. A two dimensional measurement consists of expressing the object by its projection in the two dimensional image plane (The object is represented by some chosen features), whereas a three dimensional measurement consists of expressing the object by its pose, that is, its position and orientation with respect to the vision sensor (Hutchinson et al., 1996). A visual measurement that involves both 2D and 3D information is called hybrid and consists of a decoupling of translational and rotational motions by using 2D measurements for the former and 3D measurements for the latter. The resulting control approaches are called, respectively *Image-based visual servoing* (2D), *Position-based visual servoing* (3D) and *Hybrid visual servoing* (2-1/2-D).

2.7.2.1 Image-Based Visual Servoing (IBVS)

Image-based visual servoing, like its name suggests, uses measurements about the object in terms of its current feature coordinates in the image, and moves the robot end-effector to achieve a set of desired feature coordinates. The control law is entirely defined in the image space between feature coordinates in the current and desired views, as shown in Figure 2.9. Such a control involves the computation of the "*interaction matrix*" defined in equation 2.25 to estimate the camera velocity screw that will achieve this task.

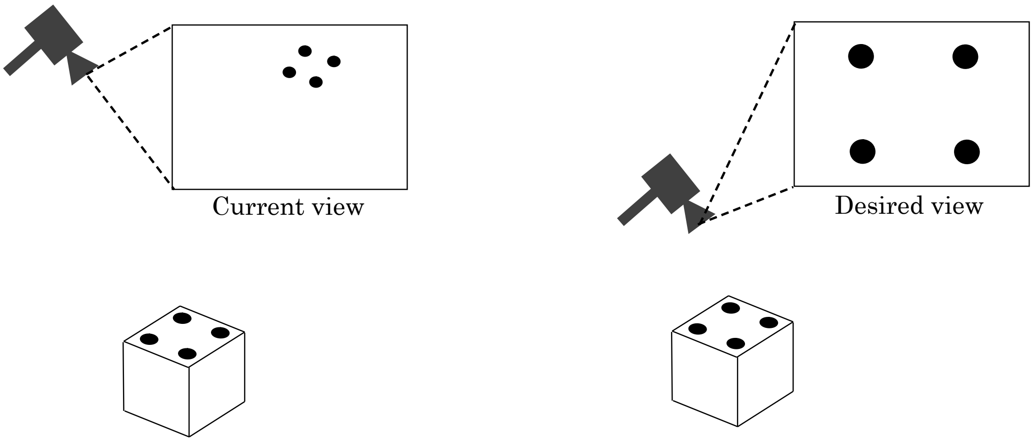


Figure 2.9: Current and desired views of the target

Control Formulation

The rate of change in feature coordinates as a function of the rate of change in the camera pose is defined as in equation 2.26. The camera velocity screw composed of the linear and angular velocities along and about the camera frame axes are as defined in equation 2.16. Given the camera intrinsic parameters represented by the calibration matrix $\mathbf{\Omega}$ of equation 2.4, the computation of the interaction matrix depends on the sole unknown variable z_c. It is assumed that the object is fixed with respect to the base frame $(\frac{\partial \mathbf{s}_d}{\partial t} = 0)$. If equation 2.26 is substituted into the time derivative of equation 2.27, it follows

$$\dot{\mathbf{e}} = \mathbf{L}_s \mathbf{v}_c \tag{2.28}$$

Adopting a resolved motion rate control (Craig, 2005), the control law is formulated to guarantee that the error tends asymptotically to zero

$$\mathbf{v}_c = \lambda_s \widehat{\mathbf{L}}_s^+ (\mathbf{s}_d - \mathbf{s}) \tag{2.29}$$

where $\widehat{\mathbf{L}}_s^+$ is an estimate of the left pseudo-inverse of $\mathbf{L}_s$ due to the estimated value of $\widehat{z}_c$; and λ_s is a dampening factor. It is important to point out that two choices for $\mathbf{L}_s^+$ are possible, mainly an estimate that requires a depth computation at each step of the control, or an estimate that uses a constant depth, usually the depth at the desired pose (Chaumette and Hutchinson, 2007). The block diagram of such a control is given in

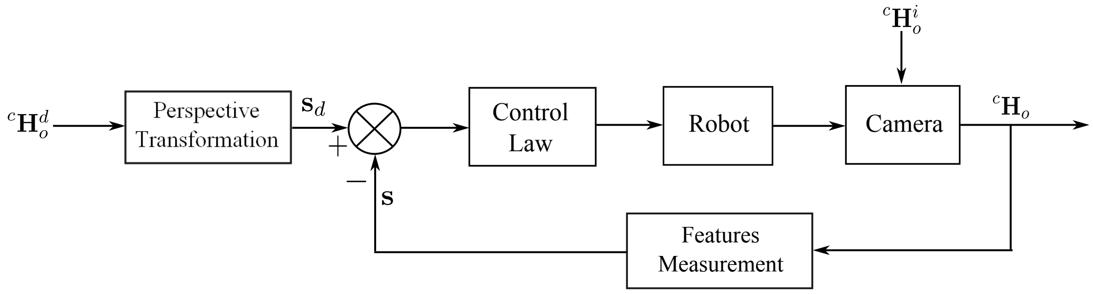


Figure 2.10: Image-Based Visual Servoing

Figure 2.10.

2.7.2.2 Position-Based Visual Servoing (PBVS)

In the position-based approach, the features extracted from the image are used along with the geometric model of the object to estimate its pose ${}^{c}\mathbf{H}_o$ with respect to the camera. The control law is formulated in terms of this 3D pose and not in terms of the image feature coordinates. To this end, the camera needs to be calibrated and the model of the object is known. The feature vector s in equation 2.27 represents a 3D measurement.

Let {C} and {O} be the camera and object frames, respectively and let ${}^{c}\mathbf{H}_o$ and ${}^{c^*}\mathbf{H}_o$ be, respectively the current and desired object poses with respect to the camera obtained using a pose estimation technique. Figure 2.11 illustrates the notation used above.

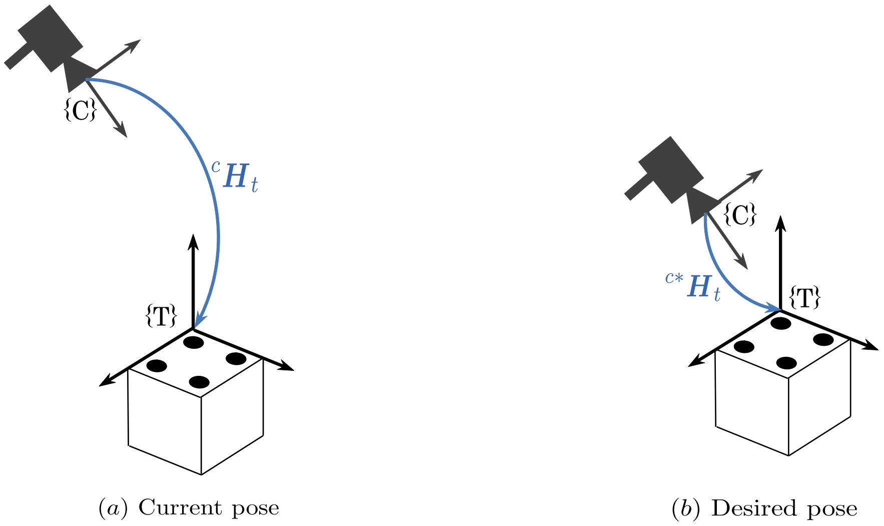
Control Formulation

The object is assumed to be motionless during the servoing process. The Position-based approach is formulated in such a way to achieve a desired camera pose from the current camera pose expressed by the following homogeneous transformation matrix

$${}^{c^*}\mathbf{H}_c = {}^{c^*}\mathbf{H}_o({}^{c}\mathbf{H}_o)^{-1} = \begin{bmatrix} {}^{c^*}\mathbf{R}_c & {}^{c^*}\mathbf{t}_c \\ 0 & 1 \end{bmatrix} \tag{2.30}$$

The error vector is computed as

$$\mathbf{e} = -\begin{bmatrix} {}^{c^*}\mathbf{t}_c \\ \phi_c \end{bmatrix} \tag{2.31}$$



(a) Current pose (b) Desired pose

Figure 2.11: Current and desired pose of the target

where ϕ_c represents the vector of Euler angles obtained from the rotation matrix ${}^{c^*}\mathbf{R}_c$. The error vector depends only on the current and desired camera poses. The control law is then designed so that the error $\mathbf{e}$ tends asymptotically to zero. Adopting a resolved motion rate control (Craig, 2005),

$$\mathbf{v}_c = \lambda_s \widehat{\mathbf{L}}_s^+ \mathbf{e} \tag{2.32}$$

It is worth noticing that in this case, because of the interaction matrix having the following form, with L_1 containing only translational components and L_2 rotational components

$$\boldsymbol{L}_s = \begin{bmatrix} L_1 & 0 \\ 0 & L_2 \end{bmatrix} \tag{2.33}$$

a decoupling of translation and rotation is achieved, and the control law can be rewritten as

$$v_c = \lambda_s L_1 \tag{2.34}$$
$$\omega_c = \lambda_s L_2 \tag{2.35}$$

with v_c and ω_c being the translational and rotational vectors of the camera velocity screw $\mathbf{v}_c$. The block diagram of such a control is given as in Figure 2.12.

The sum block in Figure 2.12 that computes the error $\mathbf{e}$ has a conceptual meaning

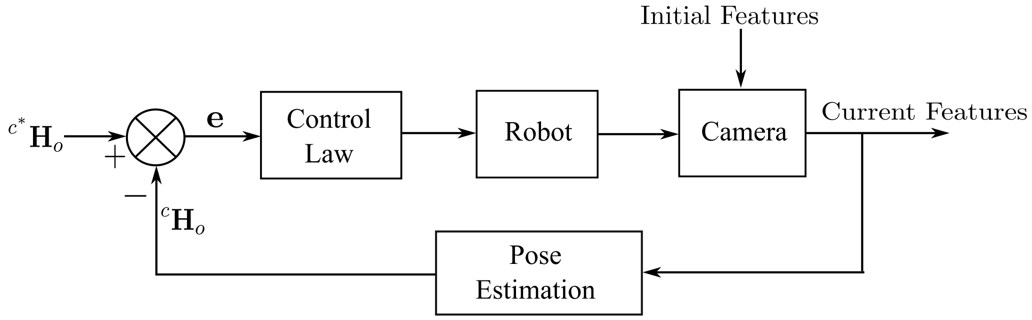


Figure 2.12: Position-Based Visual Servoing

and corresponds to the difference between two poses (matrices) and not to an algebraic subtraction.

2.7.2.3 Hybrid Visual Servoing (2-1/2-D)

The hybrid approach was first introduced by (Malis et al., 1999). It exploits the decoupling property of PBVS in conjunction with a separate translational motion control from IBVS.

Let $\mathbf{s}_t$ and $\mathbf{e}_t$ be the feature and error vectors, respectively responsible of controlling the translational motion of the camera, then

$$\dot{\mathbf{s}}_t = \mathbf{L}_{s_t}\mathbf{v}_c = [L_v \quad L_\omega]\begin{bmatrix} v_c \\ \omega_c \end{bmatrix} = L_v v_c + L_\omega \omega_c \tag{2.36}$$

$$\dot{\mathbf{e}}_t = \dot{\mathbf{s}}_t = -\lambda \mathbf{e}_t \tag{2.37}$$

Substituting equation 2.37 into equation 2.36 yields

$$L_v v_c = -L_\omega \omega_c - \lambda \mathbf{e}_t \tag{2.38}$$

which gives the translational motion control

$$v_c = -L_v^+(L_\omega \omega_c + \lambda \mathbf{e}_t) \tag{2.39}$$

Here, the term $(L_\omega \omega_c + \lambda \mathbf{e}_t)$ represents the error to be minimized. It is important to note that this error comprises the original error $\mathbf{e}_t$ to which is added an error induced by