

# Image-based visual servoing using improved image moments in 6-DOF robot systems

Sining Liu

A Thesis

In

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Applied Science (Mechanical Engineering) at

Concordia University

Montreal, Quebec, Canada

October 2008

© Sining Liu, 2008



Library and Archives  
Canada

Published Heritage  
Branch

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

Bibliothèque et  
Archives Canada

Direction du  
Patrimoine de l'édition

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
ISBN: 978-0-494-63305-2  
*Our file* *Notre référence*  
ISBN: 978-0-494-63305-2

**NOTICE:**

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

**AVIS:**

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

# **Abstract**

## **Image-based visual servoing using improved image moments in 6-DOF robot systems**

Sining Liu

Visual servoing has played an important role in automated robotic manufacturing systems. This thesis will focus on this issue and proposes an improved method which includes an ameliorative image pre-processing (IP) algorithm and an amendatory IBVS algorithm.

As the first contribution, an improved IP algorithm based on the morphological theory has been discussed for the purpose of removing the unexpected speckles and balancing the illumination during the image processing. After this enhancing process, the useful information in the image becomes prominent and can be utilized to extract the accurate image features. Then, an improved IBVS algorithm is therefore further introduced for an eye-in-hand system as the second contribution. This eye-in-hand system includes a 6 Degree of Freedom (DOF) robot and a camera. The improved IBVS algorithm utilizes the image moment as the image features instead of detecting the special points for feature extraction in traditional IBVS. Comparing with traditional IBVS, choosing image moment as the image features can increase the stability of the system and extend the applied range of objects. The obtained image features will then be used to generate the control signals for the robot to track the target object. The Jacobian matrix describing the relationship between the motion of camera and velocity of image features is also discussed, where a new simple method has been proposed for the estimation of depth involved in the Jacobian matrix. In

order to decouple the obtained Jacobian matrix for controlling the motion of camera with individual image features, a four stages sequence control has also been introduced to improve the control performance.

# Acknowledgments

First and foremost, I would like to thank my supervisors, Dr. Wenfang Xie and Dr. Chun-Yi Su, who have provided me plenty of advice and help on my research and my life here. Especially their outstanding insights of visual servoing and excellent guidance played a significant role in my whole research.

I also would like to thank my parents for their long lasting and firm support. I would like to express my sincere appreciation to them for encouraging me to overcome the difficulties and challenges.

Moreover, I would like to take this opportunity to thank my dear friends in Montreal, Ming Lei, Zuhui Ma, Xuan Han and Yuhang Luo. They not only brought me lots of wonderful memories, but also gave me the encouragement and support both in my academic and social life.

Last but not least, I would like to appreciate everyone who helped me during the past two years.

# TABLE OF CONTENTS

LIST OF FIGURES .....	viii
LIST OF TABLES.....	x
NOMENCLATURE.....	xi
CHAPTER 1 INTRODUCTION .....	1
1.1 Introduction .....	1
1.2 Background and Previous Works .....	1
1.2.1 Background of Visual Servoing .....	1
1.2.2 Background of image-based visual servoing (IBVS).....	7
1.2.3 Background of image features extraction .....	9
1.2.4 Background of image pre-processing for visual servoing .....	11
1.3 Motivation and Contribution .....	12
1.4 Thesis organization .....	14
1.5 Summary .....	15
CHAPTER 2 MORPHOLOGICAL IMAGE PROCESSING.....	17
2.1 Introduction .....	17
2.2 Background of morphological image processing .....	17
2.2.1 Gray-scale dilation and erosion.....	18
2.2.2 Gray-scale opening and closing [47].....	20
2.3 Proposed Image Pre-processing Algorithm.....	21
2.4 Simulation results .....	25
2.5 Summary .....	32
CHAPTER 3 IMAGE FEATURES EXTRACTION.....	34
3.1 Introduction .....	34
3.2 Image moments .....	34
3.2.1 Background of image moments.....	36
3.2.2 Choose the proper image features.....	38

3.2.3	Derivation of Jacobian Matrix.....	41
3.3	Depth estimation.....	52
3.4	Summary.....	54
CHAPTER 4 IBVS WITH IMAGE MOMENTS.....		56
4.1	Introduction.....	56
4.2	Proposed sequence control algorithm.....	56
4.3	Summary.....	68
CHAPTER 5 SIMULATION RESULTES OF IBVS WITH IMAGE MOMOENTS .....		69
5.1	Introduction .....	69
5.2	Simulation results of IBVS with image moments .....	69
5.2.1	Results with ideal condition.....	69
5.2.2	Robustness test.....	85
5.2.3	Comparison.....	87
5.3	Summary .....	90
CHAPTER 6 CONCLUSION AND FUTURE WORK .....		91
6.1	Introduction .....	91
6.2	Conclusion.....	91
6.3	Future work .....	93
BIBLIOGRAPHY.....		94
APPENDIX A.....		101

## LIST OF FIGURES

Figure 1-1 Different applications of robot .....	2
Figure 1-2 Dynamic look-and-move system .....	4
Figure 1-3 Direct visual servo system.....	4
Figure 1-4 Block diagram of PBVS.....	5
Figure 1-5 Block diagram of IBVS.....	6
Figure 1-6 Block diagram of HVS (2-1/2-D method [11]) .....	6
Figure 2-1 dilation and erosion [24] .....	20
Figure 2-2 The initial image .....	22
Figure 2-3 Result with $k=1$ .....	23
Figure 2-4 Result with $k=10$ .....	23
Figure 2-5 Result with $k=15$ .....	24
Figure 2-6 The initial image with unexpected speckles.....	26
Figure 2-7 The image obtained after step 2-pre-processing.....	27
Figure 2-8 The final image .....	27
Figure 2-9 The initial image with bad illumination.....	28
Figure 2-10 The image obtained after step 2-pre-processing.....	29
Figure 2-11 The final image .....	29
Figure 2-12 The initial image with uneven illumination.....	31
Figure 2-13 Result with use the algorithm proposed in [25].....	31
Figure 2-14 The final image by using proposed method .....	32
Figure 3-1 The image features in traditional IBVS of regular and irregular situations .....	35
Figure 3-2 Basic IBVS frame .....	42
Figure 3-3 The perspective camera model .....	43
Figure 3-4 Time variation of $C(t)$ .....	46
Figure 3-5 Depth estimation .....	53
Figure 4-1 Block diagram of proposed IBVS.....	57
Figure 4-2 Flow chart of the proposed IBVS algorithm .....	58
Figure 4-3 Initial position of stage 2.....	61

Figure 4-4 Desired position of stage 2 .....	61
Figure 4-5 Initial position of stage 3 .....	64
Figure 4-6 Desired position of stage 3 .....	65
Figure 4-7 Initial position of stage 4 .....	67
Figure 5-1 Initial image of task 1 .....	71
Figure 5-2 Image of task 1 after step one .....	72
Figure 5-3 Image of task 1 after step two .....	72
Figure 5-4 Final image of task 1 .....	73
Figure 5-5 Errors of image features of task 1 .....	74
Figure 5-6 Initial image of task 2 .....	75
Figure 5-7 Image of task 2 after step one .....	75
Figure 5-8 Image of task 2 after step two .....	76
Figure 5-9 Final image of task 2 .....	77
Figure 5-10 Errors of image features of task 2 .....	77
Figure 5-11 Initial image of task 3 .....	80
Figure 5-12 Final image of task 3 .....	80
Figure 5-13 Errors of image features of task 3 .....	81
Figure 5-14 Zoom in of figure 5-13 .....	81
Figure 5-15 Initial image of task 4 .....	83
Figure 5-16 Final image of task 4 .....	83
Figure 5-17 Errors of image features of task 4 .....	84
Figure 5-18 Zoom in of figure 5-17 .....	84
Figure 5-19 Initial image of robustness test .....	85
Figure 5-20 Final image of robustness test .....	86
Figure 5-21 Errors of image features of robustness test .....	86
Figure 5-22 Image in regular situation .....	88
Figure 5-23 Image in irregular situation .....	88

## LIST OF TABLES

Table 1-1 Classification of visual servoing.....	6
Table 3-1 Comparison of $P_x$ , $P_y$ and $S_x$ , $S_y$ .....	39
Table 5-1 Desired value and initial value for task 1 .....	70
Table 5-2 The desired and initial image feature values for task 3.....	78
Table 5-3 The desired and initial image feature values for task 4.....	82
Table 5-4 Comparison of point based method and moment based method .....	89

## NOMENCLATURE

DOF	Degree of Freedom
FOV	Field of View
$I$	Image
$I_i$	Initial image
$I_f$	Final image
$g$	Structuring element
$\oplus$	Dilation operation
$\ominus$	Erosion operation
$\circ$	Opening operation
$\bullet$	Closing operation
$k$	Gain of image processing
$I_{mean}$	Average grayscale value of $I$
$\Omega$	Variance
$m_{ij}$	Geometric moment of order $i+j$
$\mu_{ij}$	Central moment of order $i+j$
$M$	Hu's invariant
$(x, y)^T$	Image coordinates in image frame
$(x_g, y_g)^T$	Coordinates of object centroid
$(u, v)^T$	Pixel index of a point
$(u_0, v_0)^T$	Image center pixel coordinates (principal point)

$(X, Y, Z)^T$	Coordinates of a point in camera frame
${}^{F_1} \mathbf{H}_{F_2}$	Homogeneous transformation matrix of the frame $F_2$ with respect to the frame $F_1$ .
${}^{F_1} \mathbf{v}_{F_2}$	Translational velocity vector of the frame $F_2$ with respect to the frame $F_1$ .
${}^{F_1} \boldsymbol{\omega}_{F_2}$	Angular velocity vector of the reference $F_2$ with respect to the frame $F_1$ .
$\lambda$	The focus length of camera
$J_{image}$	Image Jacobian matrix
$\dot{r}$	Velocity screw of the camera
$s$	Image feature
$\tilde{s}$	Image features error
$Z$	Depth
$h$	Height of camera with respect to world frame
$\beta$	Angle between the optical beam and the vertical
$W$	World frame
$C$	Camera frame
$u$	Feedback controller
$K$	Proportional gain
$s_{thre2}$	Threshold

# CHAPTER 1 INTRODUCTION

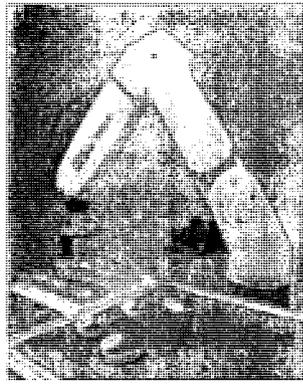
## 1.1 Introduction

According to Seth Hutchinson et al. [1], the task of visual servoing is using visual information to control the pose of the robot's end-effector relative to a target object or a set of target features. In order to indicate the relative pose, the image features are required to be extracted. In this chapter, the basic concepts and background of visual servoing and image moments which are used to extract the image features are introduced. Furthermore, the motivation and contribution will also be addressed. In the end, the organization of this thesis will be summarized.

## 1.2 Background and Previous Works

### 1.2.1 Background of Visual Servoing

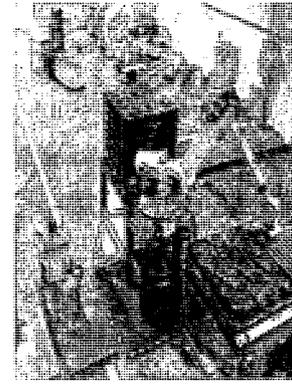
Robots are now applied widely to different areas, from fruit picking to robotized laparoscopic surgery [36]. Especially, the robot plays an important role in industrial field. For example, assembling, packaging, drilling, painting and prototyped span manufacturing (grasping objects on conveyor belts and part mating) [1] [2].



(a) Assembling [6]



(b) Packaging[8]



(c) Picking[7]

**Figure 1-1 Different applications of robots**

With the highly evolution of robot applications, the robot control technology has been well developed.

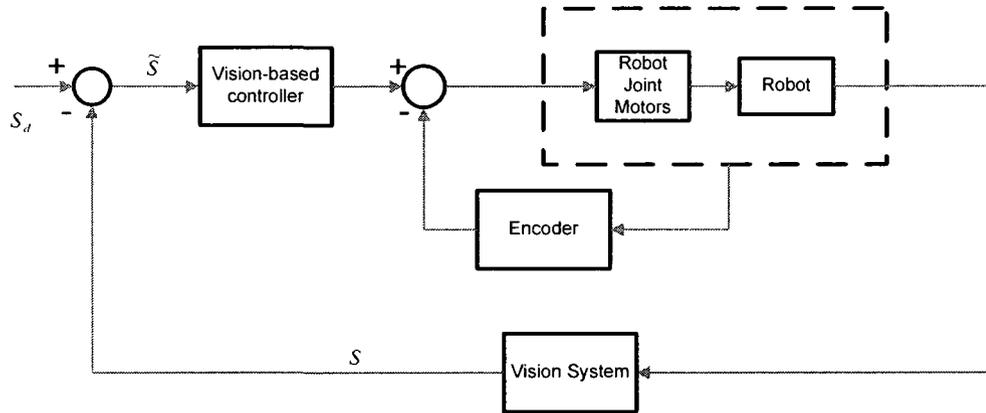
Before 1970s', *absolute motion control* was widely used in robot control systems. For such a motion control, the target object had to be put in a fixed world coordinate frame, for example, the robot base frame. Only the absolute pose (3D translation and orientation) of end-effector in robot base frame with respect to the target object would be considered as the feedback signals. In this case, the operation was easily affected by any errors about the pose of the end-effector or target object. Also, another shortcoming of the absolute motion control was that it required considerable reprogramming before new tasks.

To overcome the above- mentioned disadvantages , some sensors were added in conventional robot control system to measure the *relative motion* between robot end-effector and target object. The vision sensor was one of the low cost, but high accuracy sensors. In 1973, Shirai and Inoue described how a visual feedback loop can be used to correct the position of a robot to increase task accuracy. In 1979, the term “visual servoing” has been first introduced by Hill and Park to distinguish their

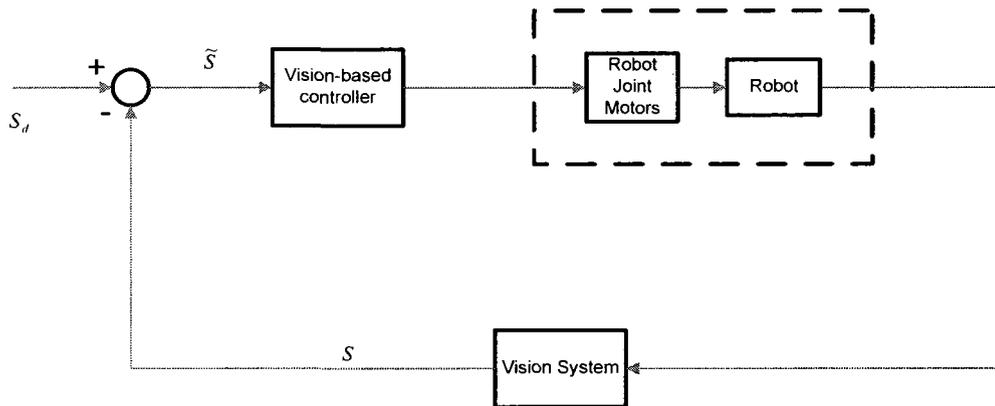
approach from earlier “block world” experiments. So far, the vision-based control or visual servoing has become one of the most significant and popular fields in robot control. There are several types of visual servoing depending on different classification methods.

Based on the different position of camera, visual servoing systems can be classified into two types: eye-in-hand and eye-to-hand. In eye-in-hand systems, the camera is fixed in the robot end-effector and move together with the end-effector. The camera can only observe the target object. In eye-to-hand systems, the camera is fixed in the workspace and can observe both the robot and target object.

According to the different controllers, there are two architectures: dynamic look-and-move system and direct visual servo system. In the first type, there are two controllers. One is called visual controller which is used to control the velocity of the camera and the other one is the robot joint controller which is used to control the motors in the robot. In the second controller, there is no joint level controller. It means that the visual controller will directly calculate the joint inputs and use vision information to stabilize the robot. In real industries, the dynamic look-and-move system is much more popular than direct visual servo system.



**Figure 1-2 Dynamic look-and-move system**



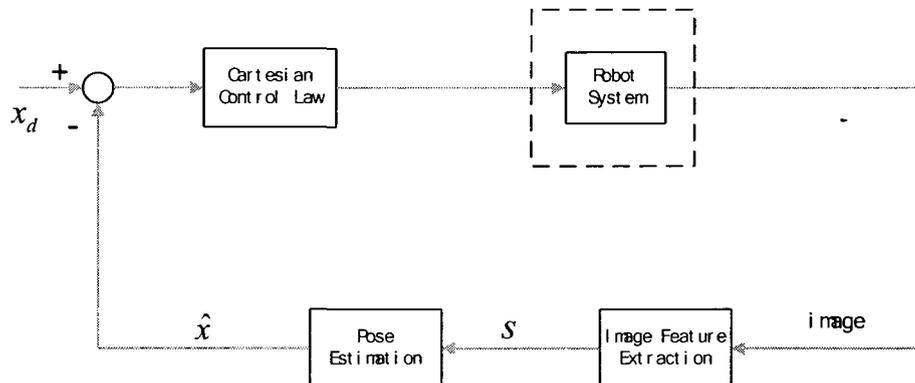
**Figure 1-3 Direct visual servo system**

The third major classification of visual servoing system is position-based control and image-based control. In position-based visual servoing (PBVS), shown in Figure 1-4, the feedback is not image features but the estimated pose of target object with respect to the camera frame. This pose is obtained from image features. The relative direct position-based control was first proposed in [21]. The error of the initial position and the desired position is defined in the 3D Cartesian space. Therefore, the trajectory of end effector is controlled to be an approximately straight line in the Cartesian space. However, position-based control is a “model-based” method. It

required the perfect geometric model of the target object for pose estimation. Second, it needs high accuracy of camera calibration and hand-eye calibration.

For the image-based visual servoing (IBVS), shown in Figure 1-5, the feedback is the errors of image features directly. It will be introduced in detail in Section 1.2.2.

Furthermore, a hybrid system (HVS) combining PBVS and IBVS is also introduced in robot control field. The image features and the estimated pose of target object have been mixed to control the trajectories of end effector in both the Cartesian and image space. The most well-known hybrid method is 2-1/2-D method [11]. The system block diagram is shown in Figure 1-6. The 2-1/2-D method combined the advantages of position-based control and image-based control, including “model-free”. The advantages of the method are that the trajectories of end effector in both the Cartesian and image space are simultaneously straight lines. Nevertheless, there are some shortcomings for hybrid system. First, it is necessary to find at least 4 and 8 different feature points for a planar and non-coplanar target object respectively. Second, it also requires partial post estimation. Third, it still needs to consider the image boundary and robot singularity.



**Figure 1-4 Block diagram of PBVS**

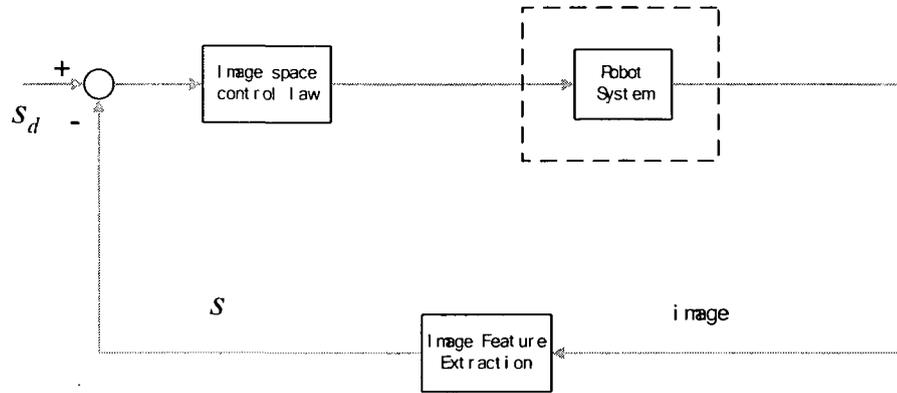


Figure 1-5 Block diagram of IBVS

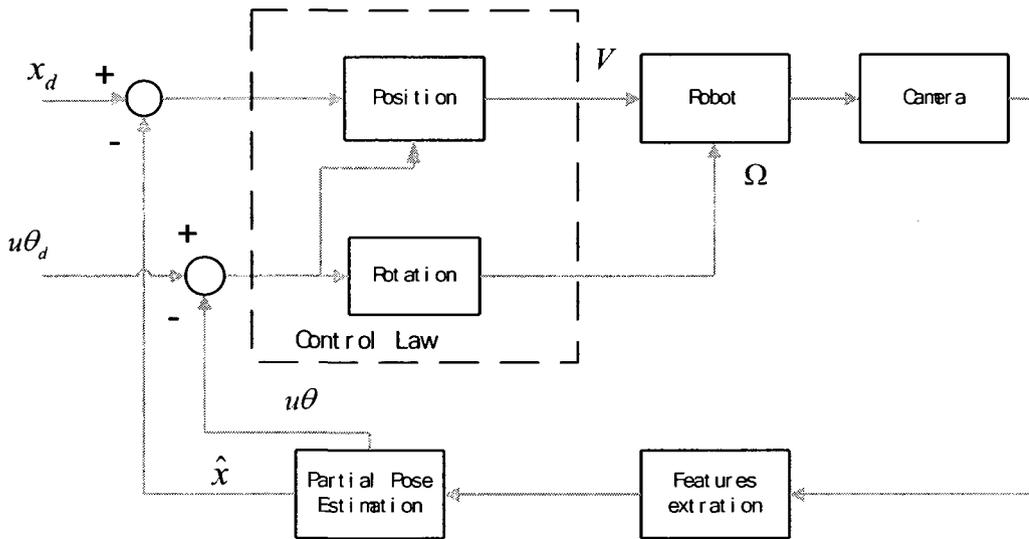


Figure 1-6 Block diagram of HVS (2-1/2-D method [11])

In summary, the classifications of visual servoing are shown in Table 1-1:

Table 1-1 Classification of visual servoing

Methods for Classification	Class
Position of the camera	Eye-in-hand
	Eye-to-hand
The controller	dynamic look-and-move
	direct visual servo

Feedback error	PBVS
	IBVS
	HVS

Because most of the real industries use the Eye-in-hand dynamic look-and-move, in this thesis, we will focus on this kind of visual servoing system. Moreover, since the IBVS does not need the post estimation, we will choose IBVS system to do our research.

### 1.2.2 Background of image-based visual servoing (IBVS)

As described above, the feedback signal of the IBVS comes from the image features errors directly. Assume that  $s$  is a vector of image features and  $s_d$  represents the desired image futures. The task of the IBVS is to drive the error  $\tilde{s} = s_d - s$  to zero, which implies that the pose of robot end-effector with respect to the target object is also controlled indirectly. Therefore, when the error of image features goes to zero, the error of robot kinematic also approaches to zero. Comparing with position-based visual servoing (PBVS), the advantages of IBVS are obvious.

First, it is a “model-free” method [3]. It does not require the model of target object. Second, the IBVS is robust to camera model errors [4]. It is not sensitive to the camera calibration error. Third, in image plane, the trajectories of image features are controlled approximately straight lines [5]. It means that, the trajectories of image features will not go beyond the image boundary during the whole operation if both the initial and desired features are set within the field of view (FOV).

However, IBVS also has some drawbacks. First of all, since the control law is almost based in the image plane, even though the trajectories of image features are merely the straight lines, the trajectories of end effector in Cartesian space are not straight lines. Besides, there might be image singularities in the Jacobian matrix. The controller of the IBVS contains the mapping between velocities in image space and in robot space. This mapping is embodied in the image Jacobian matrix. Any singularities or poor conditions in the image Jacobian matrix may lead to potential failure of visual servoing. Meanwhile, because of the nonlinearity and singularities of the image Jacobian matrix, the image local minima which were first introduced by F. Chaumette [5] will also lead to the servoing failure. In order to conquer these disadvantages and apply the IBVS in more and more fields, many methods and algorithms have been proposed. Cowan and Knditschek [9] gave a global stabilizing algorithm for a planar camera motion. Koichi Hashimoto and Toshiro Moritsugu [10] proposed a potential switching scheme to avoid the image local minima. To solve the problem of trajectories in image space and Cartesian space, Chaumette and Malis et al. [11] proposed a 2-1/2-D visual servoing. The main idea of this method is to consider both the image features and the camera orientation parameters in the controller. In [15], stereo vision was introduced into the IBVS to calculate the exact image Jacobian matrix. In [12] [49], a new partitioned approach to the IBVS is proposed to solve the indirect and seemingly contorted trajectories in Cartesian space. In this method, the z-axis rotational and translational components of the control were decoupled from the remaining degrees-of-freedom and the singularities have been avoided successfully.

In addition, the researchers never stop developing the IBVS in different fields. It was first applied to tracking a moving object by Koichi Hashimoto et al. [13]. The coordinates of points were expressed in a cylindrical coordinate system in [21] to improve the robot trajectory. In [14], a visual servoing scheme was proposed to align an airplane with respect to a runway. A linearized model of the airplane dynamics and decoupled visual features were designed to build the control scheme. Recently, the application of the IBVS has been further extended widely. In [50], IBVS has been used in the parallel robot. IBVS has also been combined with fuzzy neural networks in [51].

### **1.2.3 Background of image features extraction**

An image feature is generally defined as any measurable relationship in an image. From the background of the IBVS introduced above, it can be found out that the image features are used not only to indicate the relative pose of the end-effector with respect to target object, but also to generate the feedback signal of the whole visual servoing system. In this case, it is significant to determine a proper set of image features to obtain an optimal behavior of the system. In the past, the geometric features such as points, segments or straight lines [1] are usually chosen as the image features and utilized as the inputs of controllers. This kind of image features is very easy to be detected. However, it can only be applied to some limited objects. Also, it

is very easy for these features to be lost during the IBVS operation. Let us take the rectangle object as an example and consider the four corners as the image features. During the process, one or two corner points might be covered by some unexpected objects such as human hands or some external objects. In this case, the number of image features and that of desired image features are no longer matching, which will lead to the servoing failure. In recently years, in order to apply the visual servoing technology to track complicated objects and enhance the robustness, several novel features were adopted. For example, laser points [15, 37], the polar signature of an object contour [16], and, as in this paper, the image moments [17] are used as image features.

Image moments have been widely used in computer vision for long time, especially in pattern recognition and 3D reconstruction. They can be easily calculated from a binary image. If some part of target object has been covered, it might affect the accuracy of the result but will not lead to the failure directly. Furthermore, if redundant image features points are used, the potential problem may occur [5] while using image moments as the image features can avoid this problem. According to the definition and properties of image moments, low order image moments can represent the object in the image, for example, the area, the centroid coordinates, the orientation of principal axes, the radii of gyration, projection skewness and so on [18]. Besides, image moments also have some properties called moment invariance. These invariants are composed of moments and invariant with respect to the change of scale, position and orientation. The most famous invariants are Hu's invariants [19]. They

were derived by M. K. Hu who performed the first significant work of considering moments for pattern recognition. In IBVS, the moment invariants also play an important role. The invariant property allows the image features composed of invariant to stay the same during 3D translation and 2D rotation, which helps a lot in motion decouple and construction of the control. Based on this theory, the moment invariants can be chosen as effective features [17], [20].

#### **1.2.4 Background of image pre-processing for visual servoing**

In visual servoing, all the image features are extracted from the image taken by the camera. Therefore, the quality of the image will affect the accuracy or even the success of visual servoing system. In the real industrial environment, the good illumination cannot be guaranteed. Also, sometimes, the dust and speckles in the field of view (FOV) will influence the features extract algorithm to obtain the correct features. In order to overcome these problems, image pre-processing is required before the image features extraction in the visual servoing system. A method for removing variation in illuminations was introduced in [24], but it only works when the image foreground is lighter than the background. In [25], this method has been extended to deal with the lighter background situation. In this algorithm, a parameter  $k$  was determined to obtain the optimal result and the average grayscale intensity of image was considered. However, it failed when the illumination was uneven.

### 1.3 Motivation and Contribution

As mentioned above, choosing a proper set of image features is significant for the system performance. Although image moments have already been used in IBVS and the moments which represent the area, centroid and orientation respectively work very well, there are still some features that are need to be improved. In [17], the features that were used for controlling the rotation are derived from Hu's invariants. However, since these features have high orders, they are very sensitive to the noise. In addition, by testing these two features used in [17], the results show that they can only represent the correct pose for few objects. It means that, for many other objects, these two features cannot show the accurate rotation of the object around X axis and Y axis. Meanwhile, a new set of image features are defined in [20] based on the invariants from [22]. After combining the invariants, the authors choose 10 new features and two of them are used for the control input. Although a special error has been defined and a function of the special error is proposed to choose the appropriate features, how to choose the proper two features which represent the pose of object efficiently is still a complicate and time-consuming work. Moreover, if the target object has been changed, the designer has to recalculate the error and function value, and then re-choose the two features. In this case, a set of simple image features which can represent the pose correctly and do not need to change for different objects, are the major challenge for visual servoing.

Besides the choice of image features, there is another key issue which influence the performance of system. It is the depth estimation. A number of methods have been proposed to solve the depth estimation problem in visual servoing. The classical one is proposed by C. C. W. Hulls in 1996 [23], which uses the standard computer vision technique. In [24], an EKF-based online depth estimation method has been presented. Furthermore, one can assume that the depth is constant and set the value before starting the visual servoing. However, if the motion of camera is not parallel to the target object, this method will fail. To simplify the complexity of depth estimation, a novel method based on homogenous transformation will be proposed in this thesis.

To summarize, the research contribution includes four parts which will be presented as follow:

1. An improved image pre-processing algorithm has been introduced before image features extraction step to improve the image taken by eye-in-hand camera. Since the unexpected noisy and bad illumination will affect the accuracy of image features extraction, the image pre-processing algorithm can increase the veracity of image features by accommodating the illumination condition and removing the dust and speckles. The improved algorithm has been demonstrated in Matlab to preprocess the images with different initial conditions.
2. An ameliorative IBVS algorithm using image moments as the image features has been proposed. A new set of image moments have been chosen to control the rotation of the camera. They can not only

represent the pose of target object rightly, but also can be applied to represent the objects with different shapes. The proposed algorithm has been simulated in Matlab/Simulink and the derivation of image Jacobian matrix is provided.

3. A novel depth estimation method has been introduced. It is derived from the homogeneous transformation matrices between the robot frame (world frame) and the camera frame. Compared to other depth estimation methods, this method is very simple and easy to compute. The method has been implemented in Matlab/Simulink and simulation results show that the approximate depth can be estimated correctly and efficiently.
4. In order to decouple the obtained Jacobian matrix for controlling the motion of camera with individual image features, a four-stage control algorithm for visual servoing has been designed. The control of 2D rotation and other moments including area, centroid and orientation has been decoupled.

## **1.4 Thesis organization**

This thesis is composed of six chapters. The general content for each chapter is introduced as follows:

Chapter 1: introduces the background and basic concept of visual servoing and reviews the previous works. Also, the motivation and contribution of the research have been presented.

Chapter 2: introduces the background of morphological image processing. In this part, the basic ideas of morphological operations including gray-scale erosion, dilation, opening and closing have been explained. An image pre-processing algorithm is proposed. Also, the test results are provided.

Chapter3: Introduces the method of extracting the image features from the image. The definition of image moments will be given. After choosing the proper set of image features, the derivation of image Jacobian matrix is provided. Moreover, the method of depth estimation is showed in this chapter.

Chapter 4: The traditional IBVS system will be introduced first. Then, the four stages sequence control. The details of control system design will be explained step by step.

In Chapter 5: The simulation results for different poses and shapes will be shown. The analyses and comparisons will also be given.

Chapter 6: Conclusions and future works will be provided.

## **1.5 Summary**

In this chapter, the basic background of visual servoing has been introduced, including the fields of applications and classifications. Specially, the background of IBVS has been addressed, including the advantage, the disadvantage and the previous

progress. Besides, the background and previous work of image features extraction and image pre-processing have been indicated. The motivation and contribution of the research have been presented. In the end, the organization of this thesis is described.

## CHAPTER 2 MORPHOLOGICAL IMAGE PROCESSING

### 2.1 Introduction

In this chapter, the background of morphological image processing is introduced. The definitions of basic morphological operations including gray-scale erosion, dilation, opening and closing are explained first. In order to improve the quality of images which is used to extract image features, a novel method of pre-processing will be proposed in this chapter. After applying the proposed image processing algorithm, it is shown that the illumination condition of the image will be improved significantly and the unexpected sparkles and dirt will be removed effectively. The simulation results of proposed method will be provided under various circumstances.

### 2.2 Background of morphological image processing

In biology field, morphology is the study of form and structure in both plants and animals. In the image processing domain, mathematical morphology refers to a branch of nonlinear image processing and analysis geometric structure within an image [24]. The mathematical morphology which is based on set theory, lattice theory, topology, and random functions was originally developed by Georges Matheron and Jean Serra in 1964 [26]. At first, the mathematical morphology was applied to the

binary image. Later, from mid-1970s, it has been extended to grayscale image. In this thesis, the grayscale morphological operations will be utilized.

In grayscale morphology, the image is considered as a function  $I$  which is a function mapping an Euclidean space or grid  $E$  into  $\mathfrak{R} \cup \{\infty, -\infty\}$  where  $\mathfrak{R}$  is a set of real numbers. The basic idea of morphology is to probe the image with a structuring element which is also a function defined as  $g$ , and draw conclusions on how the element fits or misses the shapes in the image.

### 2.2.1 Gray-scale dilation and erosion

Before go through the details of the image pre-processing algorithm, some basic morphological operations will be introduced first. It should be noticed that in this chapter, the operations “+”, “-”, “\*” and “/” represent the regular addition, subtraction, multiplication and division respectively. Denote a gray-scale image in a Euclidean space or an integer grid as  $I$  and the structuring function as  $g(x, y)$ .

- **Dilation**

The grayscale dilation of  $I(x, y)$  by  $g$  is given by [47]

$$(I \oplus g)(s, t) = \max \{I(s - x, t - y) + g(x, y) \mid (s - x, t - y) \in D_I, (x, y) \in D_g\} \quad (2.1)$$

where the “ $\oplus$ ” denotes the dilation operation and  $D_I, D_g$  denote the domains of function  $I(x, y)$  and  $g(x, y)$  respectively.

There are two possible results of the dilation operation. First, if all values in structuring element are positive, the output image will become lighter than the original one. Second, if the sizes of dark details in original image are smaller than the structuring element, these details will be weakened or removed.

- **Erosion**

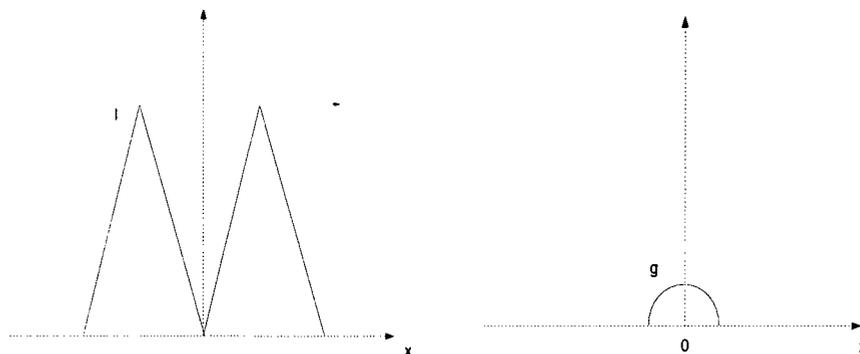
The grayscale erosion of  $I(x, y)$  by  $g$  is given by [47]

$$(I \ominus g)(s, t) = \min \{ I(s + x, t + y) - g(x, y) \mid (s + x, t + y) \in D_I, (x, y) \in D_g \} \quad (2.2)$$

where the symbol “ $\ominus$ ” denotes the erosion operation and  $D_I, D_g$  denote the domains of function  $I(x, y)$  and  $g(x, y)$  respectively.

There are also two possible results of the erosion operation. First, if all values in structuring element are positive, the output image will become darker than the original one. Second, if the sizes of light details in original image are smaller than the structuring element, these details will be weakened or removed.

The effects of dilation and erosion are shown as follows by considering  $I$  as a 1-D function:



(a) image function  $I$

(b) structuring element  $g$

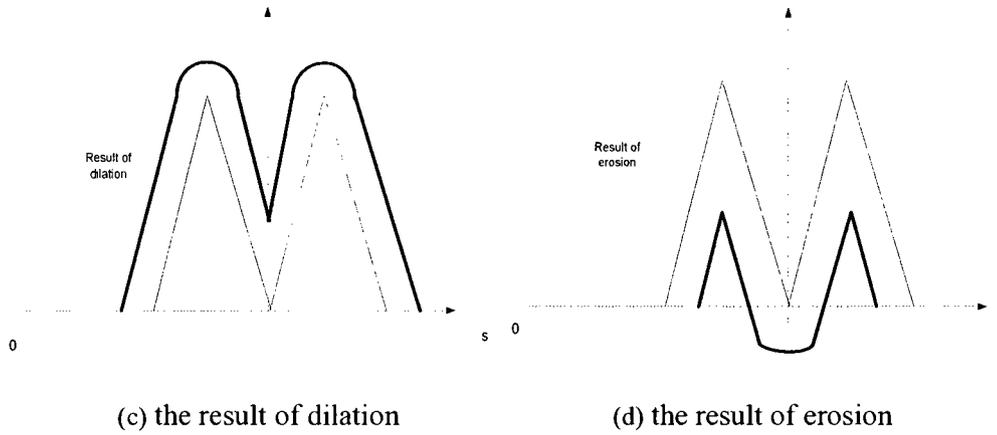


Figure 2-1 Dilation and erosion [24]

### 2.2.2 Gray-scale opening and closing [47]

- **Opening operation ( $\circ$ )**

The grayscale opening operations of  $I$  by  $g$  is given by

$$I \circ g = (I \ominus g) \oplus g \tag{2.3}$$

The main idea of opening operation is to remove the small light details and reduce the brightness in the first erosion step, then, resume the brightness in the second dilation step.

- **Closing operation ( $\bullet$ )**

The grayscale closing operations of  $I$  by  $g$  is given by

$$I \bullet g = (I \oplus g) \ominus g \tag{2.4}$$

The main idea of closing operation is to remove the small dark details and increase the brightness in the first dilation step, then, resume the brightness in the second erosion step.

## 2.3 Proposed Image Pre-processing Algorithm

The well-known Open Top-Hat Transform in morphological image processing defined as [47]

$$T(I) = I - I \circ g \quad (2.5)$$

which is very useful to adjust uneven illumination. In the equation, the “ $\circ$ ” is the opening operation. However, it can only solve the problem that the background is darker than the target object. However, in most visual servoing applications the background is lighter than the object. In order to deal with this situation, an improved method can be proposed as follow:

- **Step 1:** Read in a RGB image taken by the camera and transform it into a gray-scale image.

After obtaining the gray-scale image, the gray-scale morphological operations can be applied to enhance the image.

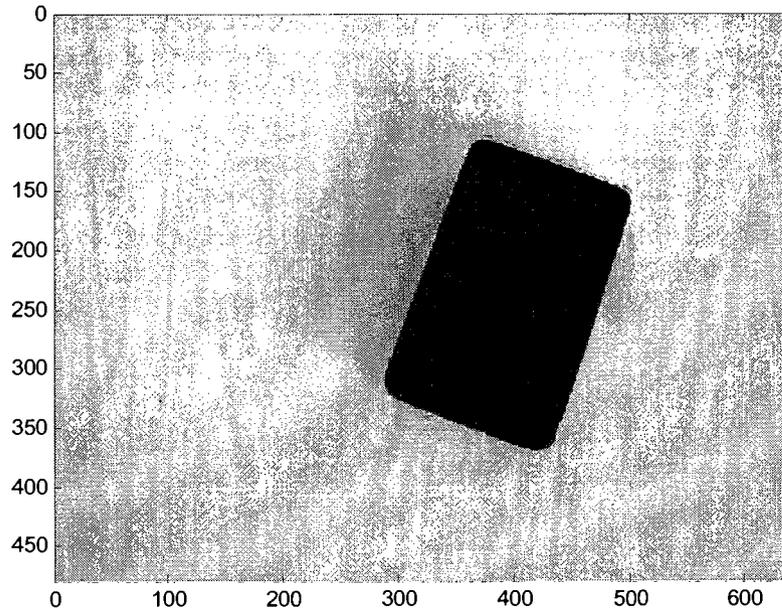
- **Step 2:** Image Pre-processing part.

In this step, gray-scale closing will be chosen as the image processing operator. Assuming  $I_i$  represents the initial image and  $g$  represents the structuring element.  $I_f$  is the final image, which could be obtained by

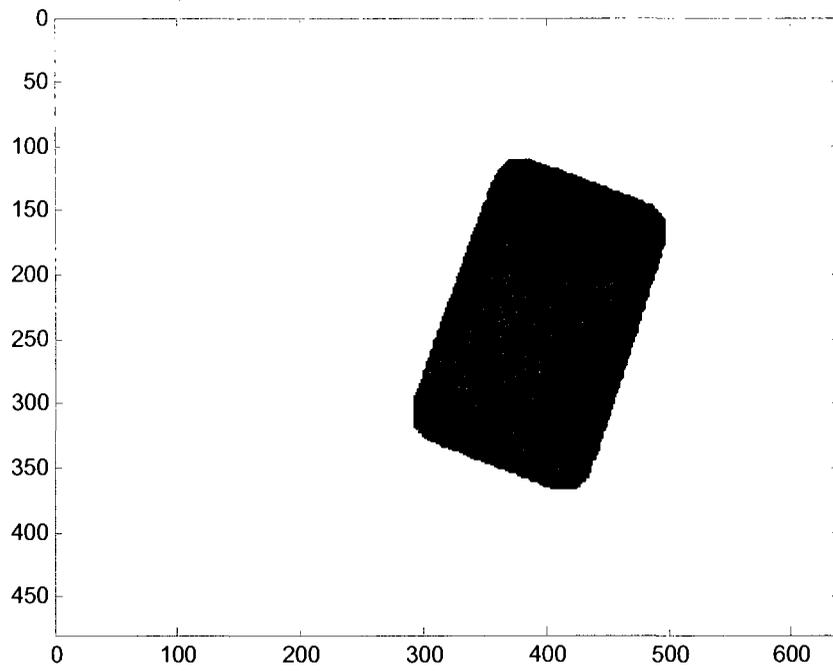
$$I_f = k * (I_i + (I_i \bullet g)) \quad (2.6)$$

where “ $\bullet$ ” is closing operation,  $k$  is a proportional gain and  $g$  is chosen as a disk with radius of 10. In order to choose the proper  $k$ , a lot of tests have been done. The results showed that with the increase of  $k$ , the gray-value of each pixel becomes larger

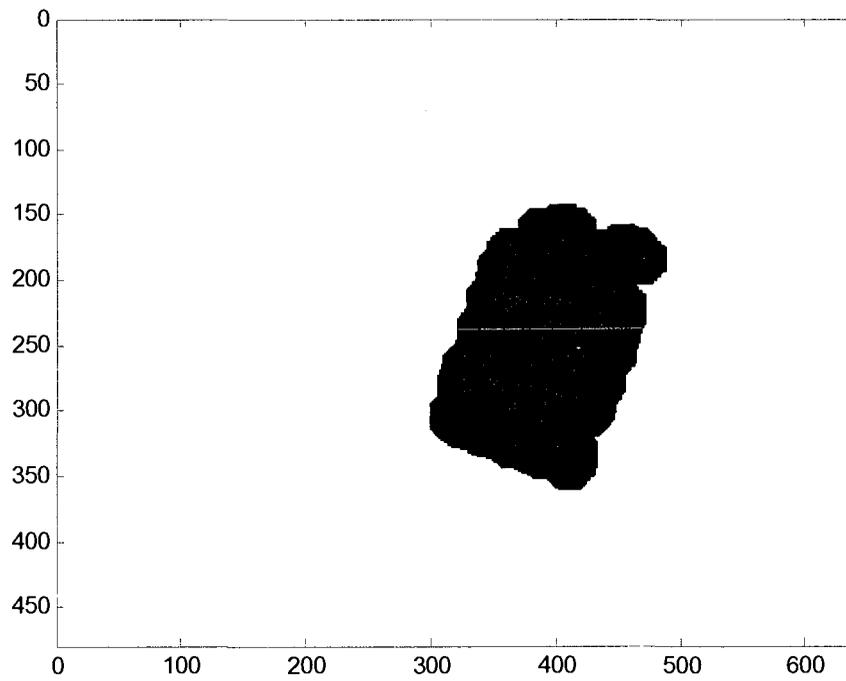
and the final image becomes brighter, which is easier for the subsequent segmentation. However, accordingly, the target object in  $I_f$  becomes smaller. Figure 2-2 shows the initial image where there is a rectangle target object. Figures 2-3, 2-4 and 2-5 show the final images with different  $k$ .



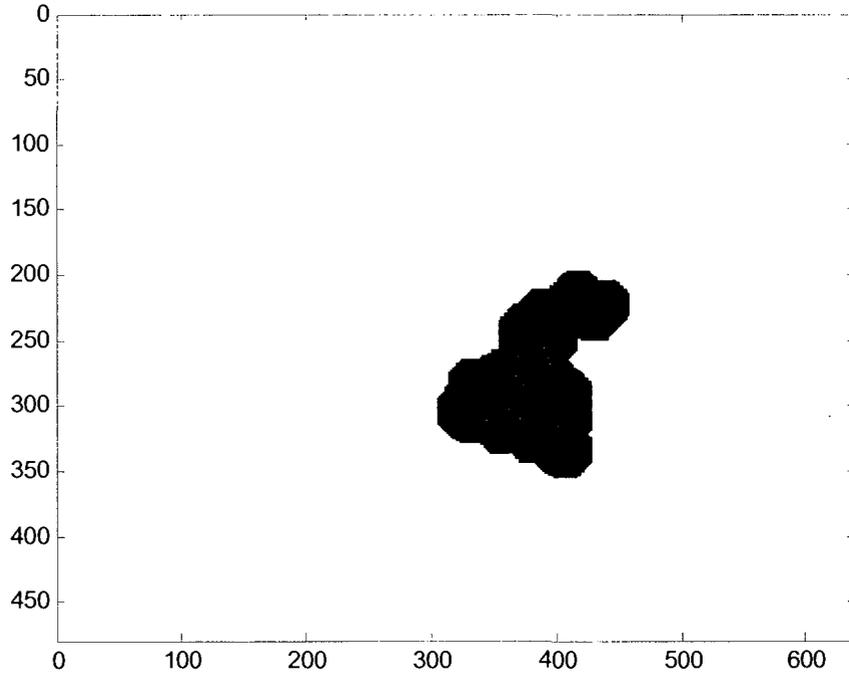
**Figure 2-2 The initial image**



**Figure 2-3 Result with k=1**



**Figure 2-4 Result with k=10**



**Figure 2-5 Result with k=15**

Finally, the following equation has been summarized by experience:

$$I_f = \frac{255}{I_{mean}} * (I_i + (I_i \bullet g)) \quad (2.7)$$

where  $I_{mean}$  is the average grayscale value of the all the pixels of initial image  $I_i$ . If the initial image is dark, the value of  $I_{mean}$  will be small. Then the gain  $k$  will be large enough to enhance the image. In the other hand, if the initial image is bright enough, the  $k$  will stay small so that the shape of object will be maintained correctly.

● **Step 3: Segmentation**

A single threshold method called “Otsu’s method [27]” has been utilized:

If consider  $t$  as the initial threshold which will divides the image into two parts, background and object. Assume  $W_b$  is the proportion of background in the

image and  $W_o$  presents the proportion of object in the image.  $U_b$  and  $U_o$  are mean gray values of background and object.  $U = W_b * U_b + W_o * U_o$  is the mean gray value for the whole image. At the same time, according to the definition of expected value,  $U$  is the expected value of  $U_b$  and  $U_o$ . Notice that  $W_b$ ,  $W_o$ ,  $U_b$ ,  $U_o$  and  $U$  are all the functions of  $t$ .

Meanwhile, according to the definition of variance,

$$\Omega = E((X - E(X))^2) \quad (2.8)$$

where  $E(X)$  is the expected value of  $X$ .

The between-class variance for the whole image should be

$$\Omega = W_o * (U_o - U)^2 + W_b * (U_b - U)^2 \quad (2.9)$$

When  $\Omega$  goes bigger, it means the difference between background and object becomes bigger.

The main idea of Otsu's method is that changing  $t$  from 0 to 255 and calculating the values of  $\Omega$  for each  $t$ . Find out the largest  $\Omega$ , then the corresponding  $t$  is the desired segmentation threshold.

- **Step 4:** Remove speckles caused by segmentation by using closing operation.

After the image processing, the final image can be used to extract the image features effectively and correctly, which will be described in detail in Chapter 3.

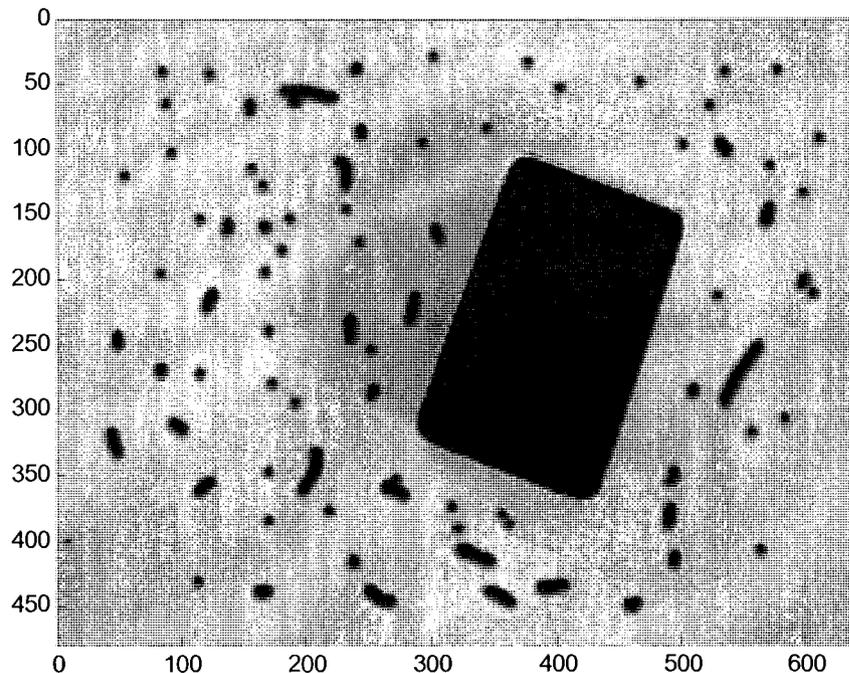
## 2.4 Simulation results

The proposed image pre-processing algorithm has been tested under three

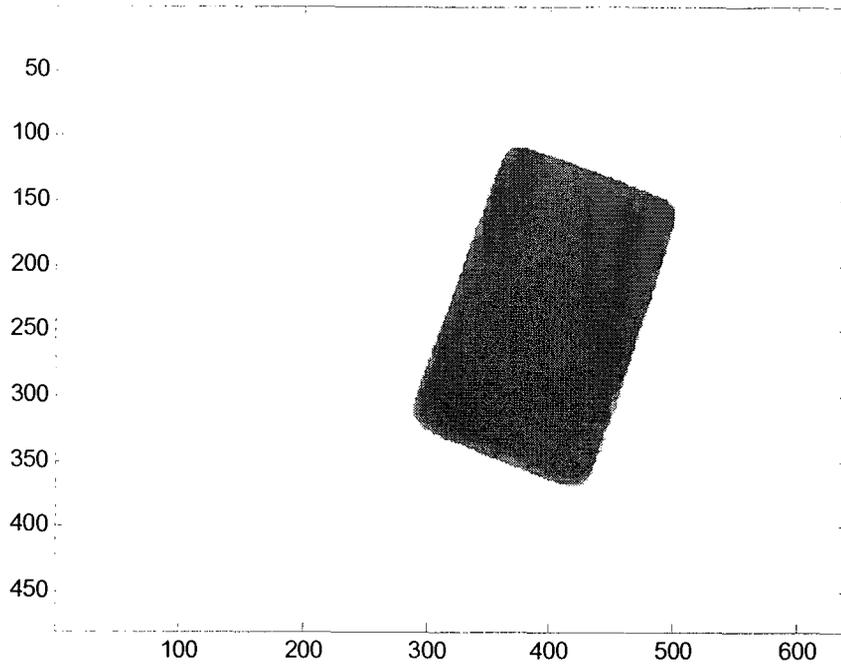
circumstances.

1. In the real industrial environment, the workspace might always be affected by unexpected dust or dirt. The followings test show whether the proposed algorithm could remove speckles under good illumination condition.

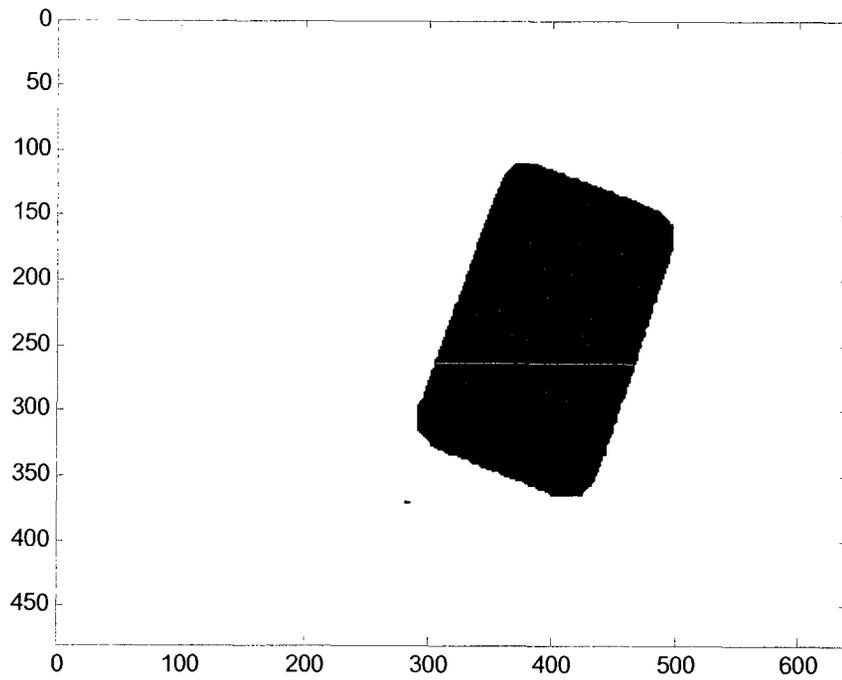
As shown below, Figure 2-6 is the initial image in which there are a lot of speckles and dirt spots. Figure 2-7 shows the temporary result by the end of Step 2. It is clear to notice that the unexpected spots have been totally removed. After that, in Figure 2-8, the image has been segmented into a binary image which is perfect for extracting image features.



**Figure 2-6 The initial image with unexpected speckles**



**Figure 2-7 The image obtained after step 2-pre-processing**



**Figure 2-8 The final image**

2. In this test, the illumination in work environment is not good enough. The obtained images may be too dark to be utilized to extract the image features directly. The following figures show the result of enhancing the bad illumination.

In the initial image which is shown in Figure 2-9, the illumination is very weak and it is difficult to figure out the exact position and shape of the target object. Figure 2-10 shows the result after step 2. At this time, the image has been improved and the object is much clearer. In the final image shown in Figure 2-11, the image has been segmented into a binary image in which the target object is displayed clearly.

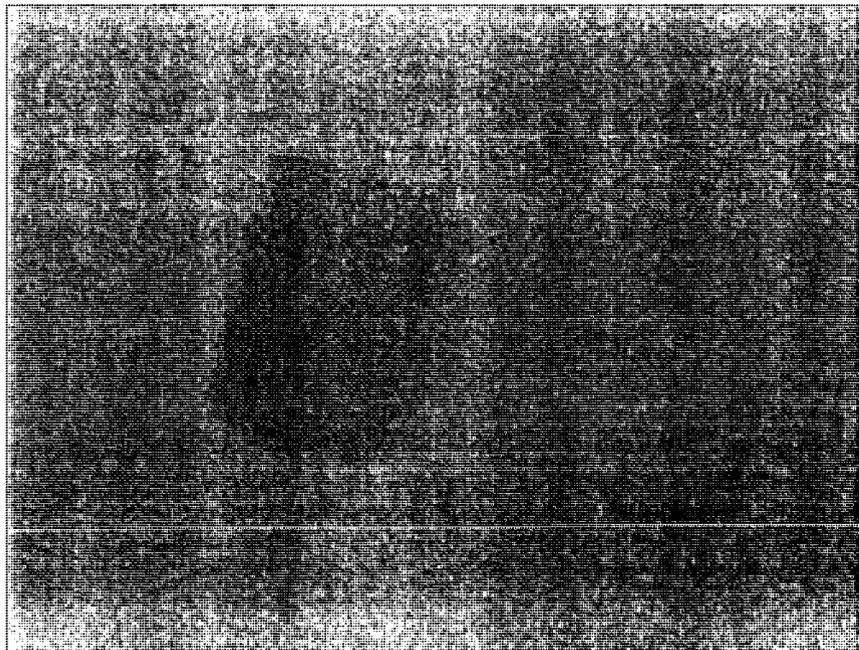
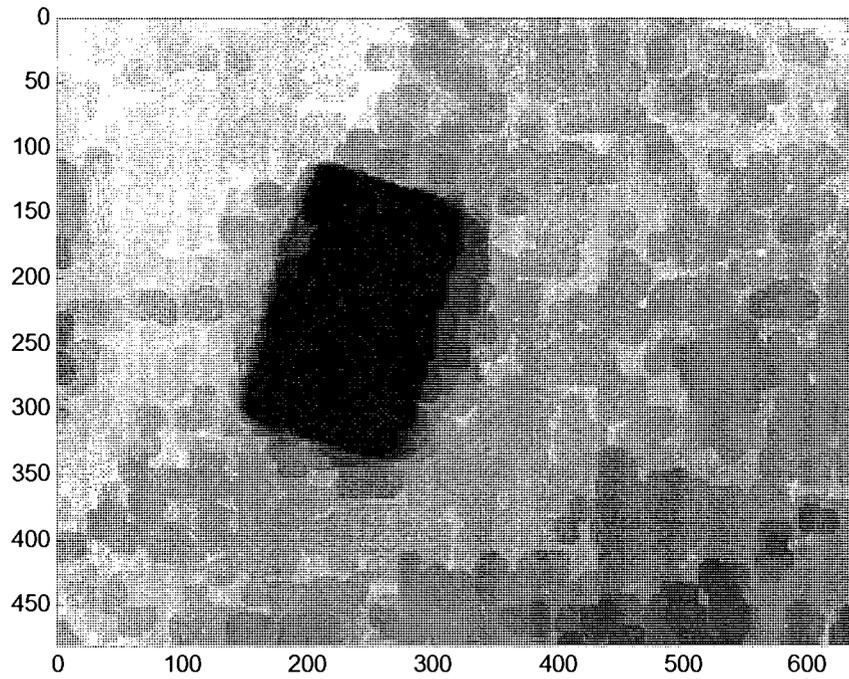
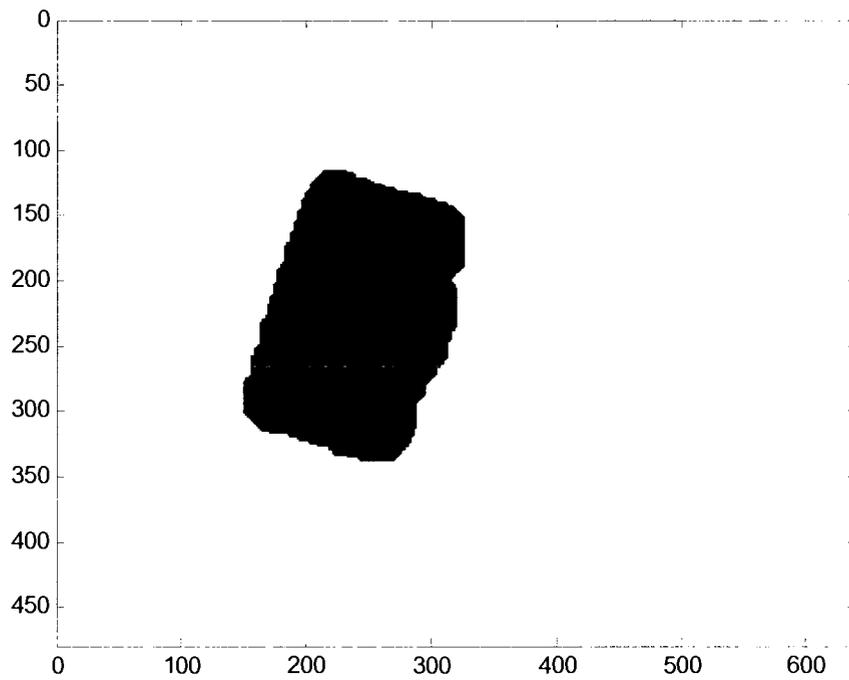


Figure 2-9 The initial image with bad illumination



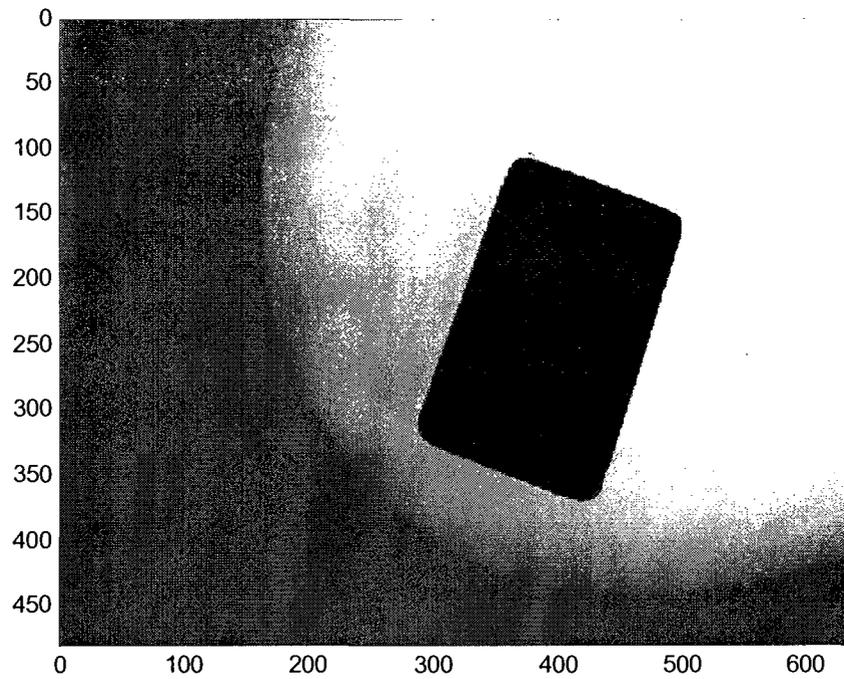
**Figure 2-10 The image obtained after step 2-pre-processing**



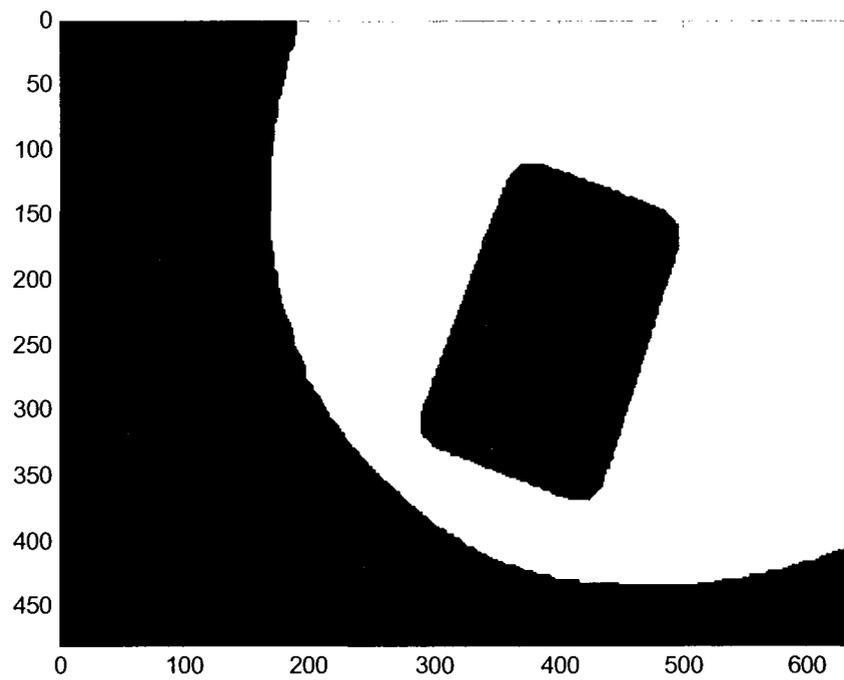
**Figure 2-11 The final image**

3. Besides the bad illumination condition, the uneven illumination could affect the visual servoing result too. The results of balancing uneven illumination are shown as follows. In this test, the proposed algorithm has been compared with another algorithm described in [25]. The superiority of the proposed algorithm is obvious.

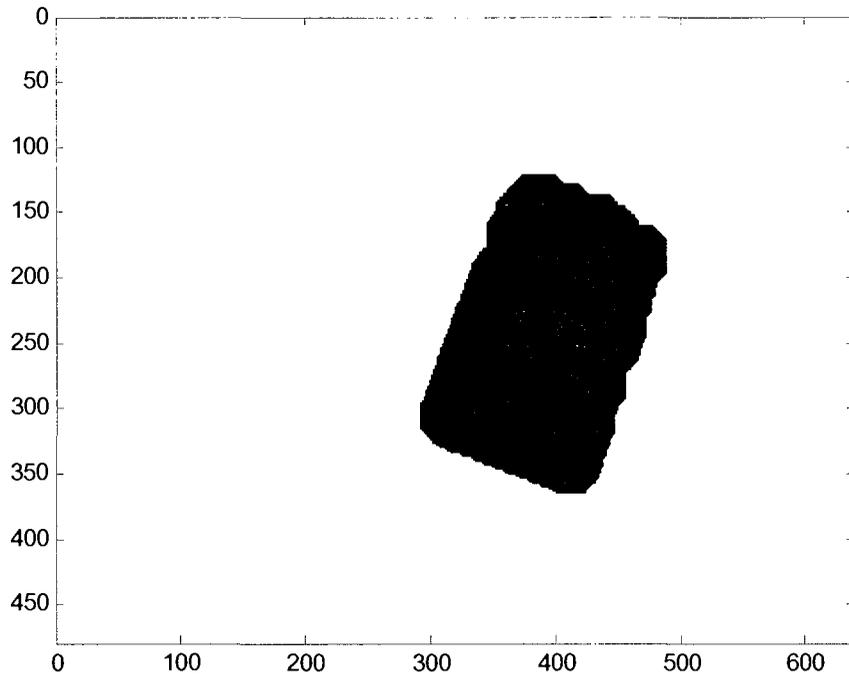
In Figure 2-12, the initial image is under the uneven illumination. The light source locates in the top right corner of the image, which causes that the partial image is partial dark and partial light. The result shown in Figure 2-13 uses the method mentioned in [25]. The segmented binary image not only has the target object, but also includes the dark background. It is obvious that the method used in [25] cannot balance the illumination successfully. Figure 2-14 shows the result using the proposed algorithm. The illumination has been balanced, and only the target object has been left in the segmented image. Therefore, the proposed image process algorithm performs better than the method used in [25], and the final image of proposed algorithm can be used for abstracting the correct image features.



**Figure 2-12 The initial image with uneven illumination**



**Figure 2-13 Result with use the algorithm proposed in [25]**



**Figure 2-14 The final image by using proposed method**

Based on the simulation results shown above, it can be concluded that the proposed image pre-processing method for visual servoing can work effectively. After removing the speckles and adjusting the illumination, the images can be applied to visual servoing.

## **2.5 Summary**

In this chapter, an improved method of image pre-processing for visual servoing has been introduced. By utilizing this method, the illumination condition of image can be ameliorated and the uneven illumination has been adjusted. Furthermore, the unexpected dirt and speckles can be removed effectively. After the

image pre-processing, the image is ready to be used to extract the image features efficiently.

## CHAPTER 3 IMAGE FEATURES EXTRACTION

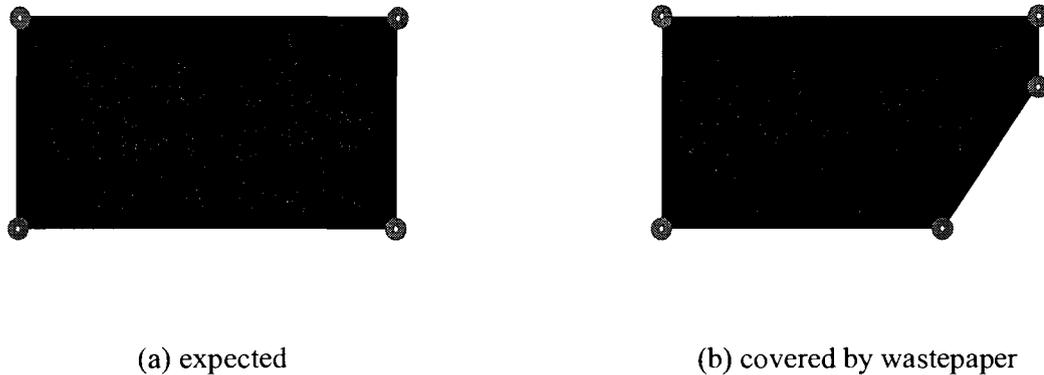
### 3.1 Introduction

In this chapter, six image features for IBVS will be chosen from the image moments to control the velocity in each dimension. In order to indicate the relationship between the velocity of image features in image frame and the motion of camera (end-effector) in the camera (end-effector) frame, the Jacobian matrix will be defined. The derivation of the Jacobian matrix based on the image moments will also be given. In addition, another significant parameter in the Jacobian matrix, the depth, will be estimated. A simple algorithm based on homogeneous transformation between different frames for estimating depth will be also proposed.

### 3.2 Image moments

As mentioned in Chapter 1, in traditional IBVS, the image features are always chosen as the corners or the special points of target object. The points are very convenient to be detected. However, it may lead to the system failure if one or more features are covered by unexpected object, such as human hands or wastepaper. For example, the surface of the target object is a rectangle and the expected image features are chosen as the four corners. During the visual servoing, if a corner has been covered by some wastepaper, the number of the image features calculated by the

traditional method will become five. Because the number of the image features associated with the desired position taught before is only four, the whole system will fail directly due to the mismatch between the number of desired and detected image features. Furthermore, if some part of the object has been located out of the field of view, the mismatch will also happen, which will cause the system to fail too.



**Figure 3-1 The image features in traditional IBVS of regular and irregular situations**

Besides the mismatch problem, the traditional image features can only apply on very limited shape of the target object which has enough corners. Let us take the date core shape as the example. In traditional way, only the two acmes could be chosen as the image features. If the object rotates around the x axis, the locations of these two image features will not change. Actually, the pose of the whole object has been changed. It means that only the two acmes can not represent the correct pose of the whole object.

In order to stabilize the visual servoing system and extend the type of target objects, image moments have been introduced for image feature extraction in the visual servoing [17], [29].

### 3.2.1 Background of image moments

Normally, the moment describes the numeric quantities at some distance from a reference point or axis. For an image with a density distribution function  $I(x, y)$ , where  $x$  and  $y$  are the coordinates in image frame, the two-dimensional geometric moment,  $m_{ij}$ , of order  $i + j$  is defined by [18]

$$m_{ij} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^i y^j I(x, y) dx dy \quad (3.1)$$

In order to describe the distribution of image with respect to its centroid, the central moments,  $\mu_{ij}$ , of order  $i + j$  is also given as follows:

$$\mu_{ij} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x - x_g)^i (y - y_g)^j I(x, y) dx dy \quad (3.2)$$

where  $(x_g, y_g)^T$  is the coordinate of centroid in image frame.

As mentioned in [18], the low-order moments have their own properties which can denote the geometric characteristics of target object in image.

- **Zero-th order moments [18]:**

The definition of zero-th order moment,  $m_{00}$ , of distribution function  $I(x, y)$

$$m_{00} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} I(x, y) dx dy \quad (3.3)$$

represents the total mass or the area of the given image. If the image is binary, the zero-th order moment indicates the area of the target object in the image.

- **First order moments:**

The two first order moments,  $\{m_{10}, m_{01}\}$ , are usually used to calculate the position of centroid of given image. They can be defined as [18]

$$\begin{cases} x_g = \frac{m_{10}}{m_{00}} \\ y_g = \frac{m_{01}}{m_{00}} \end{cases} \quad (3.4)$$

where  $m_{00} \neq 0$ . If the image is binary, the first order moments indicate the location of the target object's centroid in the image.

● **Second order moments:**

The second order moments are used to determine the principal axis of the object.

The orientation angle of principal axis can be defined as [18]

$$\alpha = \frac{1}{2} \arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right) \quad (3.5)$$

If  $\mu_{20} - \mu_{02} = 0$ , the orientation angle will depend on the value of  $\mu_{11}$ . If  $\mu_{11} > 0$ ,  $\alpha$  will be  $\pi/4$ . If  $\mu_{11} = 0$ ,  $\alpha$  will be 0. Otherwise,  $\alpha$  will be  $-\pi/4$  [18]. If the image is binary, the second order moment indicates the orientation of the target object in the image.

Besides the above mentioned three types of low-order moments, some other moments have their own geometric meanings: the third moments represent the projection skewness and the fourth moments relate to projection kurtosis.

Image moments not only can show the position or orientation of object, but also have some very useful properties. One of the most famous properties is invariance. According to Hu's invariance theory, some image moments called Hu's invariants (which are listed in the appendix A) are invariant to object scale change, position change or even orientation change. Based on this property, the invariant moments could be chosen as the image features and used to decouple the camera motions.

### 3.2.2 Choose the proper image features

In order to adjust and control the 6 DOF of camera, at least six image features should be chosen.

First of all, as mentioned before, since the low-order image moments can represent the geometric characters of the object perfectly, the image moment which represents to object's size, coordinate of centroid (on x axis and y axis respectively) and orientation angle respectively are chosen as the first four image features.

The rest two features, in some works [29], have been chosen as the well-known skewness. Moreover, to obtain the better decoupled behaviour, the invariant moments have been utilized in [17] where four image features (two sets) have been proposed. However, after the test and simulation, it shows that the first two image features used in [17] cannot represent the pose of object correctly all the time. And the other two features are very complicate and are of 4th orders. Since it is better to use moments of orders as low as possible to reduce the sensitivity of the control scheme to image noise, a better set of invariant which can tell the right pose of object and has lower order is introduced.

From the form of Hu's invariants, we choose two of them:

$$\begin{cases} M_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2 \\ M_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \end{cases} \quad (\text{refer to appendix A}).$$

For a given target object, no matter how the central moments  $\mu_{ij}$  change,  $M_2$  and  $M_3$  will stay the same. Recall the Pythagorean trigonometric identity:  $\sin^2 \theta + \cos^2 \theta = 1$ . It give us an inspiration to derivate the novel image features.

$$\text{Denote } \begin{cases} c_1 = \mu_{20} - \mu_{02} & s_1 = 2\mu_{11} \\ c_2 = \mu_{03} - 3\mu_{21} & s_2 = \mu_{30} - 3\mu_{12} \end{cases}, \text{ so that } \begin{cases} M_2 = s_1^2 + c_1^2 \\ M_3 = s_2^2 + c_2^2 \end{cases}. \text{ Assume}$$

the temporary image features are  $s_{(1+2)}$  and  $c_{(1+2)}$ . In order to make these two image features invariant to the 2D rotation and 3D translation at the same time, another parameter  $M_1 = \mu_{20} + \mu_{02}$  will be added to construct the image features.

In the end, the final improved set of features is

$$\begin{cases} s_x = 0.1 - (c_1 c_2 + s_1 s_2) / M_1^{\frac{9}{4}} \\ s_y = (s_1 c_2 - c_1 s_2) / M_1^{\frac{9}{4}} \end{cases} \quad (3.6)$$

Note that for large objects, the value of  $M_1$  will increase and the values of the image features  $s_x$  and  $s_y$  will be too small to be utilized for the visual servoing controller. Therefore, if the object is large, the parameter  $M_1$  will be defined by  $M_1 = \mu_{20} - \mu_{02}$ .

In order to compare the proposed image features with exit ones, Table 3-1 indicates the errors of first two image features in [17], denoted by  $P_x$  and  $P_y$ , and the errors of proposed image features in (3.6) with different poses for a normal rectangle object.

**Table 3-1 Comparison of  $P_x, P_y$  [17] and  $S_x, S_y$**

	$s_x, s_y$	$P_x, P_y$ [17]
$0^\circ$	$\begin{cases} e_{s_x} = 0 \\ e_{s_y} = 0 \end{cases}$	$\begin{cases} e_{P_x} = 0.15 \\ e_{P_y} = 0 \end{cases}$

Rotate around x axis by	10°	$\begin{cases} e_{s_x} = 0.00573 \\ e_{s_y} = -0.0001376 \end{cases}$	$\begin{cases} e_{p_x} = -0.004811 \\ e_{p_y} = -0.0004984 \end{cases}$
	20°	$\begin{cases} e_{s_x} = 0.006705 \\ e_{s_y} = 0.0006178 \end{cases}$	$\begin{cases} e_{p_x} = -0.044955 \\ e_{p_y} = -0.0005398 \end{cases}$
	30°	$\begin{cases} e_{s_x} = 0.01255 \\ e_{s_y} = 0.0006929 \end{cases}$	$\begin{cases} e_{p_x} = -0.09155 \\ e_{p_y} = -0.001494 \end{cases}$
	-10°	$\begin{cases} e_{s_x} = -0.005968 \\ e_{s_y} = 0.0001572 \end{cases}$	$\begin{cases} e_{p_x} = -0.01208 \\ e_{p_y} = -0.0005154 \end{cases}$
	-20°	$\begin{cases} e_{s_x} = -0.00687 \\ e_{s_y} = -0.0003059 \end{cases}$	$\begin{cases} e_{p_x} = -0.03754 \\ e_{p_y} = -0.0005861 \end{cases}$
	-30°	$\begin{cases} e_{s_x} = -0.01057 \\ e_{s_y} = -0.000962 \end{cases}$	$\begin{cases} e_{p_x} = -0.0983 \\ e_{p_y} = -0.00157 \end{cases}$
Rotate around y axis by	10°	$\begin{cases} e_{s_x} = 0.0005613 \\ e_{s_y} = 0.007008 \end{cases}$	$\begin{cases} e_{p_x} = 0.005713 \\ e_{p_y} = -0.001047 \end{cases}$
	20°	$\begin{cases} e_{s_x} = 0.0009638 \\ e_{s_y} = 0.007273 \end{cases}$	$\begin{cases} e_{p_x} = 0.04022 \\ e_{p_y} = -0.001166 \end{cases}$
	30°	$\begin{cases} e_{s_x} = 0.001228 \\ e_{s_y} = 0.007638 \end{cases}$	$\begin{cases} e_{p_x} = 0.0866 \\ e_{p_y} = -0.002376 \end{cases}$
	-10°	$\begin{cases} e_{s_x} = -0.00007224 \\ e_{s_y} = -0.007383 \end{cases}$	$\begin{cases} e_{p_x} = 0.0184 \\ e_{p_y} = -0.0009795 \end{cases}$
	-20°	$\begin{cases} e_{s_x} = 0.0006995 \\ e_{s_y} = -0.007901 \end{cases}$	$\begin{cases} e_{p_x} = 0.04589 \\ e_{p_y} = -0.001305 \end{cases}$
	-30°	$\begin{cases} e_{s_x} = 0.001063 \\ e_{s_y} = -0.01053 \end{cases}$	$\begin{cases} e_{p_x} = 0.08955 \\ e_{p_y} = -0.00241 \end{cases}$

Normally, to get the correct controller signal, the values of proper features should roughly change linearly with the linear change of object. For example, assume that when the object rotates 30 degrees around x axis and the error of the image feature relating to the rotation around x axis is A. Then if this object has rotated

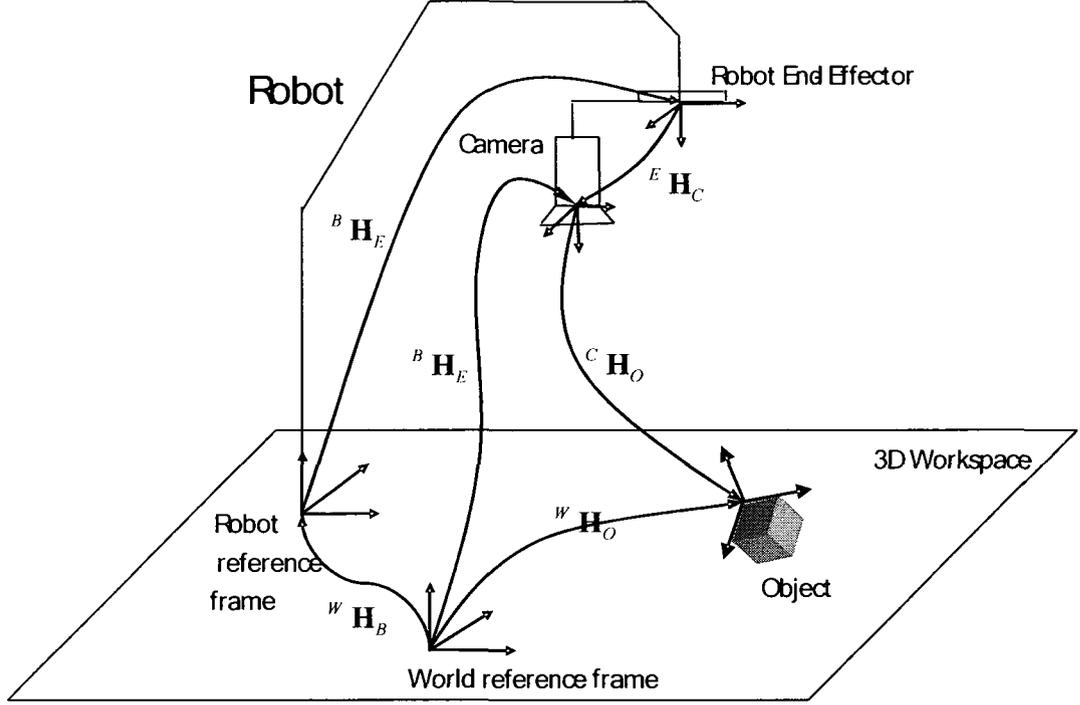
around x axis by -30 degrees, the sign of A should be changed as well to generate the visual servoing control signal with an opposite sign to drive the robot moving to a desired position. Referring to Table 3-1, it is clear that for the different pose, the proposed image features  $s_x$  and  $s_y$  can represent the pose correctly. For example, the object rotates around x by 20 degrees and the error of  $s_x$  will be 0.006705. If the object increases the rotation angle to 30 degrees, the error will also increase to 0.01255. Moreover, the features  $s_x$  and  $s_y$  can represent the pose around x and y axes respectively. That is,  $s_x$  is independent from  $\omega_y$  and  $s_y$  is independent from  $\omega_x$ . However, for the features  $P_x$  and  $P_y$  chosen in [17], they cannot represent the correct pose. For instance, if the object rotates around x axis by 30 degrees, the error of  $P_x$  will be -0.09155. Nevertheless, if the rotation angle has been changed to -30 degrees, the error of  $P_x$  will be -0.0983 which is still negative. Therefore, the features  $P_x$  and  $P_y$  cannot create the effective visual servoing control signals.

Based on the tests, the results show that the proposed image features can overcome the shortcomings of traditional ones and represent the object pose correctly.

### 3.2.3 Derivation of Jacobian Matrix

As discussed above, the six image features have been determined. However, how to describe the relationship between these features and the motion of the camera becomes the next main task. A briefly review of the establishment of the Jacobian matrix between the velocities of points in image frame and camera velocity will be

given first. Afterwards, the Jacobian matrix between the velocities of image moments in image frame and camera velocity will be derived.



**Figure 3-2 Basic IBVS frame**

Defining a point,  ${}^c P = [X, Y, Z]^T$ , in the camera frame, the corresponding projected point of  ${}^c P$  in image frame is denoted by  $f = [u, v]^T$ , where  $u$  and  $v$  are the pixel indices.

The velocity of camera is defined as

$$\dot{r} = [{}^c v_c \quad {}^c \omega_c]^T = [v_x \quad v_y \quad v_z \quad \omega_x \quad \omega_y \quad \omega_z]^T \quad (3.7)$$

The velocity of  $f$  is defined as

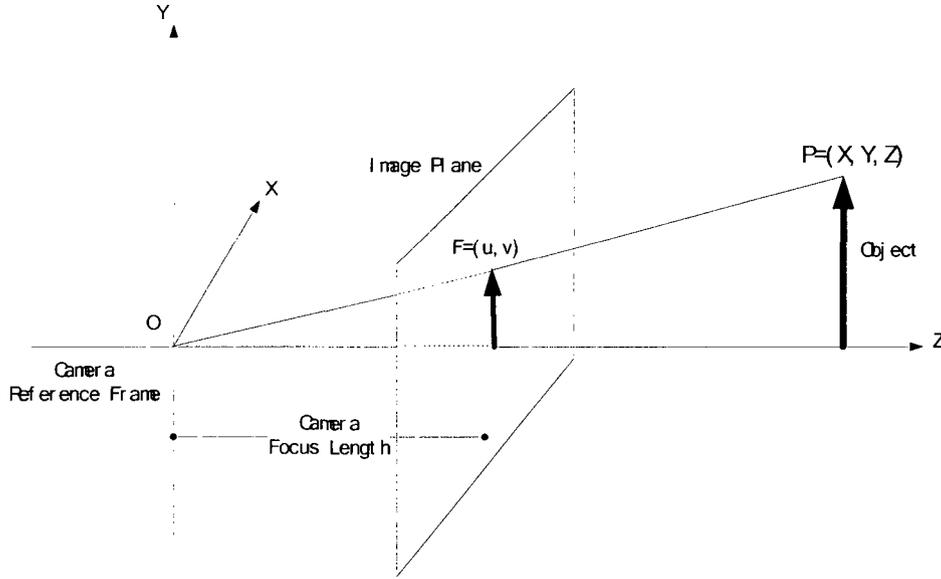
$$\dot{f} = \begin{bmatrix} \dot{u} \\ \dot{v} \end{bmatrix} \quad (3.8)$$

According to [1], the velocity of the point  ${}^c P$  is given by

$${}^c \dot{P} = {}^c \omega_c \times {}^c P + {}^c v_c \quad (3.9)$$

That is

$$\begin{cases} \dot{X} = Z\omega_y - Y\omega_z + v_x \\ \dot{Y} = X\omega_x - Z\omega_x + v_y \\ \dot{Z} = Y\omega_x - X\omega_y + v_z \end{cases} \quad (3.10)$$



**Figure 3-3 The perspective camera model**

In addition, based on the perspective projection, the relationship between  ${}^cP$  and  $f$  can be written by

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{\lambda}{Z} \begin{bmatrix} X \\ Y \end{bmatrix} \quad (3.11)$$

where  $\lambda$  is the focus length of camera.

Substituting (3.11) into (3.10), the derivatives of the coordinates of  ${}^cP$  in terms of parameters  $u, v$  is given by

$$\begin{cases} \dot{X} = Z\omega_y - \frac{vZ}{\lambda}\omega_z + v_x \\ \dot{Y} = \frac{uZ}{\lambda}\omega_x - Z\omega_x + v_y \\ \dot{Z} = \frac{Z}{\lambda}(v\omega_x - u\omega_y) + v_z \end{cases} \quad (3.12)$$

Taking derivative of both sides of (3.11), we obtain

$$\begin{cases} \dot{u} = \lambda \frac{Z\dot{X} - X\dot{Z}}{Z^2} \\ \dot{v} = \lambda \frac{Z\dot{Y} - Y\dot{Z}}{Z^2} \end{cases} \quad (3.13)$$

Note that here  $u$  and  $v$  are the pixel indices. To change it into the image coordinate, by assuming the effective size of the pixel equals to 1, we have

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} -1 & 0 & u_0 \\ 0 & -1 & v_0 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (3.14)$$

where  $(x, y)^T$  is the image coordinate for the point  $f$  and  $(u_0, v_0)^T$  is the pixel index of image center point. In this case, the coordinate of image center is  $(0, 0)^T$ .

Combining (3.12), (3.13) and (3.14), one can obtain the result:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} -\frac{\lambda}{Z} & 0 & \frac{x}{Z} & \frac{xy}{\lambda} & -\frac{\lambda^2 + x^2}{\lambda} \\ 0 & -\frac{\lambda}{Z} & \frac{y}{Z} & \frac{\lambda^2 + y^2}{\lambda} & -\frac{xy}{\lambda} \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (3.15)$$

In (3.15), the matrix which indicates the relationship between the velocity of point in image frame and the motion of camera is called Jacobian matrix, denoted by  $J_{image}$ .

The relationship

$$\dot{f} = J_{image} \dot{r} \quad (3.16)$$

is used to create the visual servoing feedback controller.

In this thesis, the image moments other than points are chosen as the image features. In this case, the most significant task in visual servoing control is to derive the Jacobian matrix which indicates the relationship between the motion of the image

features (the image moments) and the velocity of camera. In [17], the result of required Jacobian matrix has already been provided. However, there is no derivation and demonstration in detail. In this thesis, the calculation of Jacobian matrix of the image moments will be discussed step by step as follows.

According to the definition equation of geometric moment in (3.1), assume that  $R(t)$  represents the part of object projection in the image obtained by camera and  $C(t)$  is the contour of  $R(t)$ , the alternative equation of (3.1) is given by

$$m_{ij} = \iint_{R(t)} F(x, y) dx dy \quad (3.17)$$

where  $F(x, y) = x^i y^j I(x, y)$ . Note that only the function  $R(t)$  relates to the time  $t$  in  $m_{ij}$ .

Referring to the definition of derivation, the derivative of  $m_{ij}$  is:

$$\dot{m}_{ij} = \lim_{\Delta t \rightarrow 0} \frac{\Delta m_{ij}}{\Delta t} \quad (3.18)$$

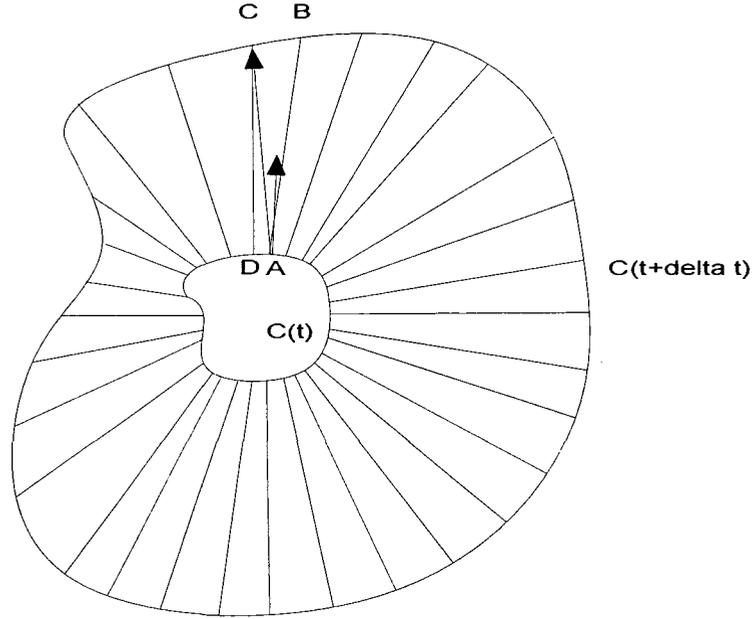
where  $\Delta m_{ij} = m_{ij}(t + \Delta t) - m_{ij}(t)$

Then, one obtains

$$\begin{aligned} \Delta m_{ij} &= \iint_{R(t+\Delta t)} F(x, y) dx dy - \iint_{R(t)} F(x, y) dx dy \\ &= \iint_{R(t)} F(x, y) dx dy \end{aligned} \quad (3.19)$$

and

$$\Delta R(t) = R(t + \Delta t) - R(t) \quad (3.20)$$



**Figure 3-4 Time variation of  $C(t)$**

Referring to Figure 3-4, assume that this target object projects contour  $C(t)$  at time  $t$  and contour  $C(t + \Delta t)$  at time  $t + \Delta t$ . Specially, point A which locates on contour  $C(t)$  at time  $t$  will move to point C on contour  $C(t + \Delta t)$  at time  $t + \Delta t$ . The region between  $C(t)$  and  $C(t + \Delta t)$  has been divided into  $N$  small regions. Assuming the coordinates of A and C are  $(x_i, y_i)^T$  and  $(x_i + \Delta x_i, y_i + \Delta y_i)^T$  respectively, where  $1 \leq i \leq N$ . Denote  $\Delta\sigma_i$  the area of each small region and  $\sigma = \max_{1 \leq i \leq N} \{\Delta\sigma_i\}$ .

According to the definition of “Definite Integration”,  $\Delta m_{ij}$  can be obtained by

$$\Delta m_{ij} = \lim_{\sigma \rightarrow 0} \sum_{i=1}^N F(x_i, y_i) \Delta\sigma_i \quad (3.21)$$

Referring to Figure 3-4, with the increase of  $N$ , point A and point D will become closer and the shape of ABCD will become a rough triangle. In this case, the area of ABCD is

$$\Delta\sigma_i = \overline{AC} * \vec{n} \cdot \quad (3.22)$$

where  $\overline{AC} = (\Delta x_i, \Delta y_i)$ ;  $\vec{n}$  is the unitary vector normal to  $C(t)$  at point A; "\*" is the inner product.

Thus,

$$\Delta m_{ij} = \lim_{\sigma \rightarrow 0} \sum_{i=1}^N F(x_i, y_i) (\Delta x_i, \Delta y_i) * \vec{n} \quad (3.23)$$

According to the definition of "Contour Integration",

$$\dot{m}_{ij} = \oint_C F(x, y) \dot{x}^T * \vec{n} dl \quad (3.24)$$

and the Green's Theorem:

$$\oint_C F * \vec{n} ds = \iint_D (\nabla * F) dA \quad (3.25)$$

We can obtain that

$$\begin{aligned} \dot{m}_{ij} &= \iint_R (\nabla * (F(x, y) \dot{x}^T)) dx dy \\ &= \iint_R [\nabla * (F(x, y) \dot{x}) - \nabla * (F(x, y) \dot{y})] dx dy \\ &= \iint_R \left( \frac{\partial(\dot{x}F(x, y))}{\partial x} + \frac{\partial(\dot{y}F(x, y))}{\partial y} \right) dx dy \\ &= \iint_R \left( \frac{\partial F(x, y)}{\partial x} \dot{x} + \frac{\partial \dot{x}}{\partial x} F(x, y) + \frac{\partial F(x, y)}{\partial y} \dot{y} + \frac{\partial \dot{y}}{\partial y} F(x, y) \right) dx dy \\ &= \iint_R \left( \frac{\partial F(x, y)}{\partial x} \dot{x} + \frac{\partial F(x, y)}{\partial y} \dot{y} + F(x, y) \left( \frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} \right) \right) dx dy \end{aligned} \quad (3.26)$$

Recall the equation (3.15). The traditional Jacobian matrix can be written as

$$\begin{cases} \dot{x} = -\frac{\lambda}{z} v_x + \frac{x}{z} v_z + \frac{xy}{\lambda} \omega_x - \frac{x^2 + \lambda^2}{\lambda} \omega_y + y \omega_z \\ \dot{y} = -\frac{\lambda}{z} v_y + \frac{y}{z} v_z + \frac{y^2 + \lambda^2}{\lambda} \omega_x - \frac{xy}{\lambda} \omega_y - x \omega_z \end{cases} \quad (3.27)$$

Thus, the following equations are obtained by

$$\begin{cases} \frac{\partial \dot{x}}{\partial x} = \frac{1}{z} v_z + \frac{y}{\lambda} \omega_x - \frac{2x}{\lambda} \omega_y \\ \frac{\partial \dot{y}}{\partial y} = \frac{1}{z} v_z + \frac{2y}{\lambda} \omega_x - \frac{x}{\lambda} \omega_y \end{cases} \quad (3.28)$$

Since the final images applied in the visual seroving are binary,  $I(x, y)$  equals

1 within the region R. So  $F(x, y) = x^i y^j$  and

$$\begin{cases} \frac{\partial F(x, y)}{\partial x} = i x^{i-1} y^j \\ \frac{\partial F(x, y)}{\partial y} = j x^i y^{j-1} \end{cases} \quad (3.29)$$

Substituting (3.27), (3.28) and (3.29) into (3.26), the Jacobian matrix of geometric image moment related to camera velocity can be obtained as:

$$\dot{m}_{ij} = J_{mij} \dot{r} \quad (3.30)$$

where

$$J_{mij} = \begin{bmatrix} m_{vx} & m_{vy} & m_{vz} & m_{ox} & m_{oy} & m_{oz} \end{bmatrix} \quad (3.31)$$

$$\left\{ \begin{array}{l} m_{vx} = -i \frac{\lambda}{Z} m_{i-1,j} \\ m_{vy} = -j \frac{\lambda}{Z} m_{i,j-1} \\ m_{vz} = \frac{i+j+2}{Z} m_{i,j} \\ m_{ox} = \frac{i+j+3}{\lambda} m_{i,j+1} + j \lambda m_{i,j-1} \\ m_{oy} = -\frac{i+j+3}{\lambda} m_{i+1,j} - i \lambda m_{i-1,j} \\ m_{oz} = i m_{i-1,j+1} - j m_{i+1,j-1} \end{array} \right.$$

By obtaining the general form of the Jacobian matrix for geometric moments, the Jacobian matrix for the first three image features can be calculated as follows.

- For the area of object  $m_{00}$ ,  $i = j = 0$ , it is,

$$J_{m00} = \begin{bmatrix} 0 & 0 & \frac{2}{Z} m_{00} & \frac{3}{\lambda} m_{01} & -\frac{3}{\lambda} m_{10} & 0 \end{bmatrix} \quad (3.32)$$

- For the coordinate of object's centroid  $x_g$  and  $y_g$ :

Denote  $G = (x_g, y_g)^T$  the centroid of object, where  $x_g = \frac{m_{10}}{m_{00}}$  and

$$y_g = \frac{m_{01}}{m_{00}}.$$

By taking the derivative of  $x_g$  and  $y_g$ ,

$$\left\{ \begin{array}{l} \dot{x}_g = \frac{1}{m_{00}} (\dot{m}_{10} - x_g \dot{m}_{00}) \\ \dot{y}_g = \frac{1}{m_{00}} (\dot{m}_{01} - y_g \dot{m}_{00}) \end{array} \right. \quad (3.33)$$

we obtain,

$$\begin{cases} J_{xg} = \frac{1}{m_{00}}(J_{m10} - x_g J_{m00}) \\ J_{yg} = \frac{1}{m_{00}}(J_{m01} - y_g J_{m00}) \end{cases} \quad (3.34)$$

That is,

$$\begin{cases} J_{xg} = \left[ -\frac{\lambda}{Z} & 0 & \frac{x_g}{Z} & \frac{4m_{11}}{\lambda m_{00}} - \frac{3x_g y_g}{\lambda} & -\frac{4m_{20}}{\lambda m_{00}} - \lambda + \frac{3x_g^2}{\lambda} & y_g \right] \\ J_{yg} = \left[ 0 & -\frac{\lambda}{Z} & \frac{y_g}{Z} & \frac{4m_{02}}{\lambda m_{00}} + \lambda - \frac{3y_g^2}{\lambda} & -\frac{4m_{11}}{\lambda m_{00}} + \frac{3x_g y_g}{\lambda} & -x_g \right] \end{cases} \quad (3.35)$$

Similarly, for the central moments  $\mu_{ij}$ , we have  $F(x, y) = (x - x_g)^i (y - y_g)^j$ .

Take the partial derivative of  $F(x, y)$  with respect to  $x$  and  $y$ , one obtains

$$\begin{cases} \frac{\partial F(x, y)}{\partial x} = i(x - x_g)^{i-1} (y - y_g)^j \\ \frac{\partial F(x, y)}{\partial y} = j(x - x_g)^i (y - y_g)^{j-1} \end{cases} \quad (3.36)$$

Referring to (3.26), we have

$$\begin{aligned} \dot{\mu}_{ij} = & \iint_{\mathbf{R}^2} (i(x - x_g)^{i-1} (y - y_g)^j (\dot{x} - \dot{x}_g) + j(x - x_g)^i (y - y_g)^{j-1} (\dot{y} - \dot{y}_g)) \\ & + (x - x_g)^i (y - y_g)^j \left( \frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y} \right) dx dy \end{aligned} \quad (3.37)$$

Considering (3.28), (3.35) and (3.37), we obtain

$$\dot{\mu}_{ij} = J_{\mu_{ij}} \dot{\mathbf{r}} \quad (3.38)$$

where

$$J_{\mu_{ij}} = \begin{bmatrix} \mu_{vx} & \mu_{vy} & \mu_{vz} & \mu_{ax} & \mu_{ay} & \mu_{az} \end{bmatrix}$$

$$\left\{ \begin{array}{l}
\mu_{vx} = 0 \\
\mu_{vy} = 0 \\
\mu_{vz} = \frac{i+j+2}{Z} \mu_{i,j} \\
\mu_{\omega x} = \frac{i+j+3}{\lambda} \mu_{i,j+1} + i \frac{x_g}{\lambda} \mu_{i-1,j+1} + (i+2j+3) \frac{y_g}{\lambda} \mu_{i,j} \\
\quad - 4i \left( \frac{m_{11}}{\lambda m_{00}} - \frac{x_g y_g}{\lambda} \right) \mu_{i-1,j} - 4j \left( \frac{m_{02}}{\lambda m_{00}} - \frac{y_g^2}{\lambda} \right) \mu_{i,j-1} \\
\mu_{\omega y} = -\frac{i+j+3}{\lambda} \mu_{i+1,j} - i \frac{y_g}{\lambda} \mu_{i+1,j-1} + (2i+j+3) \frac{x_g}{\lambda} \mu_{i,j} \\
\quad + 4i \left( \frac{m_{20}}{\lambda m_{00}} - \frac{x_g^2}{\lambda} \right) \mu_{i-1,j} + 4j \left( \frac{m_{11}}{\lambda m_{00}} - \frac{x_g y_g}{\lambda} \right) \mu_{i,j-1} \\
\mu_{\omega z} = i \mu_{i-1,j+1} - j \mu_{i+1,j-1}
\end{array} \right. \quad (3.39)$$

Since the general form of Jacobian matrix for central moments has been obtained, the exact Jacobian matrix for the rest three image features can be calculated:

- For the orientation angle  $\alpha$ :

Let  $\Delta = \mu_{20} - \mu_{02}$  and  $A = \frac{2\mu_{11}}{\Delta}$ , The definition equation in (3.5) can be

written as,

$$\alpha = \frac{1}{2} \arctan A \quad (3.40)$$

By taking the derivative of both sides, we obtain

$$\dot{\alpha} = \frac{\dot{A}}{2(1+A^2)} \quad (3.41)$$

where

$$\dot{A} = \frac{2\dot{\mu}_{11}}{\Delta} - \frac{2\mu_{11}\dot{\Delta}}{\Delta^2}$$

Thus

$$\dot{\alpha} = \frac{\Delta\dot{\mu}_{11} - \mu_{11}\dot{\Delta}}{\Delta^2 + 4\mu_{11}^2} \quad (3.42)$$

and

$$J_\alpha = \frac{1}{\Delta^2 + 4\mu_{11}^2} (\Delta J_{\mu_{11}} - \mu_{11} J_\Delta) \quad (3.43)$$

where  $J_\Delta = J_{\mu_{20}} - J_{\mu_{02}}$ :

From (3.39),  $J_{\mu_{11}}$ ,  $J_{\mu_{20}}$  and  $J_{\mu_{02}}$  can be calculated as follow:

$$J_{\mu_{11}} = \begin{bmatrix} 0 & 0 & \frac{4}{Z} & \mu_{11\alpha x} & \mu_{11\alpha y} & \mu_{20} - \mu_{02} \end{bmatrix} \quad (3.44)$$

where

$$\begin{cases} \mu_{11\alpha x} = \frac{5}{\lambda} \mu_{12} + \frac{6}{\lambda} y_g \mu_{11} + \frac{x_g}{\lambda} \mu_{02} - 4 \left( \frac{m_{11}}{\lambda m_{00}} - \frac{x_g y_g}{\lambda} \right) \mu_{01} - 4 \left( \frac{m_{02}}{\lambda m_{00}} - \frac{y_g^2}{\lambda} \right) \mu_{10} \\ \mu_{11\alpha y} = -\frac{5}{\lambda} \mu_{21} - \frac{6}{\lambda} x_g \mu_{11} - \frac{y_g}{\lambda} \mu_{20} + 4 \left( \frac{m_{20}}{\lambda m_{00}} - \frac{x_g^2}{\lambda} \right) \mu_{01} + 4 \left( \frac{m_{11}}{\lambda m_{00}} - \frac{x_g y_g}{\lambda} \right) \mu_{10} \end{cases} \quad (3.45)$$

$$J_{\mu_{20}} = \begin{bmatrix} 0 & 0 & \frac{4}{Z} & \mu_{20\alpha x} & \mu_{20\alpha y} & 2\mu_{11} \end{bmatrix} \quad (3.46)$$

where

$$\begin{cases} \mu_{20\alpha x} = \frac{5}{\lambda} \mu_{21} + \frac{5}{\lambda} y_g \mu_{20} + \frac{2x_g}{\lambda} \mu_{11} - 8 \left( \frac{m_{11}}{\lambda m_{00}} - \frac{x_g y_g}{\lambda} \right) \mu_{10} \\ \mu_{20\alpha y} = -\frac{5}{\lambda} \mu_{30} - \frac{7}{\lambda} x_g \mu_{20} + 8 \left( \frac{m_{20}}{\lambda m_{00}} - \frac{x_g^2}{\lambda} \right) \mu_{10} \end{cases} \quad (3.47)$$

$$J_{\mu_{02}} = \begin{bmatrix} 0 & 0 & \frac{4}{Z} & \mu_{02\alpha x} & \mu_{02\alpha y} & -2\mu_{11} \end{bmatrix} \quad (3.48)$$

where

$$\begin{cases} \mu_{02\alpha x} = -\frac{5}{\lambda} \mu_{12} - \frac{5}{\lambda} x_g \mu_{02} - \frac{2y_g}{\lambda} \mu_{11} + 8 \left( \frac{m_{11}}{\lambda m_{00}} - \frac{x_g y_g}{\lambda} \right) \mu_{01} \\ \mu_{02\alpha y} = \frac{5}{\lambda} \mu_{03} + \frac{7}{\lambda} x_g \mu_{02} - 8 \left( \frac{m_{02}}{\lambda m_{00}} - \frac{y_g^2}{\lambda} \right) \mu_{01} \end{cases} \quad (3.49)$$

- For the  $s_x$  and  $s_y$ :

Taking the derivative of (3.6), we obtain,

$$\begin{cases} \dot{s}_x = -(\dot{c}_1 c_2 + c_1 \dot{c}_2 + \dot{s}_1 s_2 + s_1 \dot{s}_2) / M_1^{\frac{9}{4}} + \frac{9}{4} (c_1 c_2 + s_1 s_2) \dot{M}_1 / M_1^{\frac{13}{4}} \\ \dot{s}_y = (\dot{s}_1 c_2 + s_1 \dot{c}_2 - \dot{c}_1 s_2 - c_1 \dot{s}_2) / M_1^{\frac{9}{4}} - \frac{9}{4} (s_1 c_2 + c_1 s_2) \dot{M}_1 / M_1^{\frac{13}{4}} \end{cases} \quad (3.50)$$

where

$$\begin{aligned} \dot{s}_1 &= 2\dot{\mu}_{11} \\ \dot{s}_2 &= \dot{\mu}_{30} - 3\dot{\mu}_{12} \\ \dot{c}_1 &= \dot{\mu}_{20} - \dot{\mu}_{02} \\ \dot{c}_2 &= \dot{\mu}_{03} - 3\dot{\mu}_{21} \end{aligned}$$

Thus, the Jacobian matrices for  $s_x$  and  $s_y$  are,

$$\begin{cases} J_{s_x} = -(c_2 J_{c1} + c_1 J_{c2} + s_2 J_{s1} + s_1 J_{s2}) / M_1^{\frac{9}{4}} + \frac{9}{4} (c_1 c_2 + s_1 s_2) J_{M1} / M_1^{\frac{13}{4}} \\ J_{s_y} = (c_2 J_{s1} + s_1 J_{c2} - s_2 J_{c1} - c_1 J_{s2}) / M_1^{\frac{9}{4}} - \frac{9}{4} (s_1 c_2 + c_1 s_2) J_{M1} / M_1^{\frac{13}{4}} \end{cases} \quad (3.51)$$

where

$$\begin{aligned} J_{s1} &= 2J_{\mu11} \\ J_{s2} &= J_{\mu30} - 3J_{\mu12} \\ J_{c1} &= J_{\mu20} - J_{\mu02} \\ J_{c2} &= J_{\mu03} - 3J_{\mu21} \end{aligned}$$

and the Jacobian matrix for central moment  $J_{\mu}$  can be obtained from (3.39).

To summarize, the relationship between the motion of image moments and the velocity of camera has been obtained.

### 3.3 Depth estimation

In the Jacobian matrix, the parameters are moments and the depth  $Z$ . The moments can be calculated from the image directly. However, the depth  $Z$  cannot be measured directly. In this case, the depth  $Z$  should be estimated. In this thesis, a simple method based on homogenous transformation among different frames has been

proposed.

Totally, there are four frames in the whole system: world frame (robot-base frame), camera frame (end-effector frame), object frame and the image frame. In the proposed method, the homogenous transformation between the world frame and the camera frame has been utilized. The details are introduced as follows.

Figure 3-5 shows the simple structure of visual servoing system. The dash line is the optical axis of camera, which intersects the X-Y plane of world frame at point P. Denote  $h$  as the height of camera origin O with respect to the world frame and  $Z$  is the distance from O to the point P measures along camera's optical axis. According to the definition of the depth,  $Z$  also represents the depth of the point P. Draw an imaginary perpendicular line from O to the XY-plan of the world frame. Denote the intersection point as A. Angle  $\beta$  represents the angle between OA and OP.

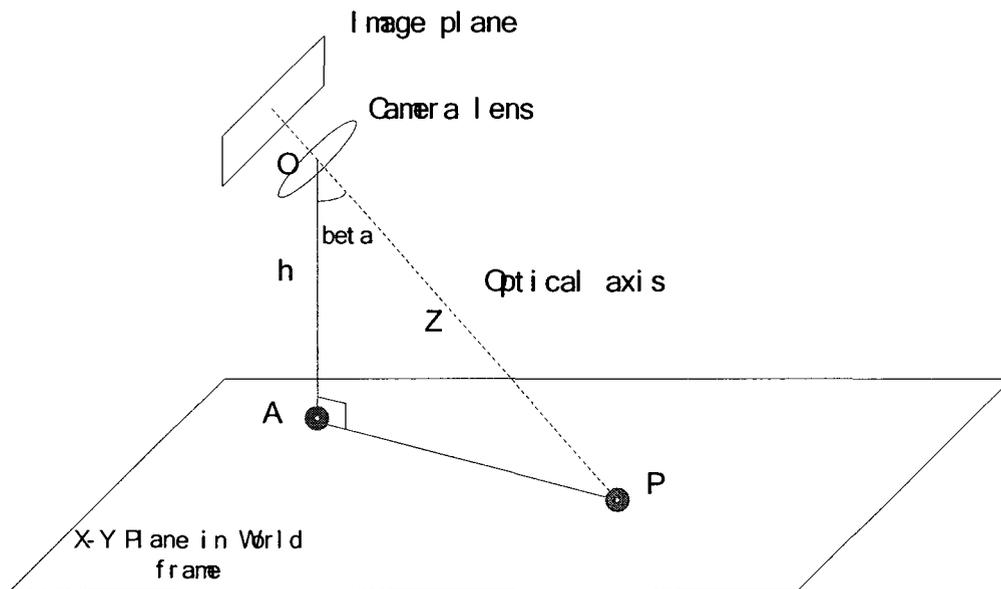


Figure 3-5 Depth estimation

In visual servoing, the robot's forward kinematics  ${}^cH$  is assumed be known.

In this case, the coordinate of O with respect to the world frame can be obtained.

Assume that the coordinate is  ${}^wO = [x_w \quad y_w \quad z_w]^T$  and note that  $h = z_w$ . Then, the coordinate of A with respect to the world frame is  ${}^wA = [x_w \quad y_w \quad 0]^T$ .

Since  ${}^cH$  is known, the inverse kinematics matrix  ${}^cH^{-1}$  is available. The coordinate of A with respect to the camera frame  ${}^cA = [x_c \quad y_c \quad z_c]^T$  can be obtained by  ${}^cA = {}^cH {}^wA = ({}^cH)^{-1} {}^wA$ .

Thus, according to the definition of cosine, in the camera frame, one has

$$\cos \beta = \frac{z_c}{\sqrt{x_c^2 + y_c^2 + z_c^2}} \quad (3.52)$$

Also, in the world frame, one has

$$\cos \beta = \frac{h}{Z} \quad (3.53)$$

Combining (3.52) and (3.53), one obtains

$$Z = \frac{h}{\cos \beta} = \frac{h \sqrt{x_c^2 + y_c^2 + z_c^2}}{z_c} \quad (3.54)$$

In this case, the depth of point P can be obtained. Based on this result, if the optical axis can be moved to the object centroid and then kept around there, the depth of the object centroid can be estimated successfully. Furthermore, if the object is small and thin enough, the depth of centroid can represent the depth of every point on the object. Therefore, the depth of the whole object is estimated.

### 3.4 Summary

In this chapter, six image features derived from image geometric and central

moments have been chosen. Also, the Jacobian matrices which indicate the relationship between the motion of selected image features and the velocity of camera have been derived. Furthermore, a simple method for the depth estimation has been proposed. The depth of object can be calculated simply, quickly and correctly. Based on these results, the visual servoing feedback control signal can be generated.

# CHAPTER 4 IBVS WITH IMAGE MOMENTS

## 4.1 Introduction

As mentioned in the previous chapter, the improved image features have been chosen to create the visual servoing control signal. In this chapter, a sequence control algorithm of image based visual servoing using the image moments as the image features is proposed to improve the robustness of visual servoing system.

## 4.2 Proposed sequence control algorithm

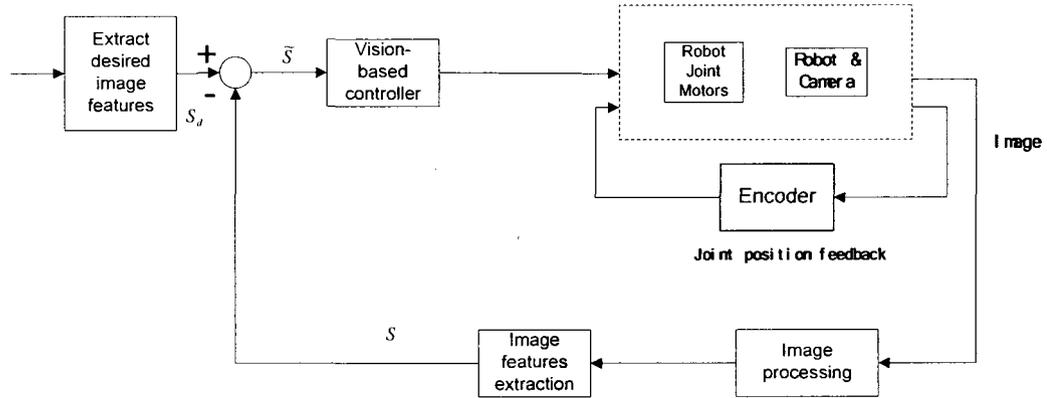
In this work, the indirect IBVS has been adopted. It means that we just have to design the visual servoing controller other than the joint position controller. The block diagram of IBVS system is presented in Figure 4-1. The general form of visual servoing controller is

$$u = -KJ_{image}^{-1}\tilde{s} \quad (4.1)$$

where  $u$  is feedback controller,  $K$  is proportional gain and  $\tilde{s}$  the error of image features.

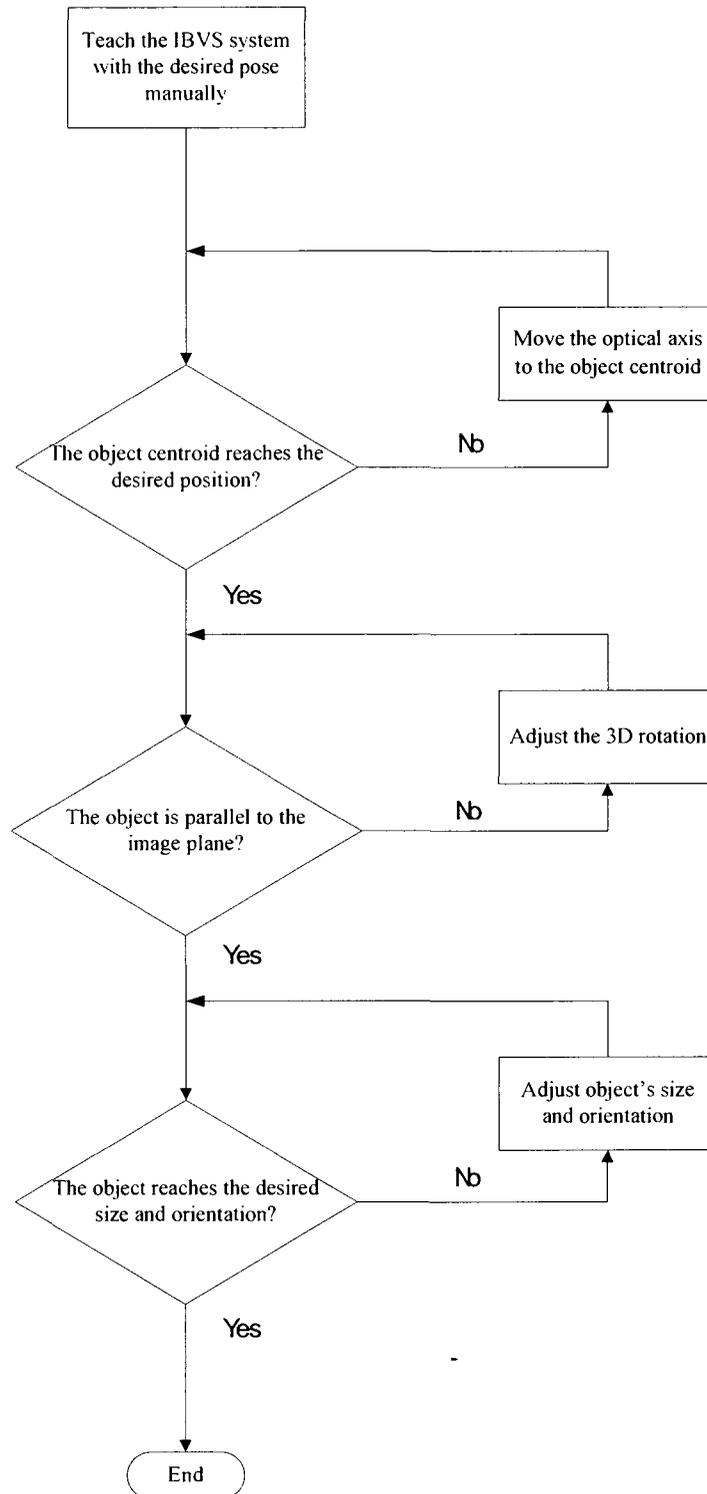
In traditional IBVS, the image Jacobian matrix might become singular because of the pure 180 degrees rotation around the optical axis [48]. In such case, the corner points want to go to the desired position by straight lines, which will cause the camera move backward forever. In the proposed algorithm, the image features will be

extracted from the whole object directly. In this case, the singularity problem of image Jacobian matrix could be avoided.



**Figure 4-1 Block diagram of proposed IBVS**

Before describing the details of the algorithm, the flow chart of the proposed IBVS algorithm is given in Figure 4-2. There are four stages of the proposed control algorithm. In each stage, different image features will be chosen and the error  $\tilde{s}$  and corresponding  $J_{image}$  will become different accordingly.



**Figure 4-2 Flow chart of the proposed IBVS algorithm**

Following are the details of the proposed IBVS algorithm.

## Stage 1: Teaching stage

Before the visual servoing process starts, we have to know the desired values of the image features. In this stage, as shown in Figure 4-1, the robot will be taught with the desired position of object manually in “Extract desired image features” block. Since the image is in the desired position, all the desired image features  $s_d$  can be calculated and recorded as the reference for the calculation of the error  $\tilde{s}$ . After this stage, the object could be located in any place in the FOV and then moved to the desired place by the end of visual servoing process. For the same object and the same desired pose, this stage is only needed to be performed once.

## Stage 2: Move the optical axis to the object centroid (drive the object centroid to the center of FOV)

In the depth estimation method mentioned above, the estimated depth of object centroid can be considered as the depth of whole object if the intersection point P can be fixed around the object centroid. Therefore, in order to obtain the depth for estimating the Jacobian matrix, the optical axis should be moved to the object centroid first.

First of all, the object centroid has to be found out and labeled as  $G = (x_g, y_g)^T$ . Since three non-collinear points could define a plane, other two reference points which are imaginary will be introduced. Those two reference points are chosen as  $G_1 = (x_g + con_1, y_g + con_1)^T$  and  $G_2 = (x_g + con_2, y_g - con_2)^T$ , where  $con_1$  and  $con_2$  are constants. Consider  $G$ ,  $G_1$  and  $G_2$  as the desired location and the desired

image features is  $s_{d2} = [x_g \quad y_g \quad x_g + con_1 \quad y_g + con_1 \quad x_g + con_2 \quad y_g - con_2]^T$  in this stage. In the image, the optical point is the image center. Hence, choose the image center  $C = (0,0)^T$  as the current location. Accordingly, choose two points,  $C_1 = (con_1, con_1)^T$  and  $C_2 = (con_2, -con_2)^T$ , near the image center as the reference points. The current image feature in this stage is  $s_{c2} = [0 \quad 0 \quad con_1 \quad con_1 \quad con_2 \quad -con_2]^T$ . Therefore, the error of image feature at this stage is  $\tilde{s}_2 = s_{d2} - s_{c2}$ .

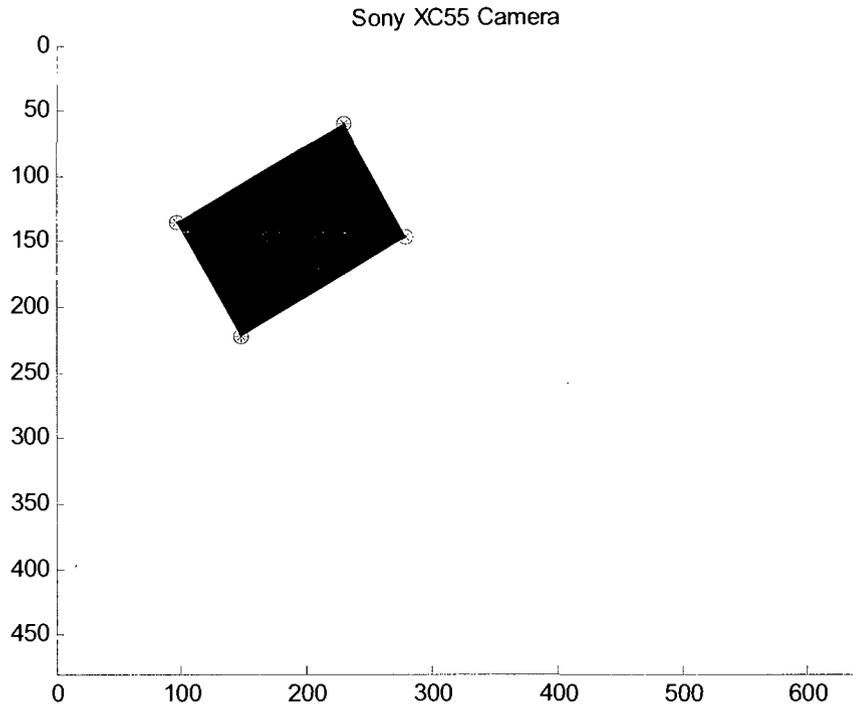
The control law at this stage is

$$u_2 = -K_2 J_2^{-1} \tilde{s}_2 \quad (4.2)$$

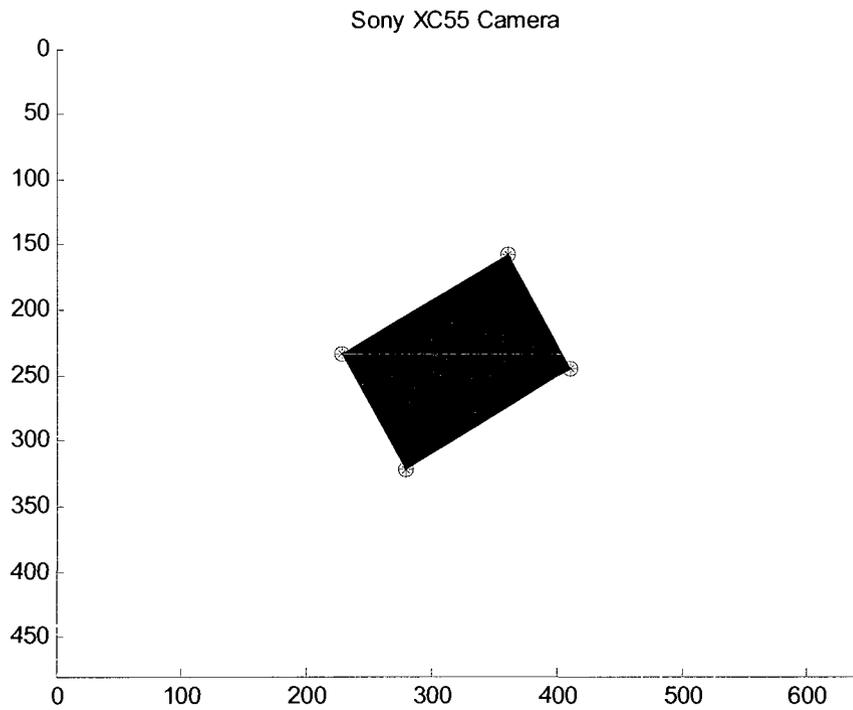
where the Jacobian matrix  $J_2$  is calculated on-line by (3.15).

At this stage, the current location is image center  $C = (0,0)^T$ , and the depth information needed here is the depth of  $C$ . According to the depth estimation method described above, the depth  $Z_c$  can be calculated by the method described before. Besides, since the other two reference points are very close to  $C$ , the depth  $Z_c$  can also be treated as  $Z_{c1}$  and  $Z_{c2}$ .

Figure 4-2 and Figure 4-3 show the initial image and the final image at this stage respectively. It is shown that the object has been moved to the center of FOV directly without changing any pose or size.



**Figure 4-3 Initial position of Stage 2**



**Figure 4-4 Desired position of Stage 2**

In order to simplify and avoid adding unnecessary camera motion in Cartesian

space, only the translational movement is used at this stage. Equation (3.15) can be decomposed into two parts, translational part and rotational part [12].

$$\begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} J_{image}^t & J_{image}^r \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ v_z \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \quad (4.3)$$

where

$$J_{image}^t = \begin{bmatrix} -\frac{\lambda}{Z} & 0 & \frac{x}{Z} \\ 0 & -\frac{\lambda}{Z} & \frac{y}{Z} \end{bmatrix} \quad (4.4)$$

$$J_{image}^r = \begin{bmatrix} \frac{xy}{\lambda} & -\frac{\lambda^2 + x^2}{\lambda} & y \\ \frac{\lambda^2 + x^2}{\lambda} & -\frac{xy}{\lambda} & -x \end{bmatrix} \quad (4.5)$$

Since the elements of the Jacobian matrix cannot be set to zero directly, a proper  $K_2$  has been chosen so that,

$$K_2 J_2^{-1} = (J_2 K_2^{-1})^{-1} = k_2 \begin{bmatrix} J_2^t & 0 \end{bmatrix}^+ \quad (4.6)$$

where “+” means the pseudo-inverses.

Thus, the controller has become:

$$u_2 = -k_2 \begin{bmatrix} J_2^t & 0 \end{bmatrix}^+ \tilde{s}_2 \quad (4.7)$$

The switch law is described as: if the error norm of  $\tilde{s}_2$  falls below a threshold, the sequence controller will transfer from Stage 2 to Stage 3.

The threshold is given as:

$$|s_{d2} - s_{c2}| \leq s_{thre2} \quad (4.8)$$

where  $s_{thre2}$  is the predetermined threshold value.

Since the threshold will affect the depth accuracy which relates to the whole system accuracy. The threshold should be chosen small enough.

As soon as the error norm in Stage 2 falls below  $s_{thre2}$ , based on the hypothesis mentioned before, the depth of optical point could be used as the depth of object.

It should be mentioned that driving the object centroid to the center of FOV can also solve the problem that partial object is out of the FOV initially. In this case, at first the centroid of partial object inside the FOV is considered as the temporary desired position  $s_{d2}$ . When the camera moves towards the object, the whole object will finally appear in the FOV.

### **Stage 3: Adjust the rotation corresponding to $\omega_x$ and $\omega_y$**

By testing the image features chosen in Chapter 3, we found that if all the six image features mentioned in Chapter 3 are used at the same time to calculate the Jacobian matrix, the Jacobian matrix will not become triangular (the matrix is too complex to show here). That means some motion will couple with the others. In this case, the two image features  $s_x$  and  $s_y$ , which relate to the rotation  $\omega_x$  and  $\omega_y$  are adjusted first in order to decouple the motions of 2D translation and 3D rotation, so that the Jacobian matrix becomes triangular and control algorithm is simplified.

In this research, we assume that in the desired position, the image plane should be parallel to the surface of object. Sometimes, the target object is not parallel but has some rotation angles with respect to the work space plane (XY-plan of world frame). In this stage, the 3D rotation angle will be adjusted by camera's rotational motion.

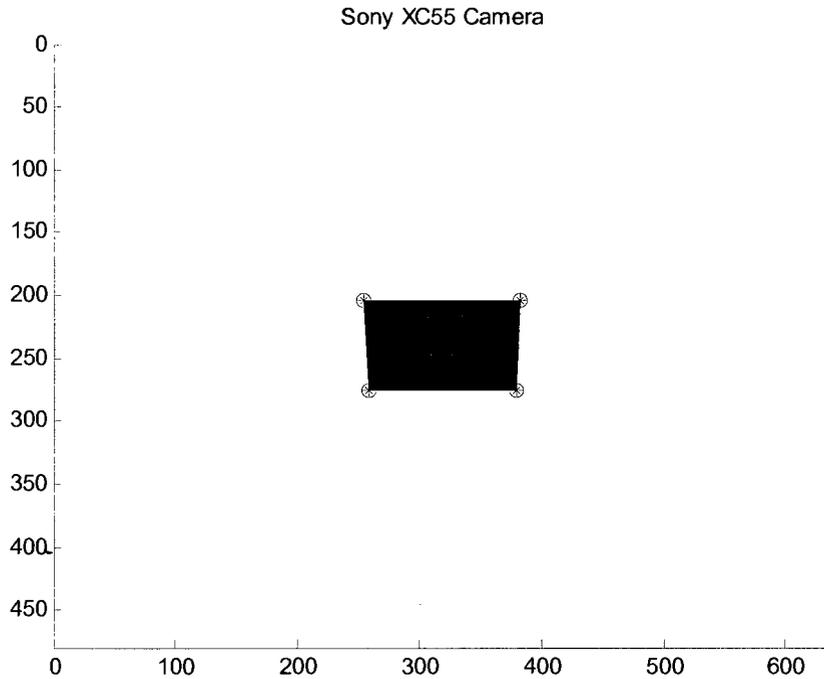
While controlling the rotation, in order to obtain the correct depth information, the object centroid will also be controlled in the center of FOV by camera's 2D translational motion.

When the image plane is parallel to the object surface, the image features will be  $s_{yxd} = (s_{xd}, s_{yd})^T$ . Assume that in the current position, the values of  $s_x$  and  $s_y$  are  $s_{xyc} = (s_{xc}, s_{yc})^T$ , the error will be  $(\tilde{s}_x, \tilde{s}_y)^T = (s_{xd}, s_{yd})^T - (s_{xc}, s_{yc})^T$ .

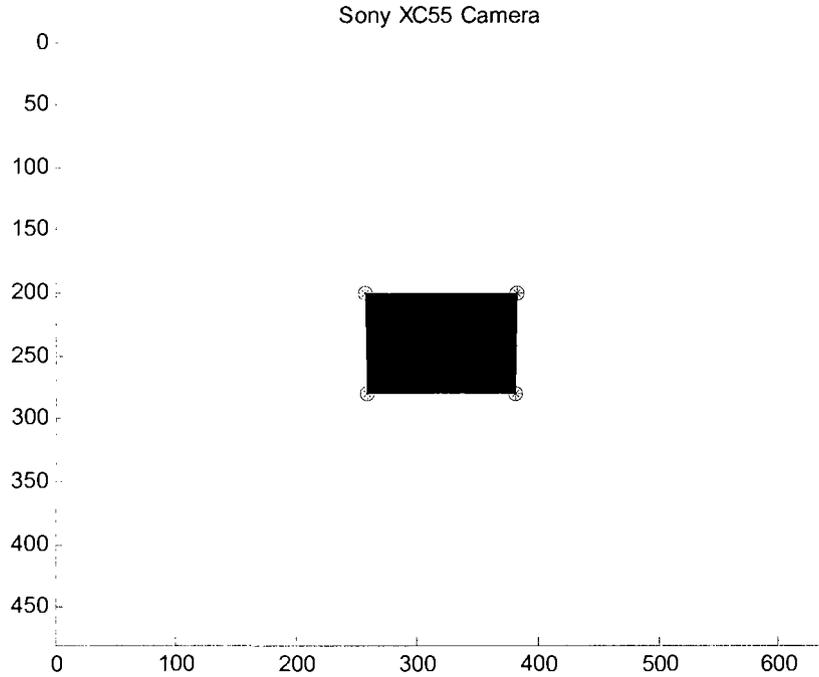
The control law is

$$u_3 = -K_3 J_3^{-1} \tilde{s}_3 \quad (4.9)$$

where  $\tilde{s}_3 = (\tilde{x}_g, \tilde{y}_g, \tilde{s}_x, \tilde{s}_y)^T$  and the Jacobian matrix  $J_3$  is calculated by (3.35) and (3.51).



**Figure 4-5 Initial position of Stage 3**



**Figure 4-6 Desired position of Stage 3**

In Figure 4-4, it is assumed that the initial object has a rotational angle round the x axis in the object frame. Therefore, in the FOV, the object is no longer a rectangle but a trapezium. After adjustment, in the final image of Stage 3, the object is almost resumed to a rectangle in Figure 4-5.

Because the pose and the centroid should be controlled at the same time, the time for execution might take long. In this stage, the Jacobian matrix is not calculated on-line but set as constant one which is calculated from the desired position.

The switch law is described as: if the error norm of  $\tilde{s}_3$  falls below a threshold, the sequence controller will transfer from Stage 3 to Stage 4.

The threshold is given as:

$$|s_{d3} - s_{c3}| \leq s_{thre3} \quad (4.10)$$

where  $s_{thre3}$  is the predetermined threshold value.

#### Stage 4: Adjust object's size and orientation

At this stage, the control goal is to adjust the size and orientation of object. Because the image features  $s_x$  and  $s_y$  are derived from invariant moments, they will stay as the same under the size change, orientation change or translational motions. Thus, they will not be changed during this stage.

The control law is

$$u_4 = -K_4 J_4^{-1} \tilde{s}_4 \quad (4.11)$$

where  $\tilde{s}_4 = (\tilde{x}_g, \tilde{y}_g, \tilde{\alpha}, \tilde{\alpha})^T$  and the Jacobian matrix  $J_4$  is calculated on-line by (3.32), (3.35) and (3.43).

As shown in Figure 4-6, in the initial image, the object is small and the current orientation is 45 degrees while the desired one is 90 degrees. Referring to Figure 4-7, by the end of Stage 4, the camera has zoomed in and rotated around the optical axis. The final image features have reached the desired values.

In order to increase the accuracy, the Jacobian matrix at this stage is estimated on-line.

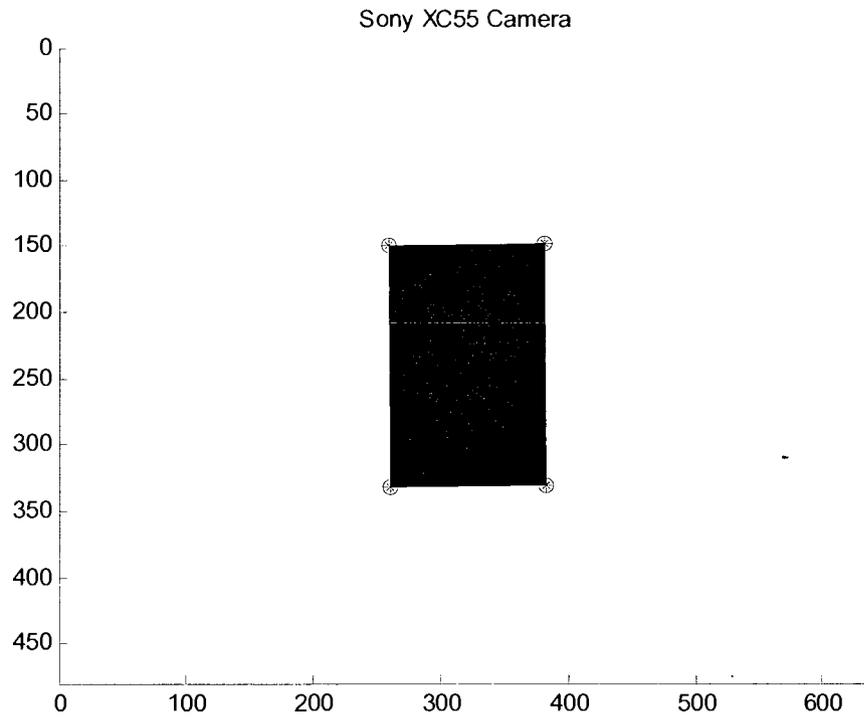
If the error norm falls below a predetermined threshold,

$$|s_{d4} - s_{c4}| \leq s_{thre4} \quad - \quad (4.12)$$

where the  $s_{thre4}$  is the predetermined threshold value, it means all the image features have reached the desired value. The control goal has been achieved successfully and the whole system will be stopped.



**Figure 4-7 Initial position of Stage 4**



**Figure 4-8 Desired position of Stage 4**

### 4.3 Summary

In this chapter, a sequence control algorithm has been proposed. The image features which are extracted from image moments have been introduced in the algorithm. The whole process includes four stages: teaching, move the optical axis to the object centroid, adjust the rotation corresponding to  $\omega_x$  and  $\omega_y$  and adjust object's size and orientation.

The basic structure and idea of the visual servoing controller are derived from the traditional IBVS, so the proposed algorithm inherits the advantages such as model free and robust to the calibration of camera. In addition, the singularity of image Jacobian matrix has been avoided. Besides, the problem that partial object locates out of FOV has been solved.

# CHAPTER 5 SIMULATION RESULTES OF IBVS WITH IMAGE MOMOENTS

## 5.1 Introduction

In this chapter, the proposed IBVS using image moments as the image features will be simulated. The simulation results on tracking the object with several shapes will be presented. The tracking errors will be analyzed and the robustness of the proposed control algorithm will be tested as well.

## 5.2 Simulation results of IBVS with image moments

The proposed control algorithm has been simulated with Matlab/Simulink and Robotics Toolbox [30]. The robot model is adopted as Motoman UPJ [31] and the camera model is Sony XC55 [32]. The camera reference frame is set to be coincident with the robot end effector frame.

At first, the simulation will be accomplished with good camera calibration. Three shapes will be tested: rectangle, star shape and date core shape. After that, in the robustness test, the camera calibration error will be added into camera system.

### 5.2.1 Results with ideal condition

- Rectangular object

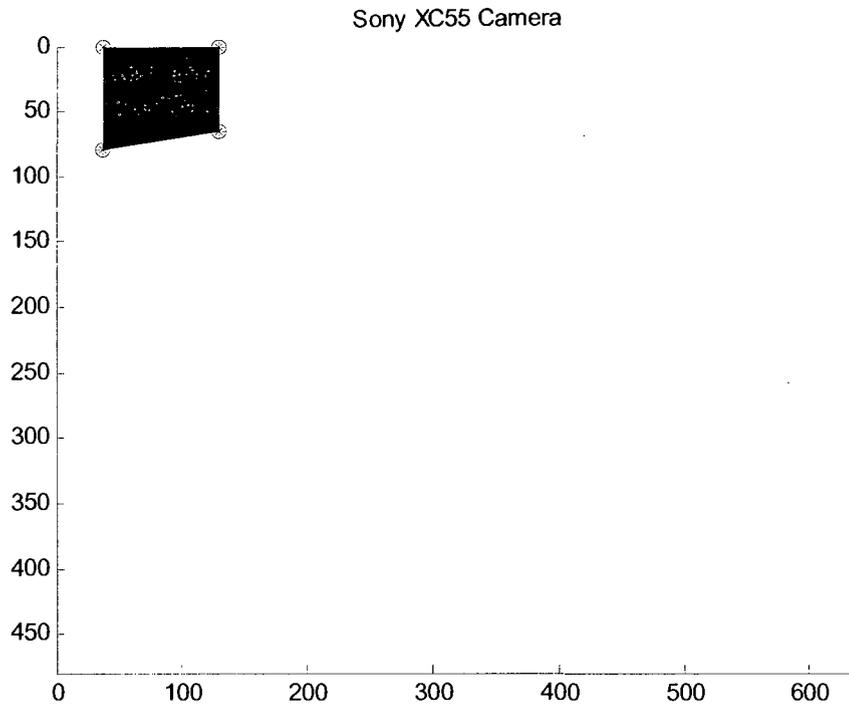
Two tasks with different initial poses for the rectangular object have been carried out. The desired and initial image features are shown as follows,

**Table 5-1 Desired value and initial value for task 1**

Image features	Desired value	Initial value in task 1	Initial Errors of task 1	Initial value in task 2	Initial Errors of task 2
$x_g$	0	-278.4	278.4	-256.2	256.2
$y_g$	0	-199.6	199.6	-181.2	181.2
$a$ (number of pixels)	22800	5250	17550	7515	15285
$\alpha$ (degrees)	45	0	45	-20	65
$s_x$	0.1	0.1187	-0.0187	0.2335	-0.1335
$s_y$	0	0.1655	-0.1655	0.3849	-0.3849

➤ Task 1

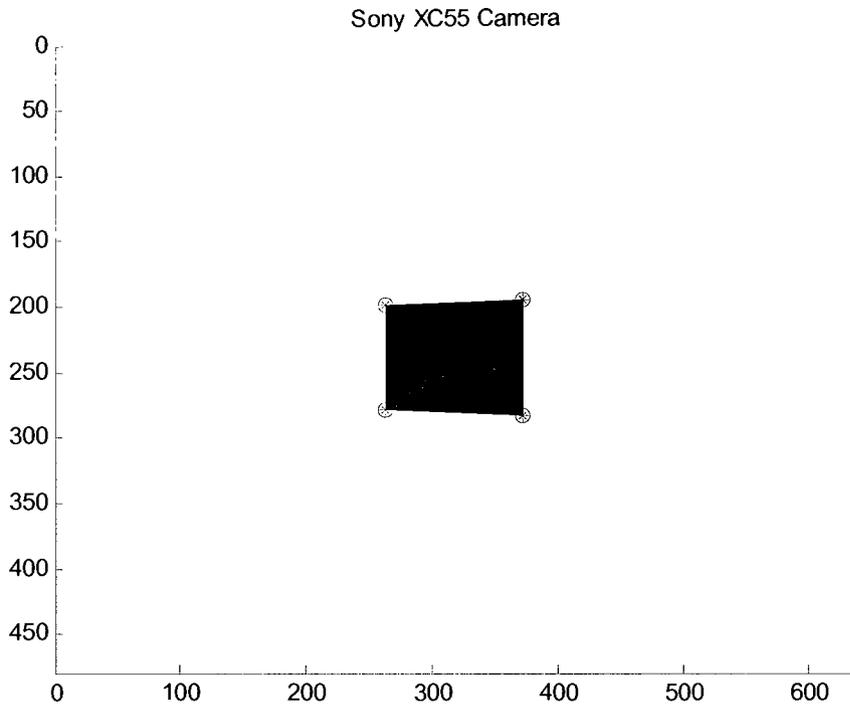
In the initial position, the object is located in a corner of FOV. It does not have 2D rotation but rotates around y axis (with respect to object frame) by -30 degrees. Figure 5-1 shows the initial image at the beginning of visual servoing process. It can be seen that the target object is in the left-top of FOV and is partially outside the FOV.



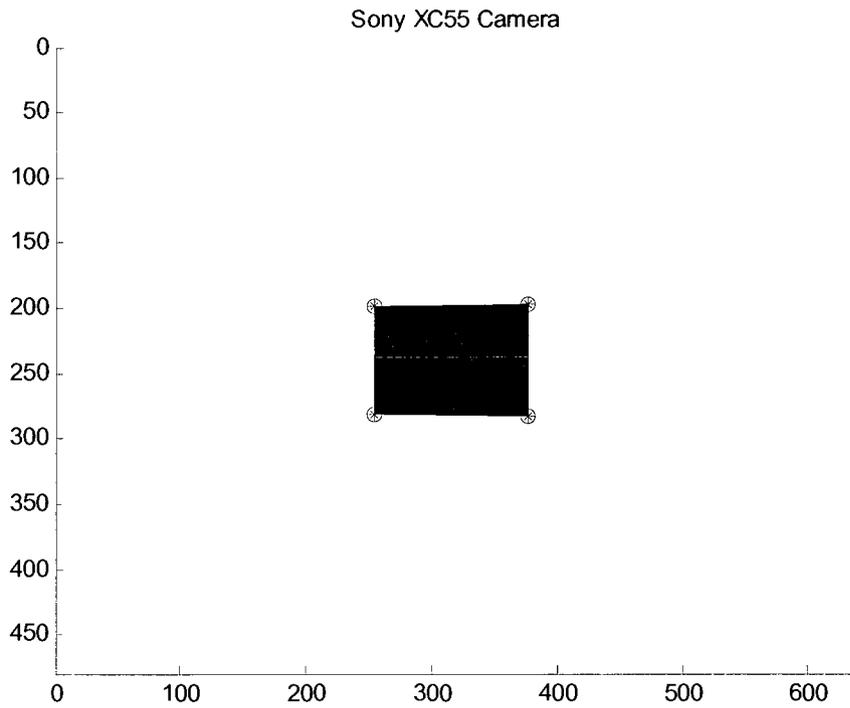
**Figure 5-1 Initial image of task 1**

Referring to the algorithm described in Chapter 4, by the end of Stage 2, the object has been moved to the center of the FOV shown in Figure 5-2. Note that in all the tests in this thesis, the constants  $con_1$  and  $con_2$  are chosen as 5 and 10 pixels respectively.

In the Stage 3, the image features  $s_x$  and  $s_y$  will be adjusted. Comparing Figure 5-2 and Figure 5-3, the difference is shown clearly. In the former image, the object is nearly a trapezium while in the later one, it is almost a rectangle. It means that the camera has been adjusted to be parallel to the object.

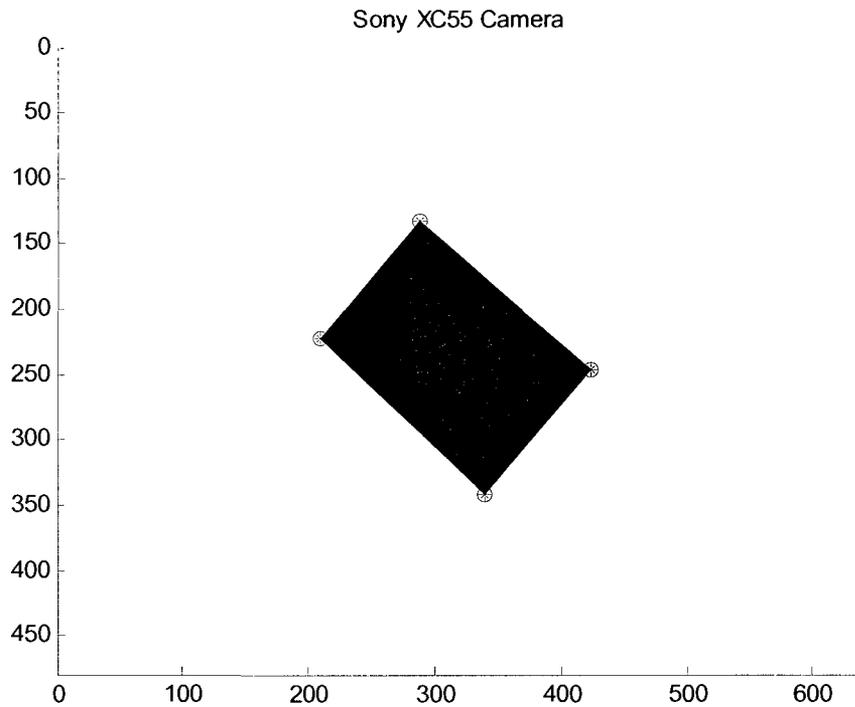


**Figure 5-2 Image of task 1 after Stage 2**



**Figure 5-3 Image of task 1 after 3**

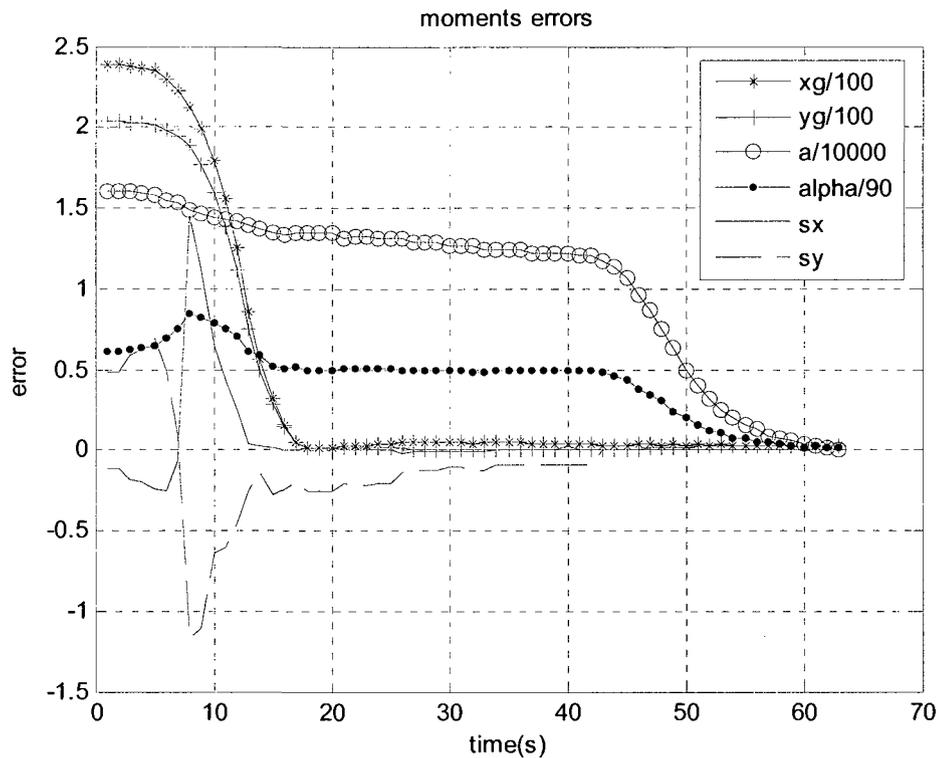
In Stage 4, the control focuses on the size and orientation. Figure 5-4 shows the final image of the task 1.



**Figure 5-4 Final image of task 1**

Figure 5-5 shows the errors of the task 1. The three stages are shown distinctly. At first, the line with plus sign and the line with stars which represent the errors of object's centroid  $x_g$  and  $y_g$  respectively decrease quickly. Meanwhile, after the whole object comes into the FOV, the area and orientation almost stay as the same. Since the object is not parallel to the image plane, the image features  $s_x$  and  $s_y$  will change during Stage 2. Because the only thing considered in Stage 2 is the error of centroid, the change of  $s_x$  and  $s_y$  will not affect the system performance. After the error of centroid is reduced under the threshold and the feedback control is switched to Stage 3, the image features,  $s_x$  and  $s_y$ , will be adjusted. For the object only rotates around y axis. Only the image feature  $s_y$  which represents the post

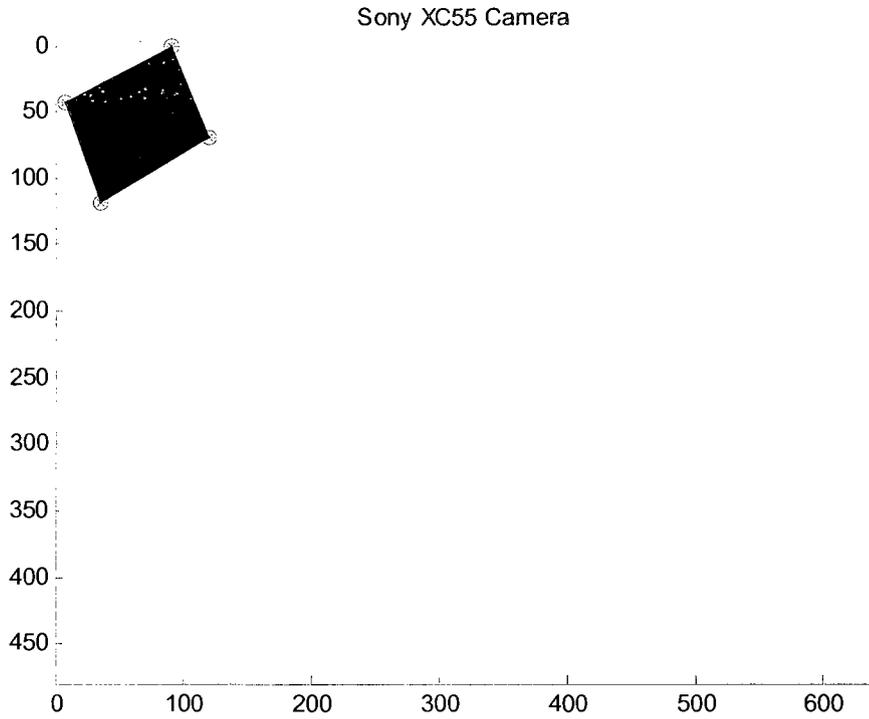
around y axis is needed to be adjusted. Referring to Figure 5-5,  $s_y$  (dash line) approaches to zero slowly and  $s_x$  (solid line) is always zero. In the last stage, the size and orientation has been rectified. It is noticed that the centroid stays as the same for Stage 3 and Stage 4 and  $s_x$ ,  $s_y$  are not changed in Stage 4.



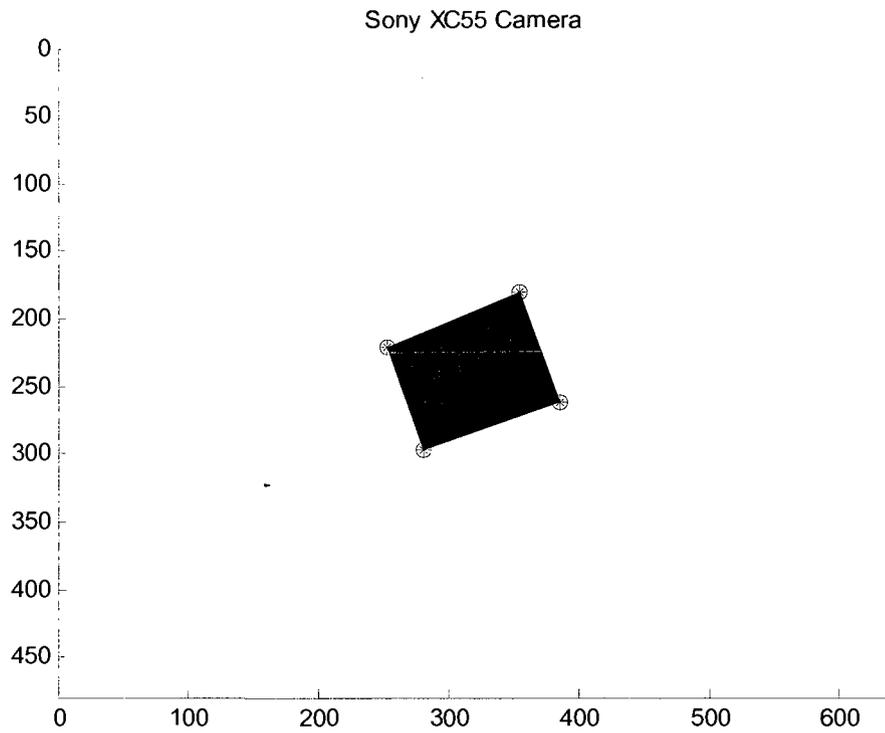
**Figure 5-5 Errors of image features of task 1**

➤ Task 2

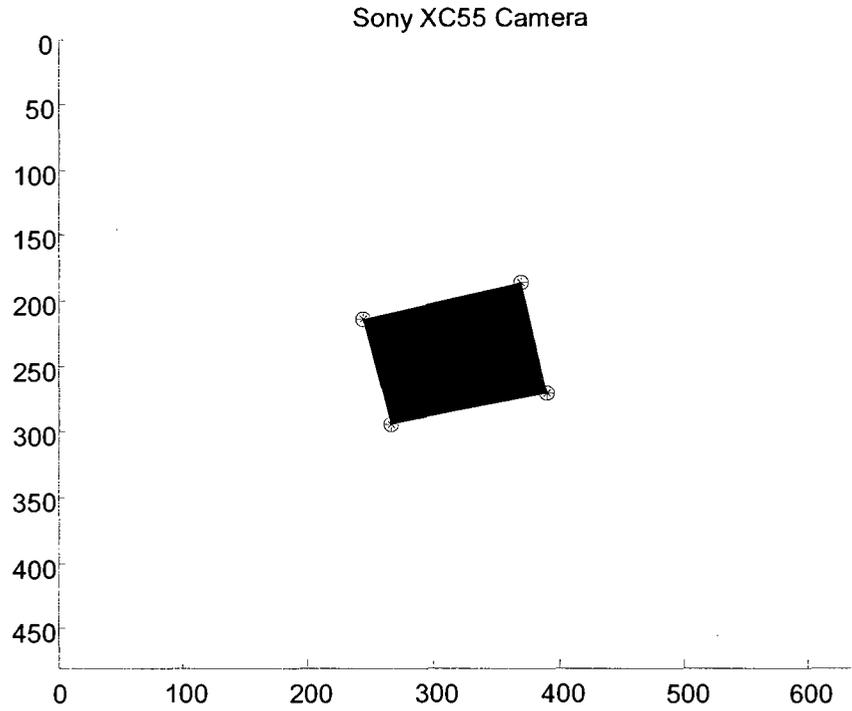
- In the initial position, the object is located in a corner of FOV too. The difference from task 1 is that the object rotates around both y axis and z axis.



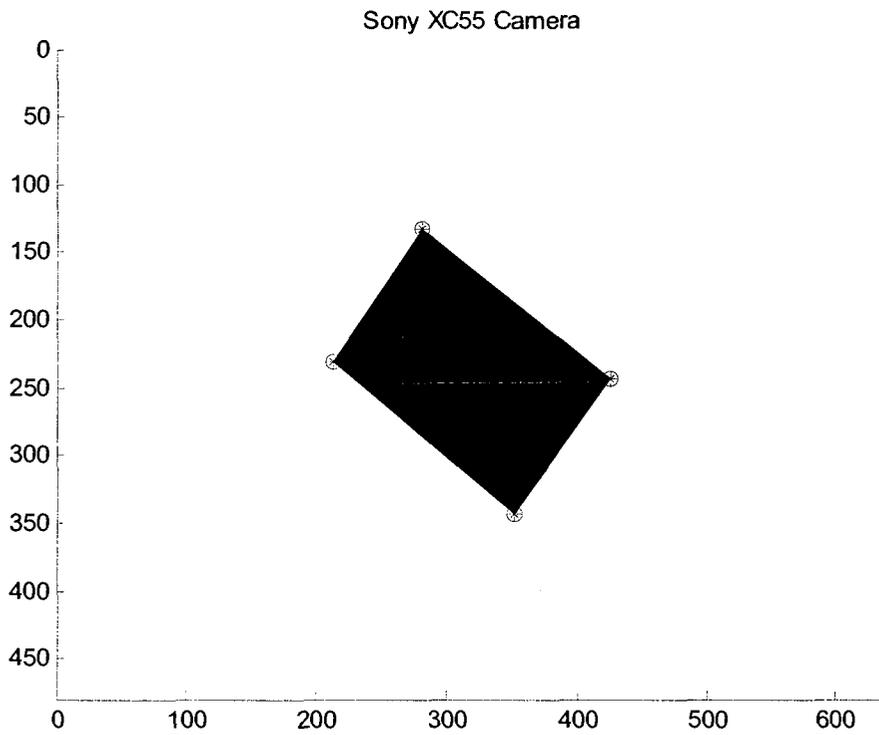
**Figure 5-6 Initial image of task 2**



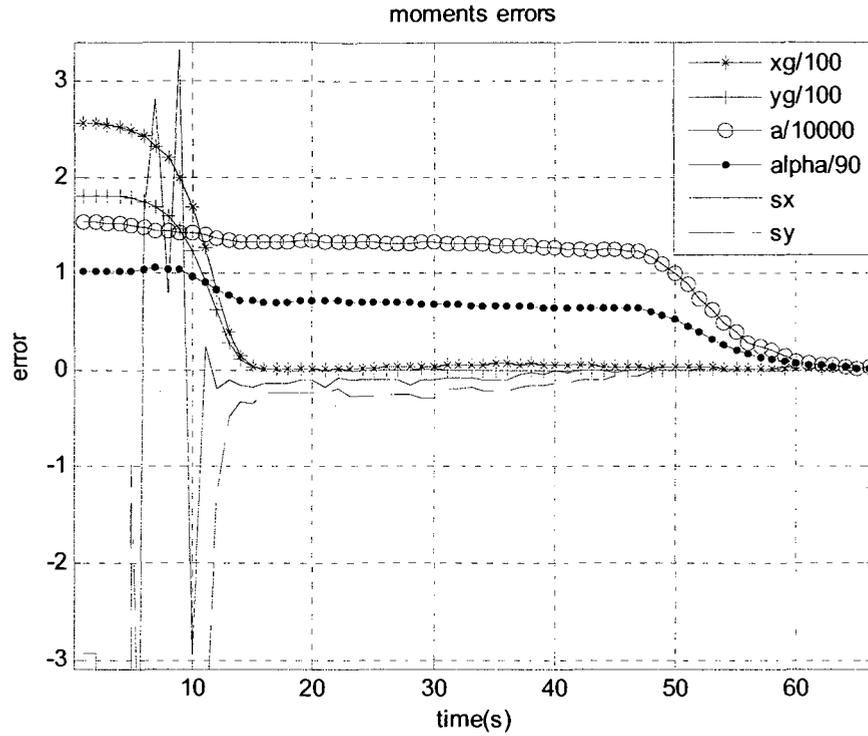
**Figure 5-7 Image of task 2 after Stage 2**



**Figure 5-8 Image of task 2 after Stage 3**



**Figure 5-9 Final image of task 2**



**Figure 5-10 Errors of image features of task 2**

Comparing Figure 5-7 and Figure 5-8, one may notice that the controller in Stage 3 has changed the trapezium into a rectangle without changing the orientation. Referring to Figure 5-10, the image features  $s_x$  and  $s_y$  are both needed to be adjusted, since the object rotates around z axis and y axis in Euler angle style.

- Star shape object (task 3)

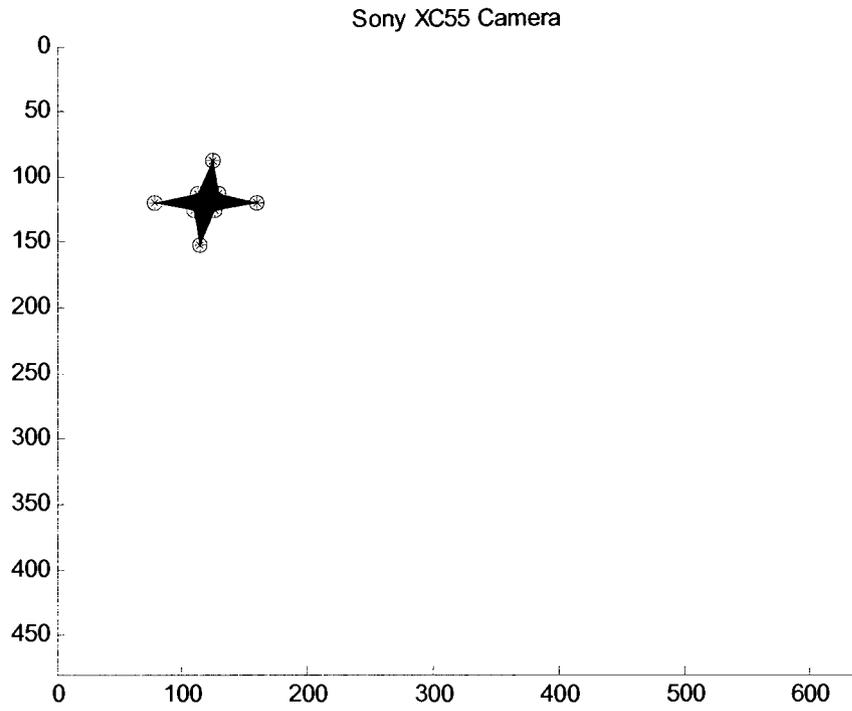
In order to demonstrate that this algorithm can also be applied to the shapes other than rectangle, two more shapes are chosen. The first one is the star shape and the other one is the date core shape. Following are the initial and desired values of image features.

**Table 5-2 The desired and initial image feature values for task 3**

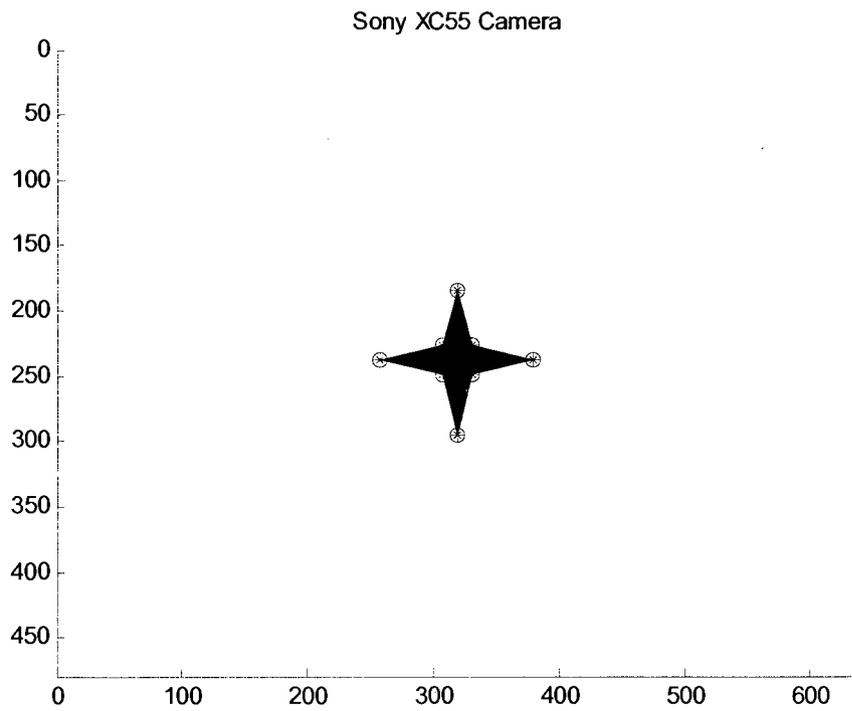
Image features	Desired value	Initial value in task 3
$x_g$	0	-202.1
$y_g$	0	-120.9
$a$ (number of pixels)	3000	1250
$\alpha$ (degrees)	0	0
$s_x$	0.1	0.09186
$s_y$	0	0.001379

In task 3, the object surface is no long the rectangle but a star. It rotates around x axis by -30 degrees and the centroid, size and orientation are also required to be adjusted.

Figure 5-11 shows the initial image of task 3 and Figure 5-12 is the final image. In Figure 5-13, the errors of all the six image features are displayed. It is demonstrated that all the errors have been controlled to converge zero gradually. It shows that even though the object is no longer a rectangle, the proposed algorithm still works effectively.



**Figure 5-11 Initial image of task 3**



**Figure 5-12 Final image of task 3**

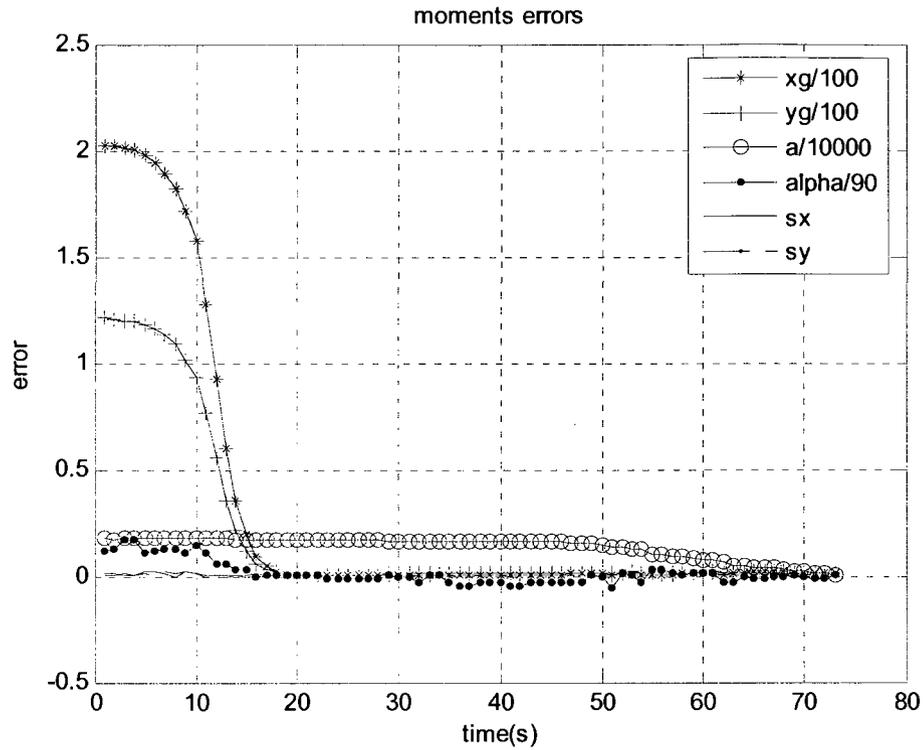


Figure 5-13 Errors of image features of task 3

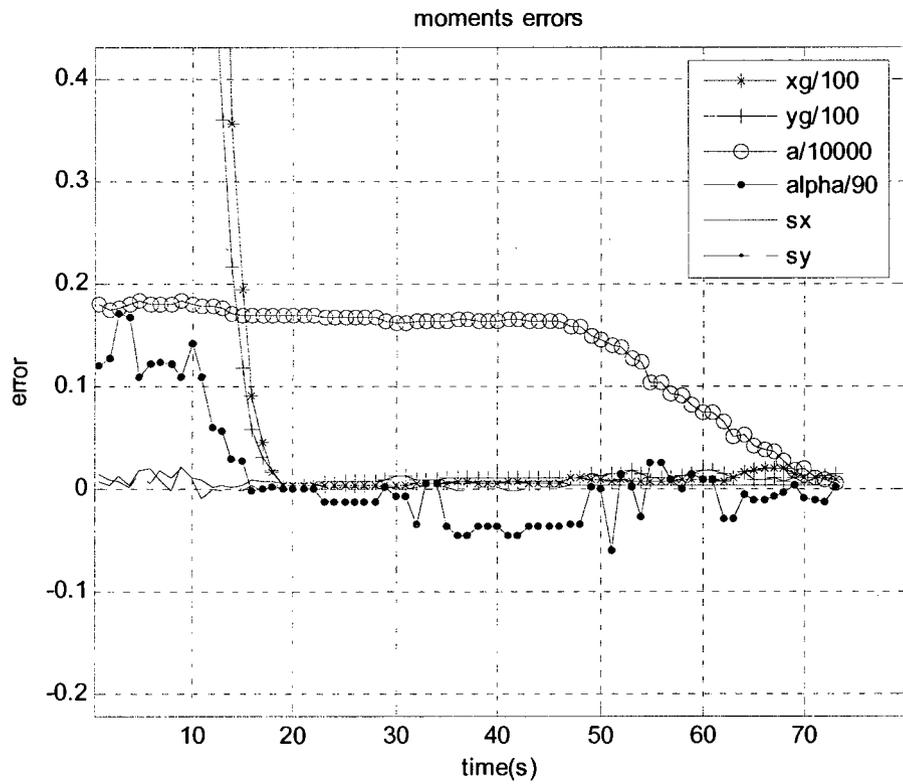


Figure 5-14 Zoom in of Figure 5-13

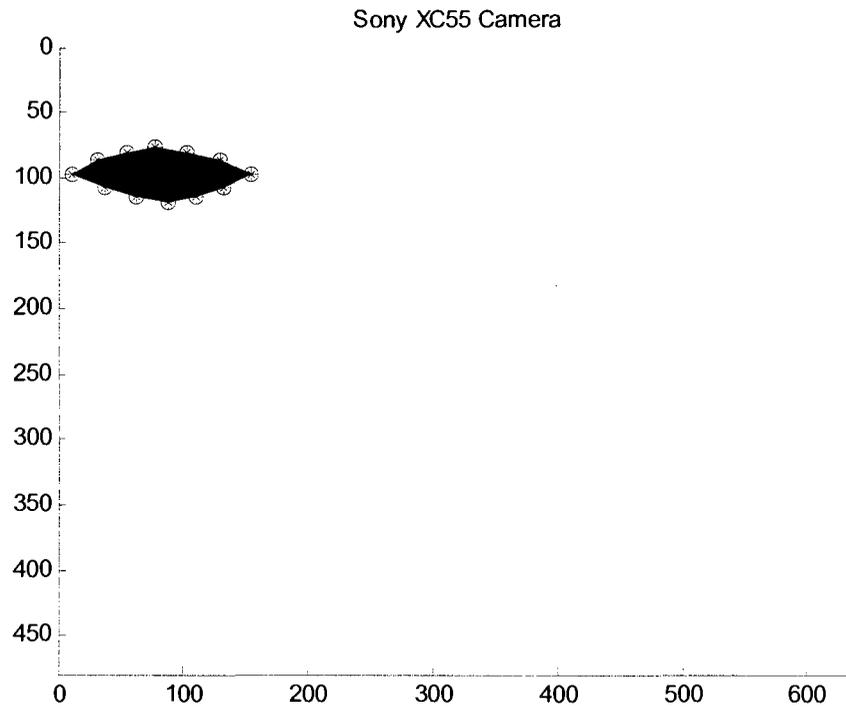
- Date core shape object (task 4)

In task 4, the object is the date core shape. Initially, It rotates around x axis by 40 degrees.

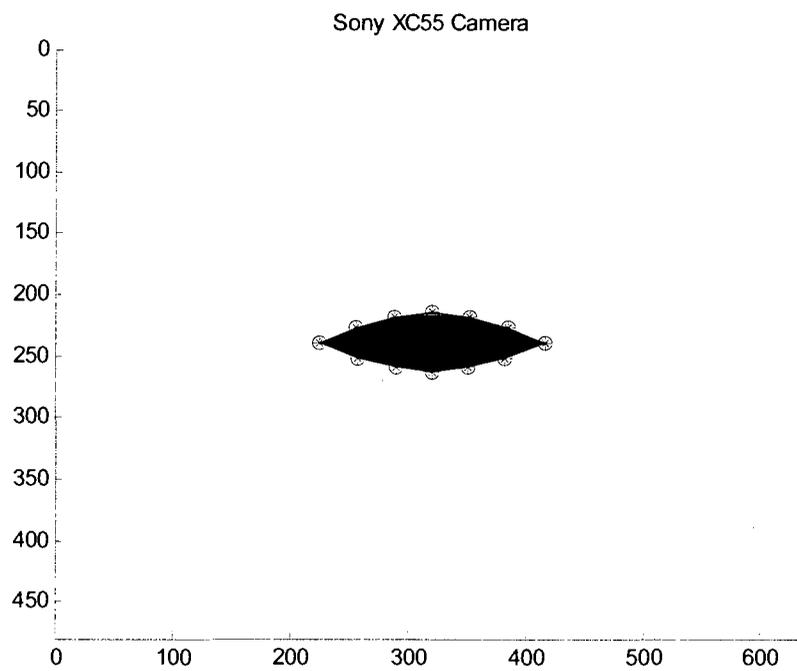
**Table 5-3 The desired and initial image feature values for task 4**

Image features	Desired value	Initial value for task 4
$x_g$	0	-238.5
$y_g$	0	-143.1
$a$ (number of pixels)	6000	3299
$\alpha$ (degrees)	0	0
$s_x$	0.097	0.1129
$s_y$	0.006	-0.003

Figure 5-15 is the initial image of task 4. And the following figure shows the final result. Figure 5-17 is the error of the whole process. It can be seen that all the image feature errors have been controlled to zero, which means the algorithm is successful for this kind of shape.



**Figure 5-15 Initial image of task 4**



**Figure 5-16 Final image of task 4**

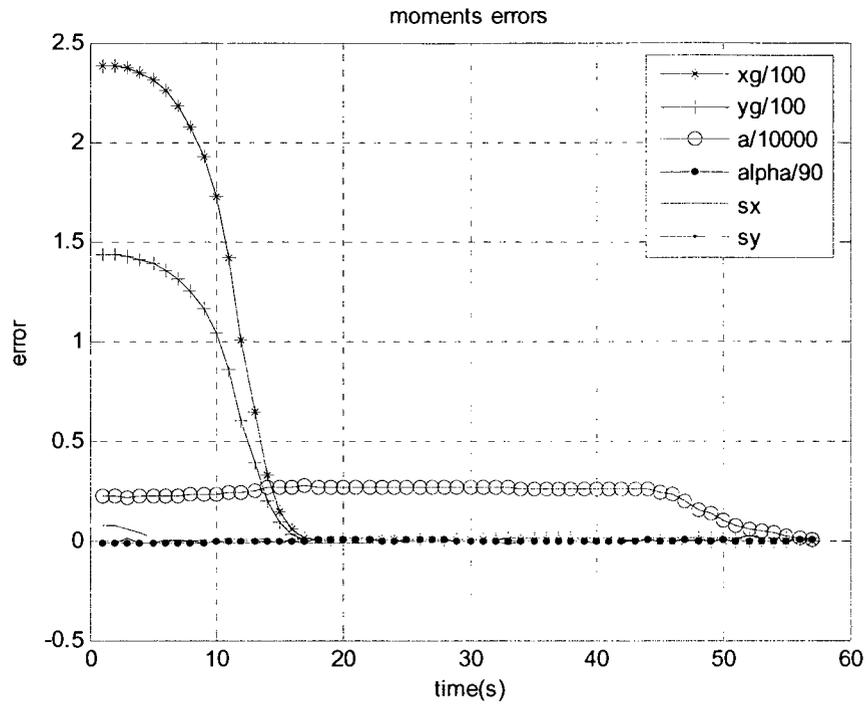


Figure 5-17 Errors of image features of task 4

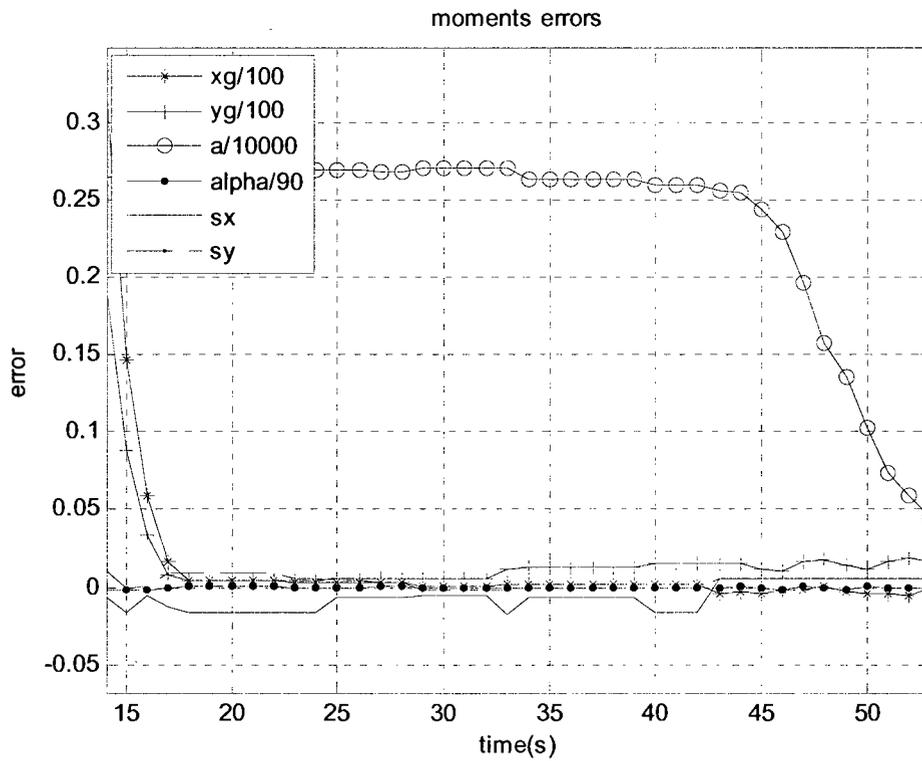
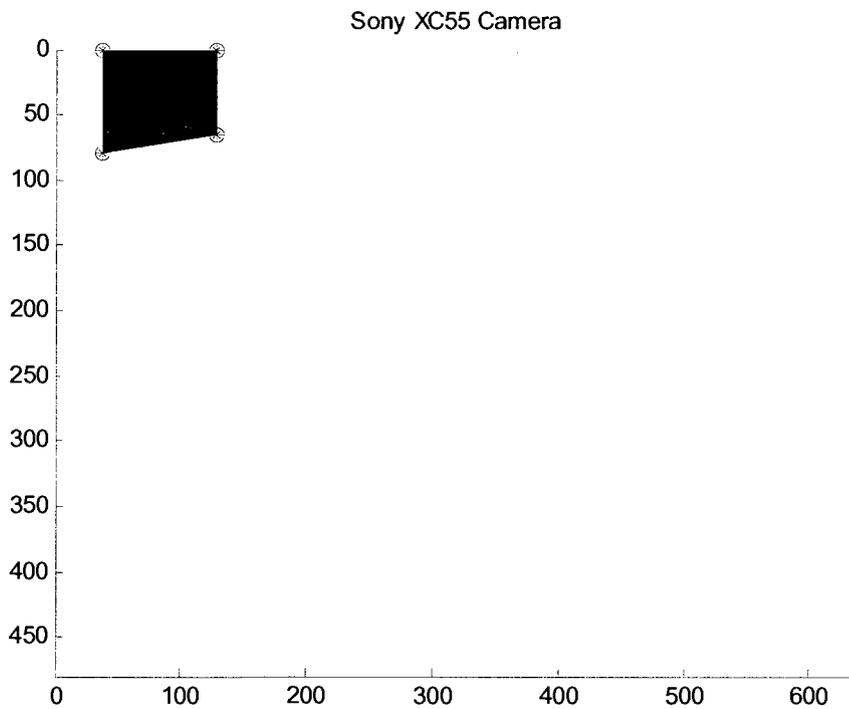


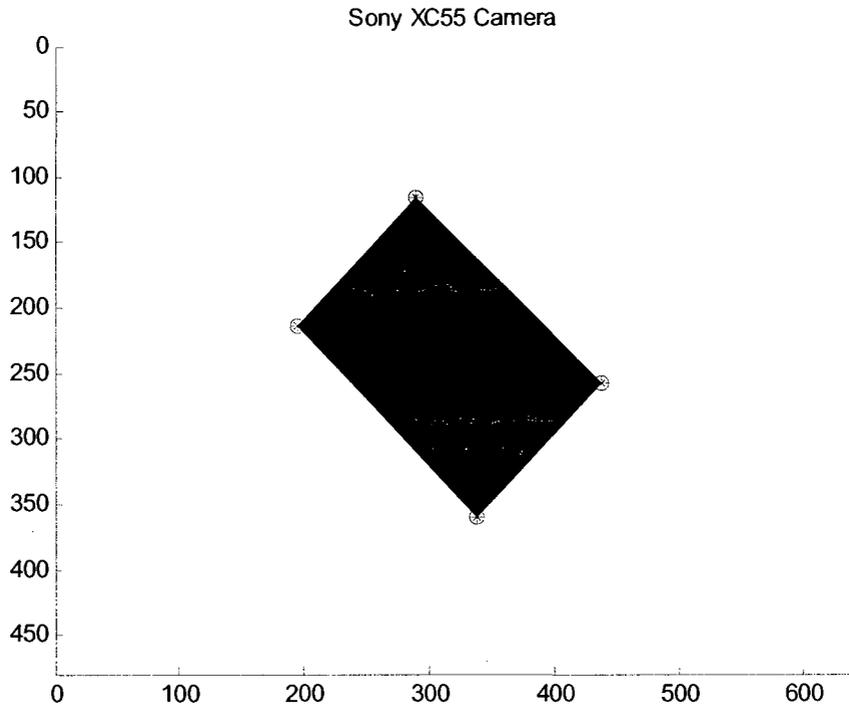
Figure 5-18 Zoom in of Figure 5-17

## 5.2.2 Robustness test

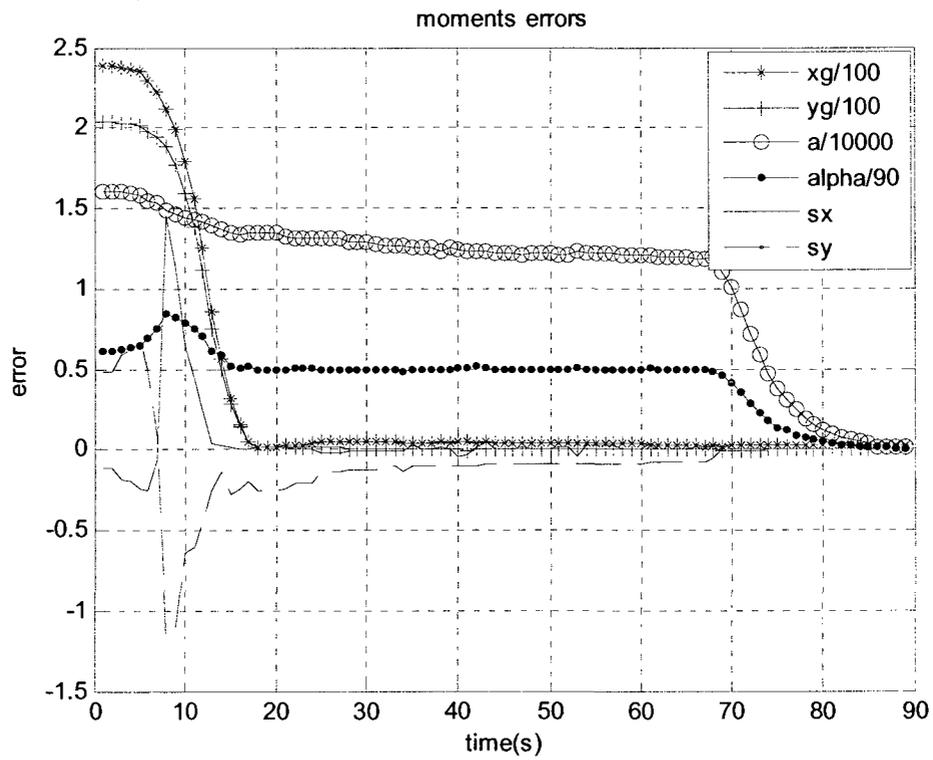
In order to test the robustness of system under the bad camera calibration, 5% error has been added to the focus length. The object and initial condition are the same as those in task 1. The initial and final images are shown in Figures 5-19 and 5-20. In Figure 5-21, the errors are displayed. Although the system needs longer time to process, it can still track the target object without being affected by the bad camera calibration.



**Figure 5-15 Initial image of robustness test**



**Figure 5-20 Final image of robustness test**



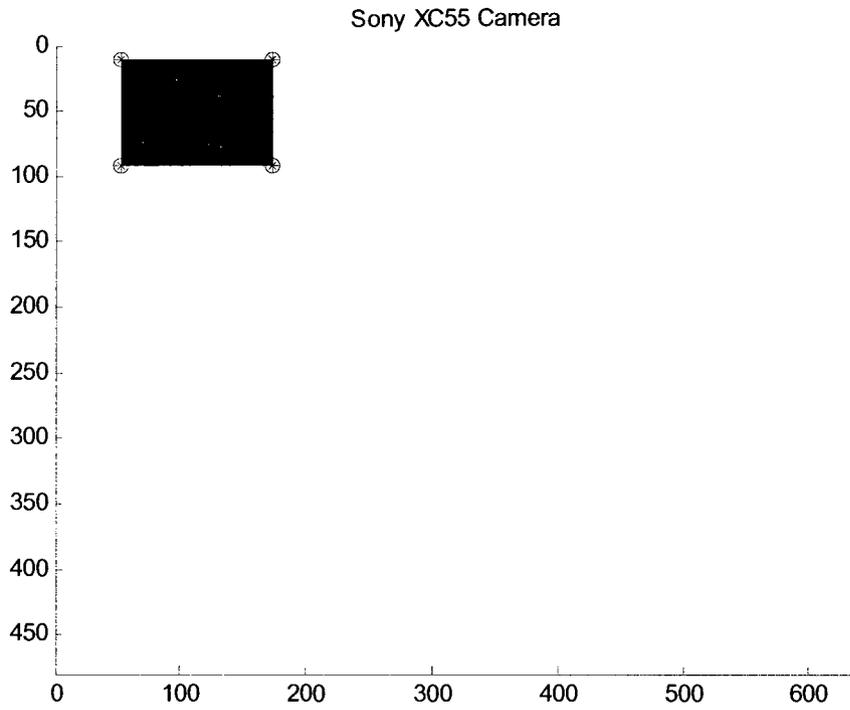
**Figure 5-21 Errors of image features of robustness test**

### 5.2.3 Comparison

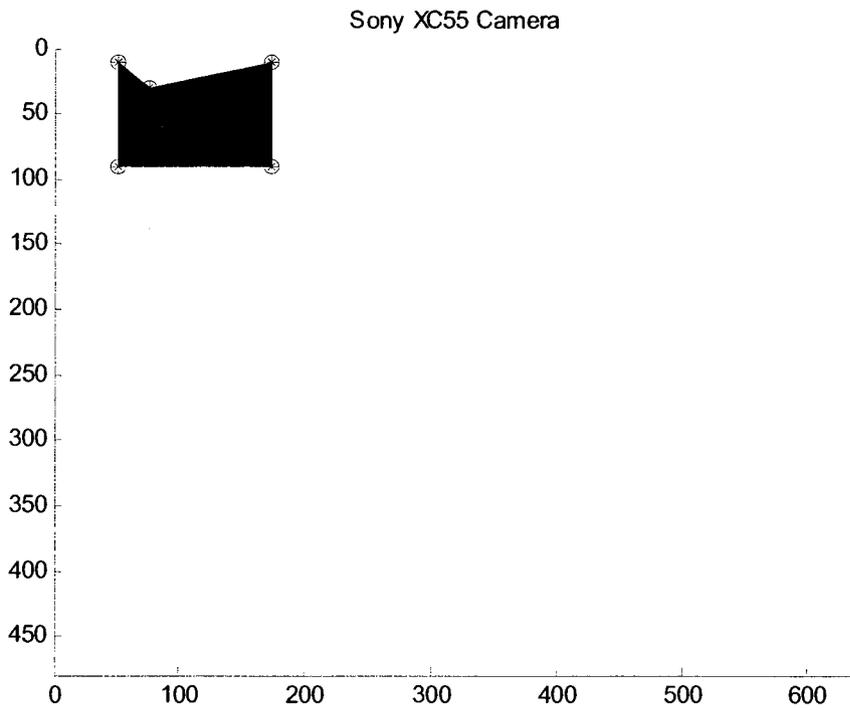
The reasons that we choose the more complicate image moments as the image features rather than using the point directly are stated in detail as follows:

First of all, for the objects with enough corners (at least three), the corners can be considered as the image features. However, if the target objects do not have enough corners or even do not have any special point which can be detected as the image features, the point based method might not work anymore. Here we take task 4 as an example. For the point based method, the points which can be marked as the image features are the two acmes. If the object rotates around x axis, the position of these two acmes will not change but the whole object has been changed actually. In other words, these two acmes cannot fully represent the pose of object correctly. In contrast, as proved above in task 4, the moment based method can achieve the goal successfully.

Another reason is that if partial object is covered by some unexpected things such as wastepaper as shown in Figure 3-1, the point based method will fail because the mismatch of the number of image features. However, if the image features are chosen from the image moment, there will only be some errors but will not cause system failure directly. Following are the images in two cases. The regular situation represents the normal working environment without any unexpected disturbance. The irregular shape means some part of target object has been covered.



**Figure 5-22 Image in regular situation**



**Figure 5-16 Image in irregular situation**

Table 5-4 shows the comparison of point based method and moment based method under both regular and irregular situations.

**Table 5-4 Comparison of point based method and moment based method**

	Point-based method	Moment-based method
$s_d$ : Desired values of image features	$\begin{cases} [-24.8, -122.5]^T \\ [-122.1, -24.4]^T \\ [25.1, 121.6]^T \\ [122.4, 23.5]^T \end{cases}$	$\begin{cases} 0 \\ 0 \\ 28800 \\ 0.1 \\ 0 \\ 45^\circ \end{cases}$
$s_{c1}$ : current values of image features in regular situation	$\begin{cases} [-274.04, -234.4]^T \\ [-274.04, -153.33]^T \\ [-152.4, -153.33]^T \\ [-152.4, -234.4]^T \end{cases}$	$\begin{cases} -212.5 \\ -194.0 \\ 10130 \\ 0.1 \\ 0 \\ 0^\circ \end{cases}$
$e_1$ : Errors between $s_d$ and $s_{c1}$	$\begin{cases} [249.24, 111.9]^T \\ [151.9, 128.93]^T \\ [177.5, 274.93]^T \\ [274.8, 257.9]^T \end{cases}$	$\begin{cases} 212.5 \\ 194.0 \\ 18670 \\ 0 \\ 0 \\ 45^\circ \end{cases}$
$s_{c2}$ : current values of image features in irregular situation	$\begin{cases} [-274.04, -234.4]^T \\ [-274.04, -153.33]^T \\ [-152.4, -153.33]^T \\ [-152.4, -234.4]^T \\ [-249.38, -213.46]^T \end{cases}$	$\begin{cases} -211.8 \\ -189 \\ 8990 \\ 0.09778 \\ 0.006142 \\ -4.027^\circ \end{cases}$
$e_2$ : Errors between $s_d$ and $s_{c2}$	<p>Numbers do not match</p> <p>FAILED</p>	$\begin{cases} 211.8 \\ 189 \\ 19810 \\ 0.00222 \\ -0.006142 \\ 49.027^\circ \end{cases}$

From the table, it can be seen that if the Point-based method is adopted, the number of errors in regular situation is 8 (4 sets) while 10 (5 sets) in irregular situation. The system will not work because of the mismatching. In contrast, for the moment based method, although there exist some warps between  $e_1$  and  $e_2$ , the number of image features is matched and the system will work normally.

On all accounts, the moment based method can deal with some problems that cannot be solved by point based method. Furthermore, the moment based method is more robust than the point based one.

### **5.3 Summary**

In this chapter, the simulation results of image based visual servoing using the image moments have been given. Three object shapes and four tasks have been tested. The robustness tests and comparisons are also provided. The results show that the proposed control algorithm can achieve the visual servoing goal successfully. Moreover, the proposed control method is robust to the calibration error and outperforms the traditional point based visual servoing.

## CHAPTER 6 CONCLUSION AND FUTURE WORK

### 6.1 Introduction

In this chapter, the conclusion of the thesis from the previous chapters will be summarized. Furthermore, the possible extension and future work will be discussed.

### 6.2 Conclusion

This thesis has presented an image pre-processing method and an improved image based visual servoing algorithm. The conclusion and motivations are summarized as follow:

1. The image pre-processing part.

An ameliorative algorithm for image pre-processing has been proposed. It can be applied before the image features extraction to adjust the illumination condition and remove the unexpected dust and speckles.

This algorithm is based on the morphological theory. Referring to the results, it has been shown that this algorithm can remove the speckles efficiently. For the bad or partial illumination condition, it can also balance the background illumination and obtain the clear and useful image.

2. The image based visual servoing part.

Image moments have been chosen as the image features. Specially, a new set of image moments derived from Hu's invariants has been introduced to control the 3D rotation. After testing, it showed that this set of image moments can represent the object pose correctly. They will not be sensitive to the disturbance due to the low orders.

Furthermore, a novel depth estimation method has been introduced. It is derived from the homogeneous transformation matrix between the robot frame, which is also the world frame, and the camera frame. Compared to other depth estimation methods, this method is relatively simple and easy to compute.

Also, an improved IBVS algorithm has been proposed. In this algorithm, the image moments have been chosen as the image features to create the control signals. Specially, two new image features which are combined by image moments have been chosen to control the 3D rotation. In order to define the relationship between velocities of the novel image features and the move of camera, the derivation of image Jacobian matrix is provided.

According to the results shown in Chapter 5, it demonstrates that the proposed image features can represent the pose of object correctly and help creating the proper controller. Furthermore, it also proves that the improved algorithm can be applied to not only the simple rectangular object, but also the other shapes. It can even be applied to the object which is difficult to detect the corner point. Besides, this algorithm is also robust to the poor camera calibration.

To sum up, an improved algorithm of image visual servoing inherit the advantage of the traditional IBVS. At the same time, it has conquered the disadvantage of the traditional point based IBVS.

### **6.3 Future work**

Although the proposed IBVS algorithm can achieve the visual servoing goal successfully, there are also some further works worth of pursuing in the future. Some principal suggestions are listed as follows.

1. To test the system on a real robot with real time control to evaluate the proposed IBVS algorithm.
2. To extend the one camera system to the binocular cameras, so that the thickness of object can also be considered.
3. To track the moving object instead of the fixed one.

## BIBLIOGRAPHY

- [1] S. Hutchinson, G. D. Hager, and P. I. Corke, "A tutorial on visual servo control,"  
IEEE Trans. on Robotics and Automation, vol. 12, pp. 651–670, Oct. 1996.
- [2] N. Aouf, H. Rajabi, N. Rajabi, H. Alanbari and C. Perron, "Visual object  
tracking by a camera mounted on a 6DOF industrial robot," Proc. of IEEE  
Conference on Robotics, Automation and Mechatronics, vol. 1, pp. 213-218,  
Dec. 2004.
- [3] L. Deng, "Comparison of image-based and position-based robot visual servoing  
methods and improvements," PhD dissertation, Electrical and Computer  
Engineering at University of Waterloo, 2003
- [4] B. Espiau, "Effect of camera calibration errors on visual servoing in robotics",  
Experimental Robotics III: The 3rd International symposium on experimental  
robotics, Springer-Verlag, Lecture notes in control and information sciences, vol.  
200, pp. 182-192, 1993
- [5] F. Chaumette, "Potential problems of stability and convergence in image-based  
and position-based visual servoing," The confluence of Vision and Control,  
Lecture Notes in Control and Information Science, D.Kriegman, G. D. Hager,  
and A. Morse, Eds., vol. 237. New York: Springer Verlag, pp. 66-78, 1998.
- [6] [http://www.fanucrobotics.com/Public\\_Photo\\_Gallery/assembly\\_robot\\_images.htm](http://www.fanucrobotics.com/Public_Photo_Gallery/assembly_robot_images.htm)
- [7] <http://www.robots.com/applications.php?app=pick+and+place>
- [8] <http://www.robots.com/applications.php?app=packaging>

- [9] N. J. Cowan and D. E. Koditschek. "Planar image based visual servoing as a navigation problem," Proc. of IEEE Int. Conf. Robotics and Automation, pp. 611-617, 1999.
- [10] K. Hashimoto and T. Noritsugu, "Potential problems and switching control for visual servoing," Proc. of IEEE/RSJ International Conference on Intelligent Robots and System, vol. 1, pp. 423-428, Oct. 2000
- [11] E. Malis, F. Chaumette, and S. Boudet. "2-1/2-D visual servoing," IEEE Trans. Robotics and Automation, vol. 15, no. 2, pp.238-250, 1999.
- [12] P. I. Corke and S. A. Hutchinson, "A new partitioned approach to image-based visual servo control," IEEE Trans. Robotics and Automation, vol. 17, no. 4, pp. 507-515, August 2001.
- [13] K. Hashimoto, T. Kimoto, T. Ebine and H. Kimura, "Manipulator control with image-based visual servo," Proc. of the 1991 IEEE Int. Conf. on Robotics and Automation, pp. 2267-2272, April 1991.
- [14] O. Bourquardez and F. Chaumette, "Visual servoing of an airplane for alignment with respect to a runway," Proc. of IEEE Int. Conf. on Robotics and Automation, pp. 1130-1135, April 2007.
- [15] Z. Li,; W.F. Xie; X.W. Tu, "Switching control of image based visual servoing with laser pointer in robotic assembly systems," Proc. of IEEE Int. Conf. on Systems, Man and Cybernetics, pp. 2383 - 2389, Oct. 2007.
- [16] C. Collewet and F. Chaumette, "A contour approach for image-based control of objects with complex shape," Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.,

vol. 1, pp. 751-756, Nov. 2000.

- [17] F. Chaumette, "Image moments: a general and useful set of features for visual servoing," *IEEE Trans. on Robotics*, vol.20, no.4, pp. 713-723, Aug. 2004.
- [18] R. J. Prokop, A. P. Reeves, "A survey of moment-based techniques for unoccluded object representation and recognition". *CVGIP: Graphical Models and Image Processing* vol.54, no.5, pp. 438--460, Sep. 1992.
- [19] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. on Information Theory*, vol. IT-8, pp.179-187, Feb. 1962.
- [20] O. Tahri and F. Chaumette, "Point-based and region-based image moments for visual servoing of planar objects," *IEEE Trans. on Robotics*, vol. 21, no. 6, pp. 1116-1127, Dec. 2005.
- [21] W. J. Wilson, "Vision sensor integration for direct manipulator end-point control," *Proceedings of Robots 12 / Vision '88 Conference*, June 1988.
- [22] A. G. Mamistvalov, "n-dimensional moment invariants and conceptual theory of recognition n-dimensional solids," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 8, pp. 819-8319, Aug. 1998.
- [23] C. C. W. Hulls, "Dynamic real-time multisensor fusion using an object model reference approach," Ph.D. dissertation, University of Waterloo, 1996.
- [24] E. Dougherty and R. Lotufo, "Hands-on morphological image processing, ser. Tutorial Tests in Optical Engineering." *SPIE Press*, vol. TT59, 2003.
- [25] P. Siva, C. C. W. Hulls, "Dynamic segmentation of small image windows for visual servoing", *Proc. of IEEE Int. Conference on Mechatronics & Automation*,

pp. 643-648, July 2005.

- [26] [http://cmm.ensmp.fr/~serra/pdf/birth\\_of\\_mm.pdf](http://cmm.ensmp.fr/~serra/pdf/birth_of_mm.pdf)
- [27] [http://en.wikipedia.org/wiki/Mathematical\\_morphology](http://en.wikipedia.org/wiki/Mathematical_morphology)
- [28] N. Otsu, "A threshold selection method from gray-level histograms", IEEE Trans. On Systems, Man, and Cybernetics, vol. SMC-9, no. 1, pp. 62-66, Jan. 1979
- [29] G. Well, C. Venaille, and C. Torras, "Vision-based robot positioning using neural networks", Image Vis. Comput., vol. 14, no. 10, pp. 715-732, Dec. 1996
- [30] P. I. Corke, "Robotics toolbox for MATLAB," IEEE Trans. on Robotics and Automation, vol 3, issue 1, pp.24 – 32, March 1996.
- [31] Motoman UPJ Reference Manual, MOTOMAN INC., 2003
- [32] <http://www.avsupply.com/details/xc-55.shtml>
- [33] N. R. Gans, P. I. Corke and S. A. Hutchinson, "Performance tests of partitioned approaches to visual servo control", Proc. of the 2002 IEEE Intl. Conf. on Robotics & Automation, pp. 1616-1623, May 2002.
- [34] F. Chaumette, "A first step toward visual servoing using image moments", 2002 IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems EPFL, pp. 378-383, Oct. 2002.
- [35] S. Maitra, "Moment invariants", Proceedings of the IEEE, vol.67, no.4, pp. 697-699, April 1979,
- [36] A. Krupa, C. Doignon, J. Gangloff, M. de Mathelin, "Combined image-based and depth visual servoing applied to robotized laparoscopic surgery", Proc. of

2002 IEEE/RSJ Intl. Conf. Intelligent Robots and Systems EPFL, Lausanne, Switzerland, pp. 323-329, Oct. 2002.

- [37] A. Krupa, J. Gangloff, "Autonomous retrieval and positioning of surgical instruments in robotized laparoscopic surgery using visual servoing and laser pointers", Proc. of 2002 IEEE intl. conf. Robotics & Automation, Washington, DC, pp. 3769-3774, May 2002.
- [38] E. Malis, F. Chaumette, S. Boudet, "Multi-cameras visual servoing", Proc. of 2000 IEEE intl. conf. Robotics & Automation, San Francisco, CA, pp. 3183-3188, April 2000.
- [39] V. Lippiello, B. Siciliano, L. Villani, "Eye-in-hand/Eye-to-hand multi-camera visual servoing", Proc. of 44<sup>th</sup> IEEE conf. on Decision and Control, and the European Control Conf. 2005, pp. 5354-5359, Dec. 2005..
- [40] N. P. Papanikolopoulos, P. K. Khosla, T. Kanade, "Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision", IEEE Trans. on Robotics and Automation. vol. 9, no. 1, pp. 14-35, Feb. 1993.
- [41] R. Kelly, "Robust asymptotically stable visual servoing of planar robots", IEEE Trans. on Robotics and Automation, vol. 12, no. 5, pp. 759-766, Oct. 1996.
- [42] L. Deng, F. Janabi-Sharifi, W.J. Wilson, "Stability and robustness of visual servoing methods", Proc. of 2002 IEEE Intl. Conf. on Robotics & Automation, pp. 1604-1609, May 2002.
- [43] J. T. Feddema, C. S. G. Lee, and O. R. Mitchell, "Weighted selection of image features for resolved rate visual feedback control," IEEE Trans. on Robotics

and Automation, vol. 7, no. 1, pp. 31-47, Feb. 1991.

- [44] R. Kelly, R. Carelli, “Stable visual servoing of camera-in-hand robotic systems”, IEEE/ASME Trans. on Mechatronics, vol. 5, no. 1, pp.39-48, March 2000.
- [45] P. Mahony, P. Corke, and F. Chaumette, “Choice of image features for depth-axis control in image-based visual servo control”, Proc. IEEE/RSJ Int. Conf. Intelligent Robots and System, vol. 1, pp. 390-395, Oct. 2002.
- [46] X. Shen, J. Pan, “A simple adaptive control for visual servoing”, Proc. of Second Intl. Conf. on Machine Learning and Cybernetics, pp. 976-979, Nov. 2003.
- [47] Y. Zhang 2005, “Image Engineering (II): Image Analysis”, Beijing: Tsinghua University Press
- [48] F. Chaumette, “Potential problems of stability and convergence in image based and position-based visual servoing,” in The Confluence of Vision and Control, D. Kriegman, G. D. Hager, and A. Morse, Eds. Berlin, Germany: Springer-Verlag, vol. 237, pp. 66-78. Lecture Notes in Control and Information Sciences, 1998.
- [49] Cheng Gao, Xin-he Xu, “Optimal motion control for IBVS by separating rotation in z-axis from general motion,” Proc. of the Seventh International Conference on Machine Learning and Cybernetics, pp 2061-2066, July 2008.

- [50] Zhen Qi and John E. McInroy, "Nonlinear image based visual servoing using parallel robots," 2007 IEEE Intl. Conf. on Robotics and Automation, pp 1715-1720, April 2007.
- [51] Miao Hao, Zengqi Sun, Masakazu Fujii, "Image based visual servoing using Takagi-Sugeno fuzzy neural network controller," 22nd IEEE Intl. Symposium on Intelligent Control Part of IEEE Multi-conference on Systems and Control, pp. 53-58, Oct. 2007.
- [52] J. P. Wang and H. Cho, "Micropeg and Hole Alignment Using Image Moments Based Visual Servoing Method," IEEE Trans. on Industrial Electronics, vol. 55, no. 3, pp. 1286-1294, March 2008.
- [53] XIE W.F., Z. Li, X. Tu, C. Perron, "Switching Control of Image Based Visual Servoing With Laser Pointer in Robotic Manufacturing Systems," IEEE Trans. on Industrial Electronics, vol. 56, no.2, pp.1-10, Feb.2009.

## APPENDIX A

### Hu's invariance [18]

Hu has defined seven values from central moments. They are invariant to object scale, position and orientation change. These seven invariants are shown as follow:

$$M_1 = \mu_{20} + \mu_{02}$$

$$M_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$$

$$M_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2$$

$$M_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$$

$$M_5 = (\mu_{30} - 3\mu_{12})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\ + (3\mu_{21} - \mu_{03})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2]$$

$$M_6 = (\mu_{20} - \mu_{02})[(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2 + 4\mu_{11}(\mu_{30} + \mu_{12})(\mu_{21} + \mu_{03})]$$

$$M_7 = (3\mu_{21} - \mu_{03})(\mu_{30} + \mu_{12})[(\mu_{30} + \mu_{12})^2 - 3(\mu_{21} + \mu_{03})^2] \\ - (\mu_{30} - 3\mu_{12})(\mu_{21} + \mu_{03})[3(\mu_{30} + \mu_{12})^2 - (\mu_{21} + \mu_{03})^2]$$