REPUBLIC OF ALGERIA, DEMOCRATIC AND POPULAR
MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH
ABOU BEKR BELKAID UNIVERSITY OF TLEMCEN
Faculty of Technology

These

Presented in order to acquire a Doctorate Degree in the (3^{rd} Cycle)

In: Automatic

Specialization: Automatic

By: DERRAR Yasser

Title

Asservissement visuel sur objets déformables

"Visual Servoing on Deformable Objects"

Publicly defended on: 11/12/2025, before the jury composed of:

Mr. BELKAID Fayçal	University of Tlemcen	Professor	President
Mr. BOUMEDIENE Abdelmadjid	University of Tlemcen	Professor	Supervisor
Mrs. SAIDI Farah	University of Tlemcen	MCB	Co-supervisor
Mr. BENNEKROUF Mohammed	ESSA of Tlemcen	MCA	Examiner
Mrs. BENKHALED Mounira	University of Tlemcen	MCA	Examiner
Mr. BENSALAH Choukri	University of Tlemcen	MCA	Examiner

Contents

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my supervisor, **Pr.BOUMEDIENE Abdelmadjid**, for accepting to supervise this doctoral work, and for his rigorous scientific guidance, valuable advice, and constant support, which greatly contributed to the successful completion of this thesis.

I am also deeply thankful to my co-supervisor, **Dr.SAIDI Farah**, for her constructive feedback, continuous encouragement, and insightful remarks that positively influenced the quality and direction of this research.

I would like to acknowledge **Dr.MALTI Abed** for his supervision and guidance during an earlier phase of this work, which provided an essential scientific foundation for the development of this research.

My sincere thanks go to my parents for their unconditional support, both moral and financial, throughout my years of study, and for encouraging me to pursue and achieve my goal of becoming a doctor.

I would like to warmly thank my sister, **DERRAR Soumia Djamila**, for her invaluable help and support at the beginning of my PhD journey, and for her continuous encouragement.

I am profoundly grateful to my wife, **FEHAIMA Sarra**, for her patience, understanding, and unwavering support throughout this doctoral work. I also dedicate this work to my beloved children, **Yousra** and **Nadjib**, who have been a constant source of motivation and strength.

I thank my colleague "Farah" and "Malik" for their support.

I would also like to thank Mr **Pierre Jannin**, Professor at the University of Rennes 1, Laboratoire Traitement du Signal et de l'Image LTIS INSERM, U1099, who welcomed me during my internship period. It is thanks to him that I was able to happily reconcile theoretical and applied research during this thesis.

This thesis is the result of the support, trust, and contributions of all those who accompanied me, directly or indirectly, during this journey.

Abstract

Français

Les robots médicaux guidés par l'image sont actuellement utilisés dans tous les domaines de la chirurgie leur objectif commun est d'offrir une précision inégalable manuellement pour assurer une réalisation optimale de la procédure chirurgicale.
Le but de la thèse est d'établir un moyen automatique pour contrôler un capteur tel un échographe, endoscope... pour suivre la déformation des objets (organes ,tissu..) et réduire la difficulté de vision et de suivi par les chirurgiens lors de ces interventions.

Les stratégies de commande envisagées dans ce domaine peuvent être divisées en deux catégories selon qu'elles reposent sur une planification pré-opératoire du geste robotique enrichie ou non d'un recalage avec une image per-opératoire, ou bien qu'elles utilisent la seule image per-opératoire pour guider le robot. Cette seconde approche permet de réaliser la tâche robotique en tenant compte des déformations et des mouvements des organes dus aux mouvements involontaires ou physiologiques du patient. Elle est ainsi particulièrement adaptée pour les opérations sur les tissus mous où l'image per-opératoire permet de visualiser la position réelle des organes tout au long de la procédure. Des tâches de positionnement et de suivi de coupes anatomiques sont réalisées où un capteur , manipulée par un bras robotique En particulier dans l'optique d'une application de suivi d'une coupe désirée en compensant les mouvements physiologiques du patient, une commande optimale pour garantir une meilleure précision du suivi. La méthode proposée ne nécessite pas la connaissance du modèle de déformation et suppose un suivi de points de maillage de l'objet déformable. Les résultats obtenus montrent que la méthode proposée peut garantir un suivi de déformation tout au long d'une trajectoire planifiées. Cette validation a pu etre testées sur plusieurs type de déformations et de trajectoires de caméra. La robustesse au bruit de détection a aussi été expérimentalement démontrée.

Keywords : Asservissement visuel, Objets déformables, Positionnement caméra rigide, Déformation de suivi.

English

Image-guided medical robots are currently used in all fields of surgery and their common objective is to offer an unparalleled manual precision to ensure an optimal realization of the surgical procedure.
The goal of the thesis is to establish an automatic way to control a sensor such as an ultrasound scanner, endoscope ... to follow the deformation of objects (organs, tissue ...) and reduce the difficulty of vision and monitoring by surgeons during these interventions. The control strategies considered in this field can be divided into two categories depending on whether they are based on a pre-operative planning of the robotic gesture enriched or not with a registration with a per-operative image, or whether they use the only per-operative image to guide the robot. This second approach allows to perform the robotic task taking into account the deformations and the movements of the organs due to the involuntary or physiological movements of the patient. It is thus particularly adapted for soft tissue operations where the intra-operative image allows to visualize the real position of the organs throughout the procedure. Especially in the perspective of a tracking application of a desired cut by compensating the physiological movements of the patient, an optimal control to ensure better accuracy of tracking is proposed.
The proposed method does not require any prior knowledge on the deformation model. It assumes as known the set of visible vertices composing the mesh of the target deformable object. The obtained results show that the proposed method enable us to track object deformations and allows us to compensate camera motion according to a pre-planned trajectory. The validation was driven on multiple type of deformations and multiple type of camera trajectories. Robustness to noise in vertex detection has also been investigated.
Keywords : Visual servoing, Deformable objects, Camera positioning, Tracking deformation.

Notation

- s : Visual primitive.
- L : The interaction matrix.
- $\dot{c}$: Camera speed.
- m_{ij} : The moments the image.
- e : Errer.
- w_i : The weights are preserved since the triangle's area varies continuously during the deformation.
- ρ, Θ : The representation that defines the configuration of the straight line.
- $g(\rho, \Theta)$: It's primitive 2D.
- Z : Depth in the reference of the camera.
- $D(t)$: "The region in the image onto which the 3D triangle is projected."
- $C(t)$: The contour edges of region $D(t)$.
- $\vec{n}$: The normal to the triangle edge
- $L_{\mu_{ij}}$: The Centered moments.
- L_θ : The triangle orientation.
- L_a : The area of the triangle being.
- L_{x_g}, L_{y_g} : The projection of the triangle's center of gravity in the image is considered for further calculations.
- T : Triangle mesh.
- c : Is a reference camera configuration.
- $w_i(u)$: Values of the weights for abscissa u along the trajectory.
- Π_{C,T_0} : The projected vertices $c_0$on the camera plane while moving.
- (X, Y, Z) : A three-dimensional vertex expressed in the coordinate system of the camera.
- im : It denotes the location of the 3D point in the image coordinate system after projection.

- v : The translation velocity of the cameras.
- w : The rotation velocity.
- (t_x, t_y, t_z) : The 3D position coordinates.
- $\theta_x, \theta_y, \theta_z$: The angle of rotation.
- $\gamma(s)$: Reference trajectory.

Introduction

General topic and background

The field of visual servoing encompasses a broad spectrum of practical applications. Also known as vision-based control, this technique leverages visual data captured by a camera to guide and regulate the motion of a robotic system or device. We can find it several industries (car assembly industry, waste sorting...), aviation (Automatic drones) and also robotic surgery. Applications may differ in term of used sensors (camera, ultrasound probe...). Robotic surgery refers to a category of surgical procedures performed with the aid of robotic systems. This approach was introduced to address the limitations of traditional minimally invasive techniques and to augment the precision and capabilities of surgeons during open surgical operations. Indeed, a surgeon performing surgical operation by hand is subject to stress, a more or less significant lack of accuracy and other inevitable human factors that may alter the quality of the final product of the surgical operation. In robotically assisted minimally invasive surgery, the surgeon operates through a telemanipulator a remote control interface allowing precise manipulation of surgical instruments mounted on the robot end-effectors to replicate standard surgical gestures. We cite as an example the DaVinci surgical system which is the most popular surgery robot till now. This contributes to significantly increase the quality of performed operation and avoiding serious post-operation complications. An attempt to remove completely the human intervention in surgical operation is to include visual servoing in robot assisted surgery. Indeed, several works attempted to do so. It consists in automatising a surgical system by allowing it to control and position its surgical instruments based on information captured from the sensor. If the sensor is a camera, the information will be real time images of the surgical scene or more precisely, a targeted object like a tumour on a tissue. This thesis explores the application of visual servoing techniques to deformable objects, with the use of a camera as the sensor for precise monitoring and control. The considered deformable object is a virtually simulated liver that can perform realistic deformations.

Literature

Historically, the first work on visual servoing in medical robotics was done in the framework of laparoscopic surgery where the imaging device used is an endoscope consisting of a camera coupled to a special cylindrical camera coupled to a special cylindrical lens.

This long lens is introduced into the patient's abdomen through a trocar previously positioned at the point of incision. A network of optical fibres placed around the periphery of the optic allows the patient's abdomen the illumination of a cold light source located outside. This type of medical image has the advantage of benefiting from the good properties of cameras, which include a high image acquisition rate (25 frames/s or more) and the presence of a low acquisition delay (around 20 ms). It should therefore be integrated into robotic applications with visual servoing using a kinematic or dynamic control architecture.

The first works[66], [44], [87], focused on the automatic positioning of an endoscope carried by a robot. To track and continuously maintain one or more surgical instruments within the camera's field of view during a surgical procedure, eliminating the need for the surgeon to manually adjust the endoscope. The applications implemented were: automatic retrieval by visual servo of an instrument in the visual field of the camera[43], an automatic suturing task [31]performed by 3D visual servoing after indication in the image of the entry and exit point of a circular needle and the compensation of physiological movements during the procedure of the patient by means of a dynamic control using a predictive corrector based on a model of the organ's movement [31]. The use of vision to guide a robot within the human body is constrained by challenges in image interpretation. Consequently, the works mentioned above have mostly considered visual information provided by artificial markers added in the scene. These markers have been attached to the tip of the instruments[44], [31], [87], [66] and/or also to the surface of the organ of interest. The surface of the organ of interest is illuminated and projected using a laser spot.[44] or by attaching light markers [31], [43].

Outline the current situation

There are several ongoing works in the field of visual servoing that aim to develop new control laws and to find robust commands. These works are done in regards to the nature of the object (deformable or non-deformable) but there is a significant focus on the deformable objects: This is because a deformable object tends to present a lot of transformations and several parameters. Modelling the behaviour of non-rigid objects under constraints is a complex problem for which an analytical solution is not always possible.

Evaluate the current situation (advantages/disadvantages) identify the gap

Few works address the case of the visualization of deformable objects. The increase in automation needs, the development of new applications in the fields of surgery, the food industry and personal services make it necessary to study input and visualization. De-

formable objects by robotic systems in order to increase their capacities. This increased complexity is mainly related to the deformable behaviour of materials. Real-time modelling and simulation is one of the challenges of visualizing deformed objects. Modeling the interaction between the processor and the deformable object presents significant challenges. The complexity of this interaction stems from the intricate relationship between the object's shape change and the tracking of its deformation.

Identify the importance of the proposed research

The vision of rigid bodies by a robotic camera is an already very complex subject of study involving many processes. When bodies are deformable, new scientific issues appear, making visualization even more complex. This increased complexity is essentially linked to the behaviour of deformable materials.One of the key challenges in visualizing deformable objects is real-time modeling and simulation. Additionally, modeling the interaction between the camera and the deformable object presents significant difficulties. The complexity arises from the relationship between the object's shape changes and the camera's position. Visual servoing, a technique that enables visual tracking of features through a camera mounted on a robot, offers a potential solution to this challenge. This technique has proven effective in tracking rigid objects. In this thesis we invest the field of non-rigid objects for visual servoing. It is then a question of defining strategies of corrections of movement of the robot according to these models. The study of stability and cases of singularities are also part of this thesis work. The technique thus developed will be used in robot-assisted surgery. It applies to the case of a robot carrying a laparoscope for monitoring a target organ.
This study shows that despite important evolutions in the field of visual servoing of rigid objects, the problem of vision of deformable objects still constitutes a major research challenge. In addition to the problems related to rigid manipulation, one must indeed take into account the complex interactions between the deformable bodies and the camera. This is an important challenge especially in the case of objects deforming in the three dimensions of space. This challenge is precisely the main objective of our work. We are particularly interested in the visualization of objects that can undergo large deformations, whereas the works of the literature usually consider small deformations.

State the research problem/Questions

Visual servoing for deformable objects is a crucial challenge in both robot-assisted and computer-assisted surgery applications. Because of its intrinsically flawed nature, the formal problem remains open to this day.

posed, and in light of the ambiguous paths that the task of assuring such objects may produce. The complexity of this problem's solution lies in the geometric and mechan-

ical characterization of deformable objects. The use of a monocular view to track and compensate for these deformations makes the task even more difficult. The goal of this thesis is to provide a general approach to visual service that can be used to any sorts of deformable objects. This thesis work provides an effective solution to this problem by proposing a nonparametric approach and employing an efficient visual control strategy to compensate for non-rigid object deformations visually. These compensations enforce camera positioning modifications in order to meet predetermined visual optimization criteria during the planning phase. The core principle of the proposed approach is to avoid making any assumptions regarding the geometry or the mechanical behavior of the object being deformed. Several experiments were conducted on various object geometries, object mechanics, and camera trajectory to validate the proposed visual preservation strategy.

The research problem and questions can be stated as follows:

1. What is the importance and effect of using visual operating methods on straight lines?

2. What strategy works better than a matrix interaction strategy that brings all the triangles together?

3. Why do we use image moments as visual features, and how does this strategy contribute to the overall approach?

4. Is there any existing object deformation parameters that can be included in visual control scheme and the nature of the deformation affect the control motion of camera?

5. Detailing in a more formal way how to pass from the visual measurement to the control quantities applied to the camera ?

6. What is the automation standard that adapts distortions and adjusts camera trajectory?

Objectives of the thesis

All of these studies indicate that these dietshas generally been examined on a case-by-case basis. Our main objective in this thesis is to find a method allowing an optical multiplication of deformable objects, Determine the paths to follow,the motion of the camera and/or the resulting distortion of the object captured from the starting composition (the state initial) to the final composition (the desired state). The approach we develop below can be divided into phases. First, we introduce a novel deformation control method that operates without the need for prior knowledge of the object's physical properties. This method estimates on line the Jacobian deformation matrix which links

the displacements of the robotic effector(s) in response to the resulting deformations. By utilizing this matrix, the control law can be computed, enabling the generation of the desired deformation. To evaluate the accuracy and stability of the method, we propose an evaluation based on visual markers and one not based on markers it is a question of defining the composition. The second phase consists of creating schematic paths that ensure throughout the process a visualization of the stability of the deformable object by considering the kinematic and dynamic constraints associated with the interaction between the camera and the object. Finally, we present the processing of the information obtained by the camera will make it possible to do this Formulate a control law to steer the robot based on changes in the object. Control aspects can be defined to synthesize vision-based control code. Experiment with a virtual organ outside the body (like the liver), and we applied a deformation to the upper segment, where the deformation of this part was induced through an imposed displacement of $22mm$. We also employed a triangular mesh representation of the visualized object, aiming to minimize the error during the process of any triangle before and after the elastic deformation on this basis where the camera is still in motion.

Content of the chapters

The manuscript consists of six chapters organized as follows:

Chapter 1: We begin by reviewing the state of the art in the modeling and visual servoing of deformable objects. Next, we explore the field of medical robotics, where tasks are performed within a 'closed-loop' control framework, utilizing visual data directly derived from medical imaging processes. Finally, we define the general problem of visual servoing for deformable objects.

Chapter 2: We compare three approaches to visual servoing across four lines, utilizing the interaction matrix for one of the lines. The three approaches involve instantiating the interaction matrix at the current point, at a reference point, and using an interaction matrix that serves as the center for the matrices of the first two approaches.

Chapter 3: We propose a visual servoing strategy for multiple triangles undergoing different piecewise rigid motions. Our approach involves first computing a correction for each triangle independently, using its associated interaction matrix. Subsequently, we apply a global correction by averaging the individual corrections to adjust the camera pose, aiming to achieve a view that closely matches the desired visual features.

Chapter 4: We propose a visual tracking approach for deformable objects, considering a pre-planned camera path and modeling the deformable object as a mesh made up of a series of triangles. A set of continuous weights are associated to the object's triangles. We formalize our approach as a least-squares minimization of the error between pre-planned views and current views at run time. Results on simulated deformations and trajectory are presented. The results shows the activeness of our approach when

considering minimization on relevant triangles of deformed objects.

Chapter 5: We define a reference trajectory for a camera tasked with observing a non-rigid object. The goal is to adapt this trajectory during execution by leveraging prior knowledge, limited to the most informative landmarks. Our method avoids the use of both parametric and non-parametric models of the object's deformation. Instead, we cast the problem as a tracking task within an optimal control framework. This framework employs visual servoing of features extracted from the deformable object, comparing their positions along the reference trajectory and during actual execution. The control objective is formalized as a weighted least-squares cost function, aiming to minimize the discrepancy between the reference features and those observed in real time. The weights are smooth, time-varying functions that quantify the relative significance of the visible features.

We end the manuscript by giving a general conclusion and suggest some perspectives for future research.

Thesis contributions

The primary contribution of this thesis is to have provided work on the study and synthesis of visual servoing on deformable objects, the main contributions relate to the following points

1. We introduce a camera-based motion planning strategy designed to compensate for deformations in non-rigid objects through purely rigid camera movements. This approach provides a formal framework to describe the interaction between the camera and the deformable object, modeled via a set of camera-triangle primitives, each corresponding to a specific geometric element of the object's mesh

2. Each pair of camera-triangle primitives is associated with a continuous weight function, ensuring a smooth variation of influence along the camera trajectory. This continuity guarantees stable and consistent tracking performance throughout the entire motion sequence.

3. We address the visual servoing and tracking problem within an optimal visual control framework that remains agnostic to any specific deformation model of the object. The weight functions previously introduced quantify the importance of each camera-triangle primitive, assigning greater influence to those most relevant to the tracking task. This formulation enables the derivation of an optimal rigid camera trajectory that effectively compensates for non-rigid deformations of the observed object.

4. Experimental validation demonstrates the robustness of the approach in the presence of noise in image detection, as well as variations in trajectories and types of deformations.

Chapter 1

Related work

1.1 Introduction

Vision control, or visual servoing, involves regulating the movements of a dynamic system by incorporating data from a vision sensor into a closed-loop control system. Initially, visual servoing techniques were developed to control manipulator-type robots from the image provided by a monocular camera. A well known classification allows to qualify the different types of visual servoing according to criteria which are: The configuration of the visual sensor in the robotic system, the type of control architecture used and the type of quantity to be controlled.

Visual servoing for non-rigid objects is an emerging area of research that presents significant unresolved challenges. Many existing methods incorporate continuous mechanical models within the interaction matrix to derive appropriate visual servoing commands. However, such approaches are heavily parameter-dependent and often demand meticulous calibration tailored to each specific type of deformable or elastic object.

1.2 The positioning of the visual sensor

The configuration of the on-board sensor: As the expression indicates, the visual sensor is mounted on the robot terminal. This configuration is called **"eye in hand"** in the literature. The purpose of the servo is usually to move the visual sensor to reach or follow for example a moving object [43] (see figure 1.1[4]).
The remote sensor configuration: For this **"eye to hand"** in this configuration, the visual sensor is not mounted on the slave robot. (see figure 1.1 [4]).

Because of the limited places within the human body and the possibility of occlusions, surgery frequently necessitates a "eye-in-hand" viewing system. This system allows the

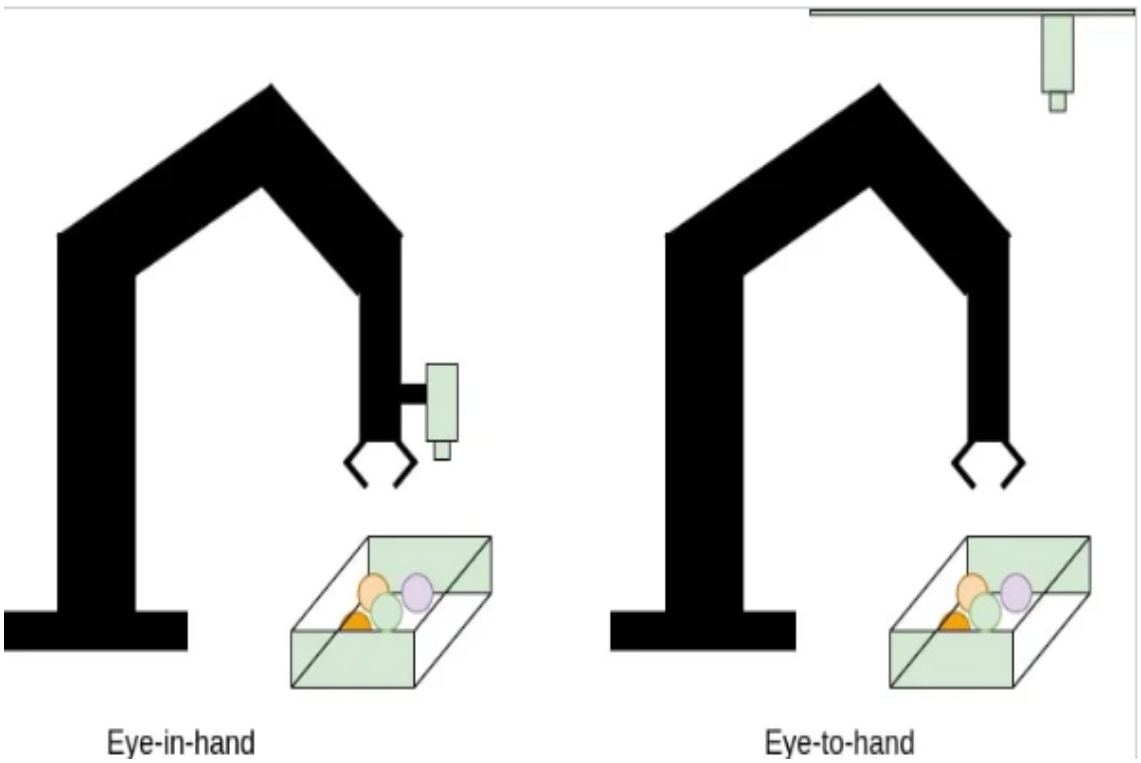

Figure 1.1: Camera-mounting-based visual servoing paradigms [4]

operator to view the surgical robot's camera directly from the control panel, improving procedure precision and reliability. However, it is crucial to note that several robot control schemes can be used depending on the needs of each procedure.

1.3 Controlled quantities

3D visual quantitiy : In 3D visual servoing, the objective is to control the robot's motion directly in Cartesian space. The controlled variable is defined as the pose of a frame rigidly attached to the robot's end-effector, expressed relative to a reference frame fixed to the object of interest. A significant limitation of this approach lies in the requirement for a 3D reconstruction step to estimate the pose, which introduces additional complexity and potential sources of error. Such reconstruction can be achieved, for instance, through triangulation using a calibrated stereoscopic vision system, or via pose estimation techniques in the case of a monocular camera. [43]. Pose reconstruction techniques use primitives extracted from the image, these methods generally rely on prior knowledge of a model of the observed scene. Unfortunately regardless of the pose reconstruction method used, attitude estimation is very sensitive to measurement noise and calibration errors of the visual sensor. As a result, after convergence of the result is a bias

between the robot position and the object of interest when the steady state is reached. This drawback is the major problem of 3D visual servoing [32] [79], if current methods do not allow for sufficient precision, it is prudent to refrain from using 3D primitives in this work. It is always crucial to prioritize the safety and quality of care for patients.

2D visual quantitiy: In 2D visual servoing, the quantity to be served is expressed in the form of visual primitives in the image. For this type of servo, we also speak of "image referenced servo"[43]. In contrast to a 3D visual servo, the robot is not controlled in Cartesian space, but directly in the image [71]. This form of visual quantity is considered in our work since it is similar to the vision and perception criteria used in surgery.

1.4 Visual servoing on rigid objects

Visual servoing involves the use of computer vision data, captured by one or more cameras, to control the movement of a robot. Specifically, rigid visual servoing employs visual feedback to guide the robot's motion relative to a rigid object within the environment. This technique has been widely studied and successfully applied in numerous robotic applications.

Vision-based robot control refers to visual servoing. It is a way of guiding a robot's mobility by utilizing real-time feedback from visual sensors to execute tasks [18, 84]. By understanding the image's "vision" we are able to regulate the camera's movement. To solve these issues, several techniques have been proposed [84, 29].

Visual servo control is generally divided into two primary approaches: Position-Based Visual Servoing (PBVS) and Image-Based Visual Servoing (IBVS). PBVS depends on estimating the 3D pose of the object relative to the camera to compute the control law, while IBVS directly utilizes visual features extracted from the image to guide the robot's motion, thereby eliminating the need for explicit 3D reconstruction [73, 86]. In this work, we use the image based visual servoing and assumes a calibrated perspective camera with known intrinsics.

The interaction matrix characterizes the relationship between the motion of the camera and the variation of visual features observed on the object. In image space, this matrix can be determined by points, lines, or ellipses and moments [26, 5]. In this work, we compare the usage of points, lines and image moments as visual features for servoing on deformable objects. Janabi et al. [36] reduced a set of features, including holes, circles, and wedges, was selected due to their widespread occurrence in industrial components and their ease of robust extraction.

The works of Chaumette[16] and Tahri [84] utilized moments derived from image segmentation to determine the analytical form of the interaction matrix. In Cartesian space, this matrix can be determined from the pose and coordinates of 3D points. Molnar et al. [63]

proposed a marker-based visual servoing technique to automate camera positioning in minimally invasive surgery. This method adjusts the endoscopic camera's pose to maintain the surgical instrument within the field of view, without the requirement to keep it centered. Furthermore, the interaction matrix can be determined via a hybrid visual servoing approach that integrates image-based and position-based techniques. Iheb [77] developed a predictive control algorithms for image and position-based visual techniques, accounting for robot, camera, and measurement uncertainties.

1.5 Visual servoing on deformable objects

Robotic manipulation of non-rigid objects presents significant challenges due to the deformations that occur under applied forces, including stretching, torsion, and compression. This complexity is relevant to a wide range of potential applications in areas such as augmented reality, medical imaging, and robotic manipulation, involving the handling of diverse materials such as tissues, paper, rubber, viscous fluids, cables, and food products[70]. Human organs are good examples of non-rigid objects with a variety of shapes and deforming behaviours[48]. Hu et al.[34] presented a controller based on a deep neural network to govern the position and shape of deformable objects with unknown properties. Its multilayer architecture captures the objects' nonlinear behavior. Formulated in acceleration-resolved form, this approach ensures seamless integration with visual servoing systems. Jia et al. [37] introduced a histogram of oriented wrinkles (HOW) to characterize the shape variations of highly deformable objects. By applying Gabor filters and separating the high- and low-frequency components, the object's characteristics were computed. An off-line training step was employed to establish a correlation between these visual features and the end-effector's speed, enabling the precomputation of visual feedback. Hu et al. [35] developed a servoing algorithm to learn a nonlinear deformation function concurrently with the manipulation process. Gaussian Process Regression was employed to model and estimate the deformation parameters of the soft object.

Visual servoing with mechanical parameters

Hybrid visual servoing, which integrates mechanical parameters with visual data, is a control technique used to guide a robotic system's motion. This approach has been widely studied in the literature and applied to a broad range of robotic applications. Xu et al [88] proposed a novel robot control algorithm using position-based control and 3D reconstruction from stereo vision to achieve desired 3D shapes.

Makiyeh et al [50] proposed an innovative approach that leverages the analytical structure of the underlying system of equations. Additionally, they circumvented the use of nonlinear optimization procedures, which can be computationally expensive. Shetab-

Bushehri et al [81] proposed a shape servo scheme based on the As-Rigid-As-Possible (ARAP) method. This approach is both straightforward and efficient. Importantly, it does not require prior knowledge of the object's mechanical deformation parameters, nor does it depend on a Jacobian derived from data collected during the robot's movement over time. Xu et al [89] note that to overcome the limited field of view (FOV) of pCLE, a method involving image stream acquisition and mosaic assembly has been established. However, soft tissue deformation poses a significant challenge to consistently obtaining high-quality images. To address this, the authors proposed a position-based visual servoing scheme for generating high-precision, gap-free mosaics. This scheme has demonstrated reduced error in both circular and spiral trajectories compared to open-loop control, highlighting its potential for producing large, accurate mosaics in soft-tissue endomicroscopy. Aranda et al [9] proposed a model-based shape servoing scheme to solve the monocular visual shape servoing problem, which involves deforming a deformable object to a desired shape in 3D space by robots using 2D vision monocular. The scheme addresses the problem of under-constrained perception and under-activation of form control, and is validated by simulations and real-life experiments. Kennedy et al [40] aimed to develop combined haptic and visual servoing control methods to eliminate the need for mechanical stabilizers in Coronary Artery Bypass Graft (CABG) surgery. Their approach seeks to provide the surgeon with a stationary operating area using vision and haptic feedback. Experimental validation involved real-time tracking of deformations on a rubber membrane and the 3D position of a moving point on a surface. Navarro-Alarcon et al [67] proposed visually servoed deformation controller for unknown elastic objects using passivity-based framework with iterative estimation of deformation Jacobian matrix. Validates the theory through experimental study. Lagneau et al [49] presented the ADVISEd method, which employs online estimation of the deformation Jacobian for model-free visual servoing of soft objects. This approach demonstrates robustness to observation noise.

Ott et al [69] introduced a motorization approach for flexible endoscopes to improve stabilization on areas of interest and simplify manipulation during surgeries. Liu et al [68] presented a vision-based deformation control method for robots interacting with unknown elastic objects, which used a dynamic state feedback velocity control law and avoid pattern identification steps by exploiting visual information in real time to estimate the Jacobian deformation matrix. Stability is assured despite inaccurate estimates, and the approach is validated by experiments with several deformation tasks. In this work, no mechanical models are used to create visual servoing stains on deformable objects. they proposed putting in place weights that represent the significance of deformed regions. These weights assist to direct the correction so that the corrected view is as close to the reference view as feasible.

1.6 Tracking deformable objects

The tracking and manipulation of deformable objects is much more complex than that of rigid objects because of the deformations these objects suffer when external forces are applied to the [47]. Current industrial robots lack the ability to control object deformation due to the need for both material property knowledge and real-time deformation monitoring, which are not currently integrated [49]. Tracking the geometry of deformable objects, like ropes and cloth, is particularly challenging due to their continuous characteristics, which involve an infinite set of degrees of freedom. This problem has been investigated across several fields, including computer graphics, computer vision, surgical simulation, and robotics [24]. Deformation tracking has been the focus of numerous studies across various research domains [92, 21, 82]. Researchers are striving to advance tracking techniques from simple rigid objects observed with a static camera to complex non-rigid object tracking under dynamic conditions involving a moving camera. Chen et al.[20] introduced a two-step object tracking method designed to combine efficiency with precision. First, a kernel-based approach robustly locates the object under complex conditions, including camera motion. Subsequently, a contour-based method is applied to precisely follow the object's shape, refining the initial localization. Cao et al. [12] proposed an approach based on extension deformation for extended object tracking. They hypothesized that an object's extension would deform relative to a reference extension by shifting particular control points from the latter to those of the former. Where, given the reference extension, the attributes of an extension may be fully represented by the control points. In the work of Joo et al. [38] a generative body deformation model was introduced, capable of representing the motion of each principal body part. Motion capture is achieved by aligning a 3D template model to the observed measurements. Royer et al. [76] introduced a method for tracking deformable anatomical targets within 3D ultrasound volumes. Their approach estimates physiological motion-induced deformations by integrating an intensity-based technique with a physically-based model, which enhances robustness to image noise. A key advantage is that it operates without fiducial markers and functions in real time. Royer et al. [75] proposed a 3D tracking method that incorporates regularization via a statistical motion model derived from biomechanical simulations. A notable limitation of this approach is its reliance on the manual identification of specific points on the prostate surface in each ultrasound frame to guide the model. Kajihara et al. [39] developed a system for tracking a line trajectory drawn on a flexible object. The system's performance in terms of speed and accuracy was evaluated under various conditions of uncertainty using dynamic compensation techniques. In this work, we consider deformation tracking based on the trajectory of a reference camera seeing a deformable object. We base our tracking strategy on weight values that represent the relevance of various deformed regions. These weights guide trajectory adjustment without requiring knowledge of deformation mechanics. The proposed algorithm constructs

weight functions that quantify the local relevance of obstacles and govern the switching strategy between environmental landmarks.

1.7 Visual servoing in medical applications

A fundamental challenge in the manipulation of deformable objects lies in the continuous variation of their shape throughout the manipulation process[92]. To address this, various real-time tracking methods for deformable objects have been proposed in the literature [75, 76].

In intraoperative procedures such as biopsies, it is critical to continuously track deformable tissue motion to accurately guide needle positioning under image feedback[10]. These procedures are commonly performed under ultrasound (US) guidance, which offers real-time visualization of soft tissue deformations[74].

Krupa et al.[46] introduced a robotic control method for 3D steering of a beveled-tip flexible needle using a duty-cycling strategy, allowing control over three degrees of freedom. Similarly, Hong et al.[41] developed a robotic system for rigid needle guidance toward a moving target tracked via 2D ultrasound. In this setup, the flexible needle, modeled as a polynomial curve, is tracked using particle filtering during automated insertion, although it requires alignment with the ultrasound image plane. Bernardes et al.[11] combined Rapidly-exploring Random Tree (RRT) path planning with a duty-cycling control approach to steer a needle within a 2D plane, using closed-loop visual feedback from a camera. This approach was later extended to navigate 3D trajectories, modeled as sequences of 2D planar arcs. Additionally, Chatelain et al. proposed a technique for three-dimensional steering of a flexible needle, guided by real-time 3D ultrasound imaging.

Pourtaherian et al. [2] proposed a needle detection algorithm for 3D US volumes, which is solely based on a directionally sensitive Gabor transformation without employing any external tracking devices, modifications of the acquisition system or the needle itself. Chevrie and al [22] presented a method to steer a beveled-tip flexible needle towards a target embedded in moving tissue. Needle steering is performed using a needle insertion device attached to a robot arm. This method uses a mechanics-based interaction model that is updated online.

Nowadays, the application fields of the visual servoing systems are very wide. Mathiassen et al. [60] proposed a visual servoing method to move the ultrasound probe, using a robot to align the image plane of the probe with the needle. The method segments the needle and updates a set of visual features based on a model of the needle. A state machine is used to keep track of the alignment process, and different visual features are used to control the probe in the different states, Mura et al. [64]presented a vision-based haptic feedback system to assist in the movement of an endoscopic device during capsule endoscopy. The haptic guidance module includes a haptic device that allows the user

to control the capsule's movement along a pre-defined path. Additionally, the module transforms 3D maps and relative paths into a guiding virtual force. By measuring the current relative distance between the user's input and the boundaries of the maps, the haptic system determines whether the user is moving toward or away from the colonic walls, generating feedback forces to aid the operator during navigation. The user will also sense an attractive virtual feedback force toward the generated path that will help the user in the navigation. In the following, we present three examples of applying visual servoing techniques in medical applications.

The experimental setup in [6] consists of a toilet bowl model, surgical robotic arms, a needle driver for placing brachytherapy seeds, a laparoscopic ultrasound (US) probe, and electromagnetic trackers. The system employs ultrasound to guide the probe and track the needle's position and orientation. Furthermore, correlation data derived from speckle noise in the ultrasound images is used to compensate for organ motion using a 6-degree-of-freedom (6-DoF) robotic ultrasound system, illustrated in Figure 1.2.

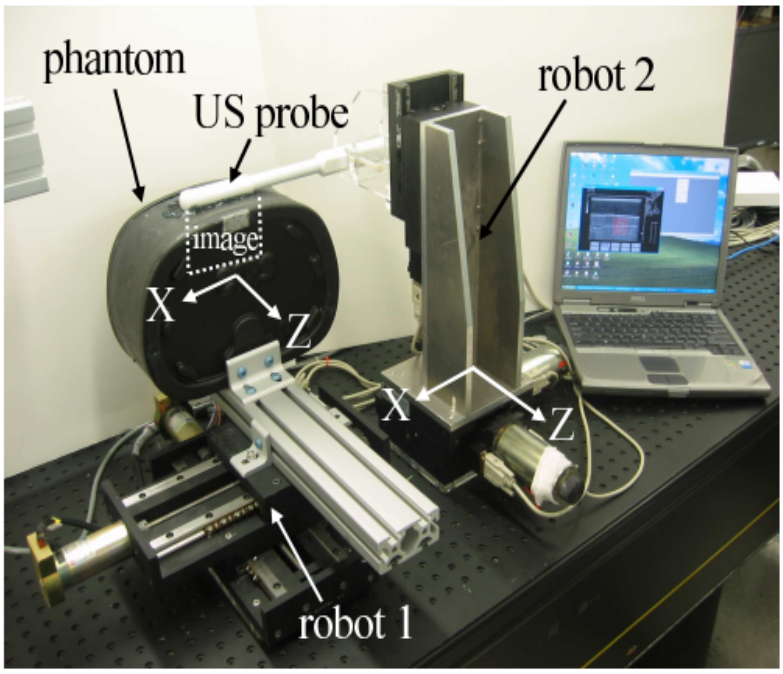

Figure 1.2: Collaborative surgical robot arms operating within a video-assisted, minimally invasive thoracoscopic surgery testbed [6].

In the second experimental setup in [85], they proposed to readjust the instrument-carrying robot by relation to the ultrasound probe by means of an algorithm using only observation of the points resulting from the intersection of the tool with the cutting plane a method show in figure 1.3.

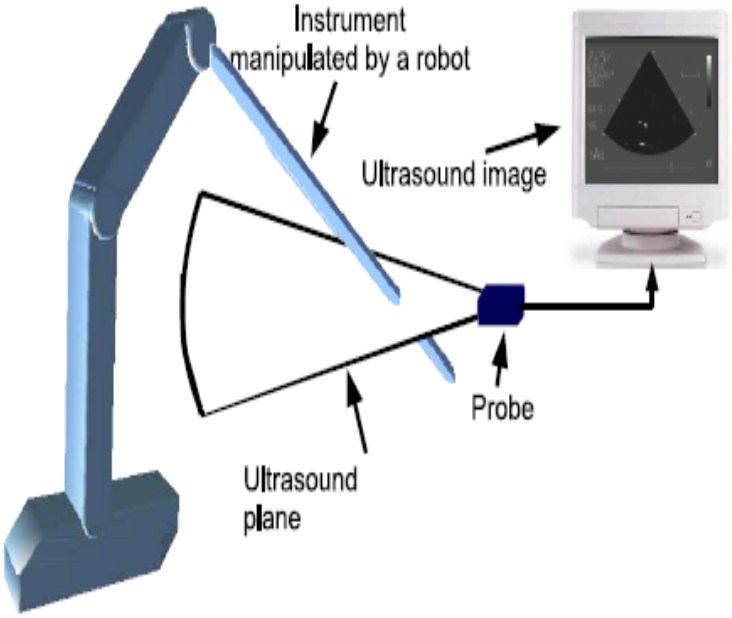

Figure 1.3: An ultrasound probe continuously monitors the instrument held by the robot, transmitting real-time imaging data to a computer for subsequent processing [85].

In [22] the experimental setup integrates a robotic arm equipped with a needle insertion device, 2D ultrasound imaging for tracking an artificial target, and an electromagnetic tracker to monitor the needle's tip position and orientation as shown in Figure 1.4. To compensate for tissue motion, the method uses real-time force feedback and an online-updated mechanics-based interaction model.

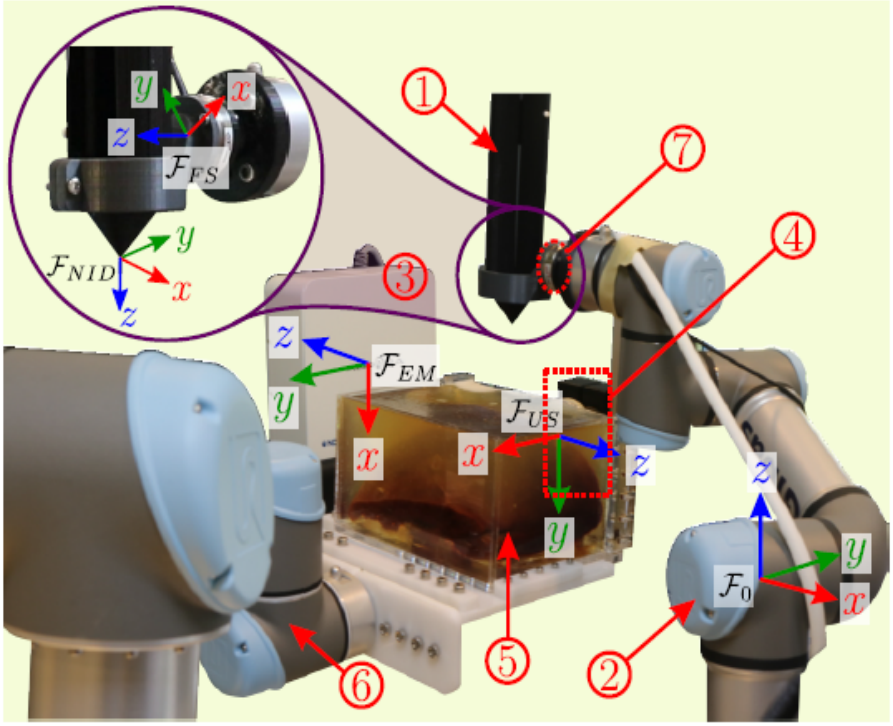

Figure 1.4: A beveled-tip needle is introduced into soft tissue using a dedicated needle insertion mechanism [22].

1.7.1 Robot or needle positioning

Ramirez et al. [72] introduce the Covariant Hamiltonian Optimization for Motion Planning (CHOMP) framework is utilized to plan motions involving deformable objects. Within this framework, an energy-based objective functional is formulated, incorporating a term that accounts for the potential energy of the elastic material. A study of the weighting parameters in this functional reveals that, although changes in these weights have limited impact on trajectory smoothness and collision cost, a specific configuration leads to the lowest energy outcome. In related work, Moll et al.[62] introduced a method for efficiently computing stable configurations of a wire under manipulation constraints, where the resulting shapes correspond to minimal-energy curves. By restricting the planner to these energy-optimal paths, the execution complexity is substantially reduced. This approach is used for planning the motions of deformable linear objects subject to multiple constraints.

1.7.2 Augmented Reality

Systems for augmented reality aim to combine real and computer-generated virtual objects in a single scene in a real environment. Unlike Virtual Reality, which immerses

the user completely in a virtual environment, Augmented Reality seeks to overlay virtual objects onto the real world, enhancing the user's perception of their surroundings. This technology provides or enables access to information that would otherwise be inaccessible, the synthetic objects enhance the user's perception of their environment. They enable, for instance, the display of actual hidden objects, the presentation of graphical or textual annotations, and the superposition of temperature-related sensor data. In this sense, the reality of augmented reality applications can be thought of as assisting the user in a real task to make it easier to complete. In general, any form of composition of real and virtual visual objects is considered to be part of the domain of Augmented Reality, as long as the two types of objects appear to coexist coherently within the same environment. Real and virtual environments are frequently considered to be 3D de-noised and 3D spatial coherence is sought. Taking into account liver deformations and tissue heterogeneity,Haouchine et al [33] introduced an augmented reality framework that employs a real-time biomechanical model to compute a volumetric displacement field from the partial 3D motion of the liver surface. This model facilitates the estimation of internal liver structures, such as blood vessels and tumors. In our method, we employed a decomposed triangle mesh to represent the liver. By analyzing the positions of a triangle before and after deformation, we applied robust control commands to the camera, allowing it to track and follow the deformation in real time. This work is based on canceling errors before and after the deformation, and the camera will move in different directions as the object deforms.

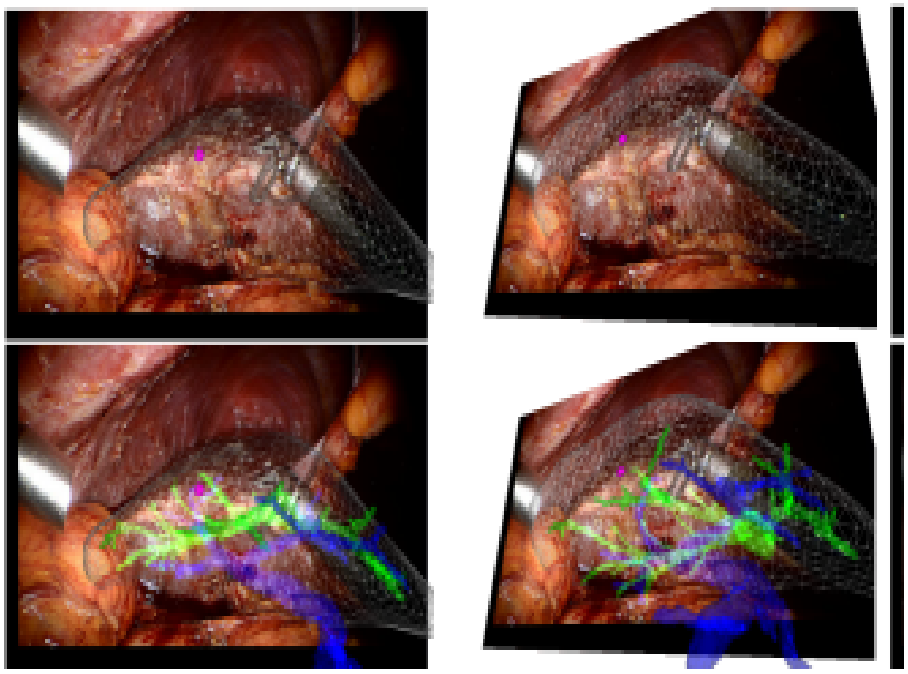

Figure 1.5: Images illustrating a real-time 3D biomechanical model superimposed on the human liver during minimally invasive hepatic surgery, demonstrating liver deformation caused by instrument interaction. The liver is represented with a wireframe structure, the tumor is indicated in purple, the hepatic vein is highlighted in blue, and the portal vein is accentuated in green [33].

Figure 1.5 represent a method for augmenting the vascular network and tumors in real-time during minimally invasive liver surgery. Internal structures, obtained from pre-operative CT scans, can be superimposed onto laparoscopic images to aid in surgical planning. In our approach, we focus on the liver, which was represented as triangular meshes. We track a triangle before and after deformation, applying robust control commands to the camera so it follows the deformation. This method is based on error cancellation before and after deformation, while simultaneously guiding the camera along a different trajectory to track the deformation of the object.

1.7.3 Visual servoing applied to augmented realit

Visual servoing shows great potential for 3D tracking and augmented reality applications. The challenge lies in controlling the movements of a virtual camera to minimize the difference between the image captured by the real camera and the view projected onto it by the object model. The real camera's pose is determined by the position of the virtual camera that minimizes this discrepancy, which is achieved when the 'virtual' visual servoing converges. The application of this approach in 3D tracking and augmented reality follows naturally from this principle [19]. In Marchand et al [58] presented a framework for the design of enhanced reality systems based on vision, taking into account

tracking strategies based on 3D models or tracking approaches without a 3D model. In each case, the method is based on the minimization of a cost function expressed in the image, and this minimization is accomplished using a visual service control law. Comport et al [25] presented formulation for position calculation, along with an illustration of its application in augmented reality. This approach involves adjusting the parameters of a virtual camera (including orientation and other key settings) while utilizing the visual servoing framework to capture the model back-projection through the extracted image data. Chang et al [14] presented visual servo assisted by AR. AR is being tested in the alignment and insertion of micro-assemblages based on 3D models.

1.8 Conclusion

This chapter is dedicated to the review of topics related to analysis, synthesis, measurement of take quality, and bibliographic research in the field of robotic visual.
This bibliographic study demonstrates that, despite significant advances in the field of robotic vision of rigid objects, the problem of seeing deformable objects remains a major research challenge. In addition to issues related to rigid vision, it is necessary to consider complicated relationships between deformable bodies and visualisation. This is a significant difficulty, especially when an object deforms in all three dimensions of space. This problem is, in fact, the primary goal of our efforts. We are particularly interested in tasks involving the visualization of objects that can undergo large deformations, whereas the literature focuses on small deformations.
As we have previously stated, modelling the interaction between the camera and the deformable object is a major challenge in our application setting. The interaction's complexity stems from the relationship between object shape change and camera position.

Chapter 2

Points and lines as visual primitives for visual servoing

2.1 Introduction

The camera is a sensor capable of converting the light intensity it receives into an image composed of pixels. It is then possible to restore on an image the shape of an object on which the light is reflected, as the human eye does on the retina. A camera provides raw data in the form of digital values for each pixel. By applying dedicated image processing techniques, one can extract various geometric features such as points, lines, or ellipses as well as identify the characteristics or type of the observed object. However, defining the geometric representation of an object in an image is essential. We offer a comprehensive explanation of how to compute the interaction matrix for both a point and a line. The visual configuration is defined by the orientation of the projected line within the image plane, as well as its perpendicular distance from the origin of a frame centered in that plane. The three methods compared involve computing the interaction matrix at: (1) the current feature position, (2) the reference feature position, and (3) a midpoint matrix that represents an average between the first two configurations matrix [91].

2.2 Basic visual primitives

These primitives are used to create more complex shapes and images by combining and manipulating them in various ways. These techniques have broad applicability across various domains, such as computer-aided design, interactive entertainment (e.g., video games), and scientific data visualization.

2.2.1 Points as visual primitives

Visual servoing on points refers to a control strategy used in robotics where a robot manipulator is controlled based on the position of one or more specific points in its field of view. The camera mounted on the robotic system continuously acquires images of the environment and monitors the spatial positions of designated points of interest [84]. When a 3D vertex is supplied in the camera frame with the coordinates $(X_L, Y_L, Z_L)^\top$ its projection onto the camera plane, in terms of a 2D perspective, is expressed as:

$$x_L = X_L/Z_L, \tag{2.1}$$
$$y_L = Y_L/Z_L. \tag{2.2}$$

Where $im = (x_L, y_L)^\top$ Disregarding the focal length and the principal point, the image coordinates of the 3D point are determined (in our study, we assume a calibrated camera with known intrinsic parameters). The time derivative of the aforementioned projection is expressed as:

$$\dot{x_L} = \dot{X_L}/Z_L - \dot{Z_L}x_L/Z_L, \tag{2.3}$$
$$\dot{y_L} = \dot{Y_L}/Z_L - \dot{Z_L}y_L/Z_L. \tag{2.4}$$

Let's assume that the cameras have translation and rotation velocities of $v_L = (v_x, v_y, v_z)^\top$ and $\omega_L = (\omega_x, \omega_y, \omega_z)^\top$ respectively, such that the total vector of camera velocities is expressed as $\dot{c_L} = (v_L^\top, \omega_L^\top)^\top$. The following formula from the classic work of can be used to calculate the relationship between the 3D point's velocity and the camera's speeds [17]:

$$\dot{X_L} = -v_x - \omega_y Z_L + \omega_z Y_L, \tag{2.5}$$
$$\dot{Y_L} = -v_y - \omega_z X_L + \omega_x Z_L, \tag{2.6}$$
$$\dot{Z_L} = -v_z - \omega_x Y_L + \omega_y X_L. \tag{2.7}$$

Replacing the equations eq.((2.5),(2.6),(2.7)) in eq.((2.3),(2.4)) , we obtain

$$\dot{x_L} = -v_x/Z_L + x_L v_z/Z_L + x_L y_L \omega_x - (1 + x_L^2)\omega_y + y_L \omega_z, \tag{2.8}$$
$$\dot{y_L} = -v_y/Z_L + y_L v_z/Z_L + (1 + y_L^2)\omega_x - x_L y_L \omega_y - x_L \omega_z. \tag{2.9}$$

The aforementioned system may be rewritten as

$$(\dot{x_L}, \dot{y_L})^\top = L(x_L, y_L, Z_L)\dot{c_L}. \tag{2.10}$$

$$L(t) = \begin{bmatrix} \frac{1}{Z(t)} & 0 & \frac{-y(t)}{Z(t)} & -x(t)y(t) & 1+x(t)^2 & -y(t) \\ 0 & \frac{1}{Z(t)} & \frac{-y(t)}{Z(t)} & -(1+y(t)^2) & x(t)y(t) & x(t) \end{bmatrix} \tag{2.11}$$

The interaction matrix of a point is a 2×6 matrix that describes how the coordinates of an image of a point change in response to changes in its 3D location in space.
The interaction matrix of a point is widely used in visual servoing work because it allows for the calculation of the derivative of an imaged point in relation to camera movements, which is essential for the design of visual controllers for robots.

2.2.2 Straight lines as visual primitives

A straight line in an image can be represented as an infinite-length segment (see Figure 2.1). Its configuration is commonly defined using the polar coordinates (ρ, Θ). Where ρ denotes the orthogonal distance from the origin to the line, and Θ indicates the orientation of the normal to the line. In three-dimensional space, a straight line can be interpreted as the intersection of two planes, denoted h_1 and h_2 [91].

3D primitive : $H(\mathbf{X})$ with $\mathbf{X} = (X, Y, Z)$

$$H(\mathbf{X}) = \begin{cases} h_1 = A_1X + B_1Y + C_1Z + D_1 \\ h_2 = A_2X + B_2Y + C_2Z + D_2 \end{cases} \tag{2.12}$$

Where A_i, B_i, C_i and D_i with $i = 1, 2$, are the parameters defining each plane are denoted accordingly, and the intersection of these two planes yields the following equation:

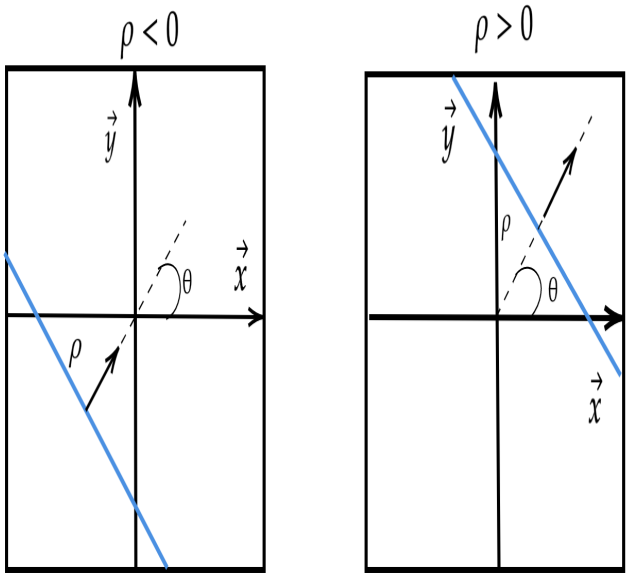

Figure 2.1: Reprsentation of a line with (ρ, Θ).

The relationship that connects the depth Z of a point lying on the line to its corresponding position in the image plane is expressed as follows:

$$\frac{1}{Z} = Ax + By + C \tag{2.13}$$

We consider a particular case in which the first plane passes through the origin. In this situation, the parameter

$$D_1 = 0$$

and consequently, the equation of the first plane h_1 becomes:

$$H(\mathbf{X}) = \begin{cases} h_1 = A_1X + B_1Y + C_1Z \\ h_2 = A_2X + B_2Y + C_2Z + D_2 \end{cases} \tag{2.14}$$

The equation of a straight line in a 2D image is given by the following equation:

$$Ax + By + C = 0 \tag{2.15}$$

We define our equation in terms of the following parameters: (ρ, Θ).

$$im(\mathbf{x}, p) = xcos\Theta + ysin\Theta - \rho \tag{2.16}$$

2D primitive: $im(\mathbf{x}, p)$.
Minimal parameterization $p = (\rho, \Theta)$

$$\Theta = arctan(B/A)$$

$$\rho = \frac{C}{\sqrt{A^2 + B^2}}$$

$im(\mathbf{x}, p)$ is seen in the camera frame before cameras intrinsics are applied to the camera plane and p it's parametrization with (ρ, Θ).

$$x = f(y, \rho, \Theta)$$

Let us recall that we use the (ρ, Θ) representation for a line defined by:

$$x = \frac{\rho}{\cos\Theta} - y\tan\Theta \tag{2.17}$$

The temporal derivative of 2.16

$$\dot{im}(\mathbf{x}, p) = 0 \tag{2.18}$$

we derive with respect to (ρ, Θ)

$$\frac{\partial im}{\partial \mathbf{x}}(\mathbf{x}, p) + \frac{\partial im}{\partial p}(\mathbf{x}, p) = 0 \tag{2.19}$$

By differentiating Equation (2.16), which is based on the assumption that the projection of a line remains linear in the image plane regardless of the camera's motion, we obtain:

$$-\dot{\rho} + (x\sin\Theta - ycos\Theta)\dot{\Theta} + \dot{x}\cos\Theta + \dot{y}sin\Theta = 0 \tag{2.20}$$

$$-\frac{\dot{\Theta}}{cos\Theta}y + \dot{\rho} + \rho tan\Theta\dot{\Theta} = yK_1c + K_2 * c \tag{2.21}$$

Where

$$\begin{pmatrix} K_1 \\ K_2 \end{pmatrix} = \begin{bmatrix} -\lambda_1 cos\Theta & \lambda_2 sin\Theta & \lambda_1\rho & -\rho & -\rho tan\Theta & -\dfrac{1}{cos\Theta} \\ -\lambda_2 cos\Theta & \lambda_2 sin\Theta & \lambda_2\rho & -sin\Theta & \dfrac{cos\Theta + \rho^2}{cos\Theta} & \rho tan\Theta \end{bmatrix} \tag{2.22}$$

$$\lambda_1 = Atan\Theta + B \tag{2.23}$$

$$\lambda_2 = \frac{A\rho}{cos\Theta} + C \tag{2.24}$$

$$\tag{2.25}$$

$$\dot{\Theta} = K_1 cos\Theta c \tag{2.26}$$
$$\dot{\rho} = (K_1 + K_2 \rho \sin\Theta) c \tag{2.27}$$
$$\tag{2.28}$$

The interaction matrix corresponding to a 2D line primitive, expressed in terms of its visual features, is given by the following matrix:

$$L_s^T = \begin{pmatrix} L_\rho^T & L_\Theta^T \end{pmatrix} = \begin{bmatrix} -\lambda_\rho cos\Theta & -\lambda_\Theta cos\Theta \\ \lambda_\rho sin\Theta & \lambda_\Theta sin\Theta \\ \lambda_\rho \rho & -\lambda_\Theta \rho \\ (1-\rho) sin\Theta & -\rho \cos\Theta \\ -(1+\rho^2) cos\Theta & -\rho \sin\Theta \\ 0 & 1 \end{bmatrix} \tag{2.29}$$

Where

$$\lambda_\rho = -A\cos\Theta + B\rho\sin\Theta + C \tag{2.30}$$
$$\lambda_\Theta = Bcos\Theta - Asin\Theta \tag{2.31}$$

2.3 Servoing loop for points and lines

We now present the process of constructing a control law based on 2D visual information. To this end, we consider the vector im, which represents the set of visual measurements extracted from the image. In our case, these measurements correspond to the vector $im = (\rho,\, \Theta)^T$. We aim to minimize the visual error, which is defined by the following expression:

$$e_l = im - im_0 \tag{2.32}$$

$im$is the current measure and im_0 is the reference measure. We need to determine the relationship between the speed of the visual cues $\dot{im}$ and the velocity of the camera c_l. This final equation summarizes the control strategy implemented in the visual servoing process.

$$\dot{e}_l = L_s c_l \tag{2.33}$$

using the equation (2.33) we will eliminate the error by introducing a proportional gain λ

$$c_l = -\lambda L_s^+ e_l \tag{2.34}$$

where L_s^+ is the pseudo-inverse of L_s. Continuing with the development, we obtain

$$c_l = -\lambda L_s^+ (im - im_0) \tag{2.35}$$

We complete the servoing loop by utilizing equation (2.33), assuming that the execution is performed perfectly c_l

$$\dot{e}_l = -\lambda L_s L_s^+ e_l \tag{2.36}$$

This will enable the definition of the servoing scheme representation for a polygon with four edges, treating each edge as a separate line.

$$\begin{pmatrix} \dot{\Theta}_1 \\ \dot{\rho}_1 \\ \dot{\Theta}_2 \\ \dot{\rho}_2 \\ \dot{\Theta}_3 \\ \dot{\rho}_3 \\ \dot{\Theta}_4 \\ \dot{\rho}_4 \end{pmatrix} = \begin{pmatrix} L_{\Theta_1} \\ L_{\rho_1} \\ L_{\Theta_2} \\ L_{\rho_2} \\ L_{\Theta_3} \\ L_{\rho_3} \\ L_{\Theta_4} \\ L_{\rho_4} \end{pmatrix} c_l = L_s c_l \tag{2.37}$$

In this case, the use of four lines is justified as it yields eight equations, which are sufficient to solve for the six unknowns associated with the camera's velocity.
The L_s matrix can be computed through various methods, including

1. L_s At the current point im
2. L_{s^*} at the reference point im_0
3. $\frac{L_s + L_{s^*}}{2}$ as the center between the first two.

The next section presents a comparison of these three different approaches, focusing on their convergence properties[91].

2.4 Results on simulated data

This section displays some illustrative instances of our simulation-based verification tests employing two types of objects: rigid and non-rigid polygons. On both rigid and non-rigid deformations, we conducted tests. We evaluated our approach and we employed a perspective camera with a 1500 pixel we will also consider the focal length to assess the robustness of our method. We used Matlab2015a and an Intel Core i3 laptop with 4 Go of RAM to do our simulations.

2.5 Results on rigid polygon

The visual servoing method is evaluated using a free camera (with 6 degrees of freedom) and four lines. We compare three different approaches for visual servoing on lines:

1. One approach utilizes the L_s interaction matrix evaluated at the current state.

2. Another approach employs the interaction matrix L_{s^*} which is taken at the reference state.

3. a strategy that makes advantage of the preceding interaction matrices' center, the interaction matrix $\frac{L_s+L_{s^*}}{2}$. Initially, the camera is aligned with the world marker in the reference configuration, after which it undergoes a rotational transformation during the operation 5 degrees around its optical axis Z in the initialization state. The camera's focal length is $f = 1500$. footnote The lengths are measured in millimeters, while the angles are measured in degrees. We use the lines from the plane $X+Y+Z = 20$. The following points are covered: Line 1 ($[3, 3, 14]; [4, 6, 10]$), Line 2 ($[4, 6, 10]; [-2, 2, 20]$) Lines 3 and 4 are ($[-2, 2, 20]; [-2, -2, 24]$) and ($[-2, -2, 24]; [3, 3, 14]$) respectively. We use $\lambda = 0.001$ for the servoing and set a requirement that the servoing must end when the average change in visual cues is less than 0.01. The figures 2.3, 2.4, 2.5, 2.5 and 2.7 display the findings. The four lines' images in the initial state relative to their images in the reference state are depicted in the figure 2.2. We also ran a similar set-up on that uses the interaction matrix L_{s^*} taken at the reference state and the obtained control converges as well as can be seen in figure 2.8, 2.9, 2.10 and 2.11, 2.12, According to the findings of the studies, the interaction matrix used in the $\frac{L_s+L_{s^*}}{2}$ technique appears to provide the highest accuracy and state-of-reference attainment in the figure2.13. After $2,000$ iterations, we observe that the rate of convergence is modest. Convergence is impossible at larger λ values.The figures 2.14, 2.15, 2.16 and 2.17 display the results. Furthermore, it has been noted that when the initial configuration deviates significantly from the reference exceeding 10 degrees in rotation or 300 mm in translation the system fails to converge back to the desired state [91].

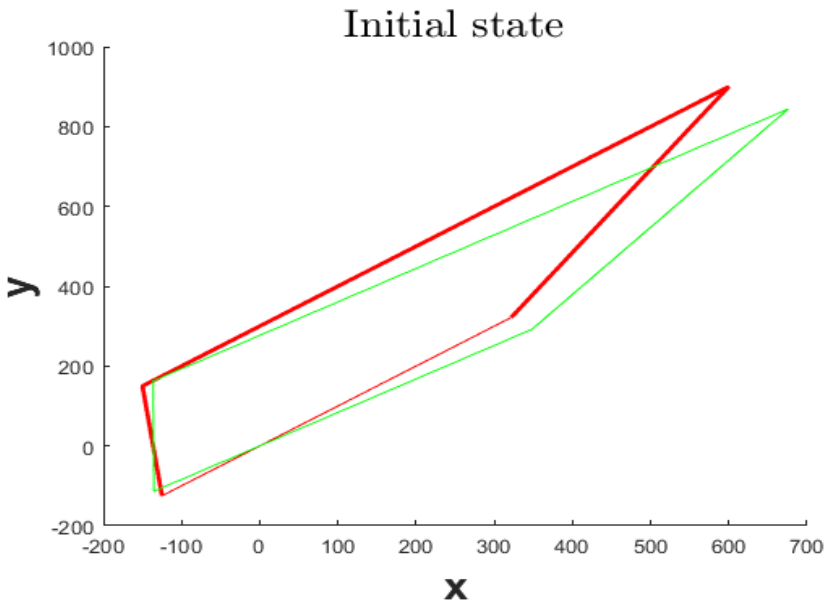

Figure 2.2: Initial State: The reference lines are shown in red, while the lines observed in the initial state are in green, depicted on the XY image plane in 2D.

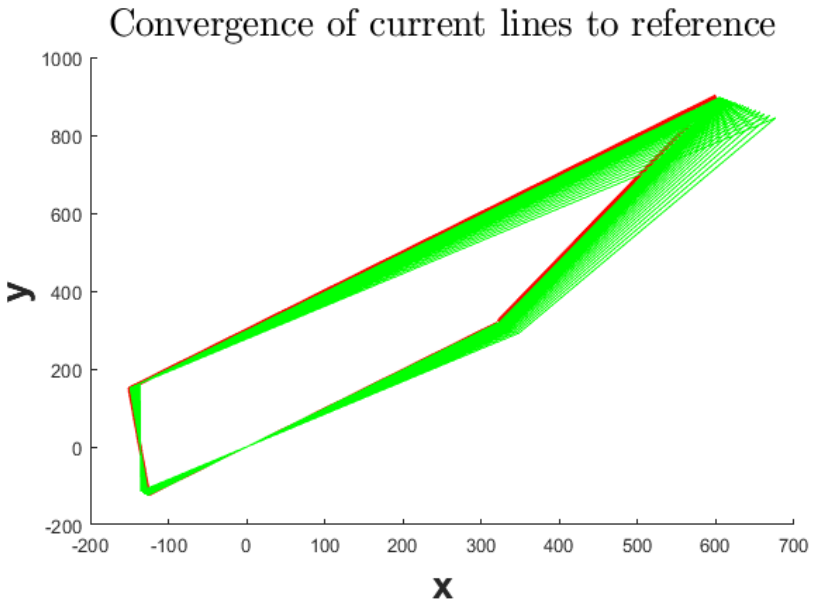

Figure 2.3: The test results are presented. Green lines represent the reference view, red lines indicate the corrected view, and all are displayed in the 2D XY image plane.

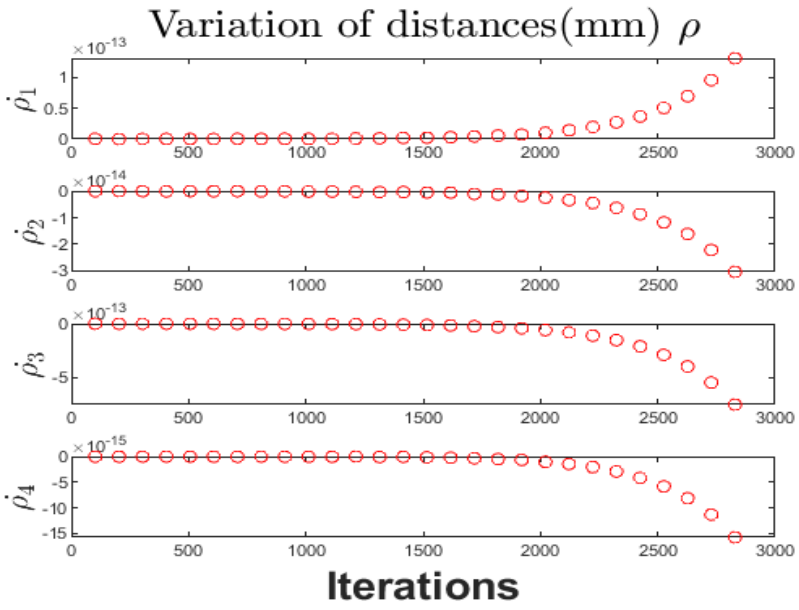

Figure 2.4: Results of variation of distances with L_s of four lines.

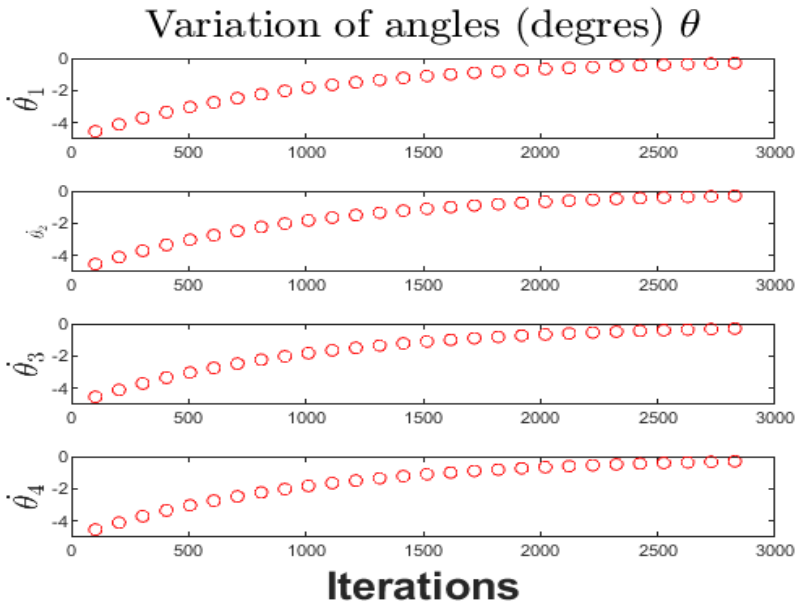

Figure 2.5: Results of variation angles with L_s of four lines.

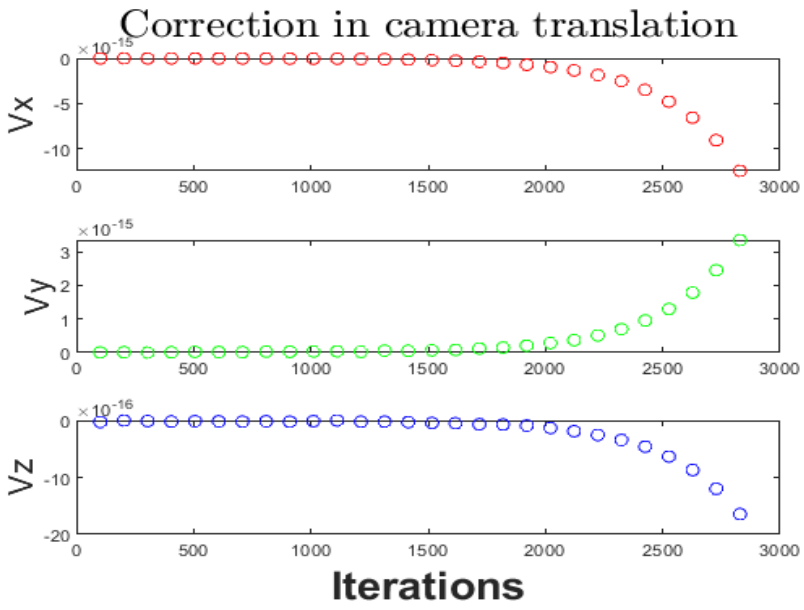

Figure 2.6: Results of the experiment with L_s represents camera speed in translation.

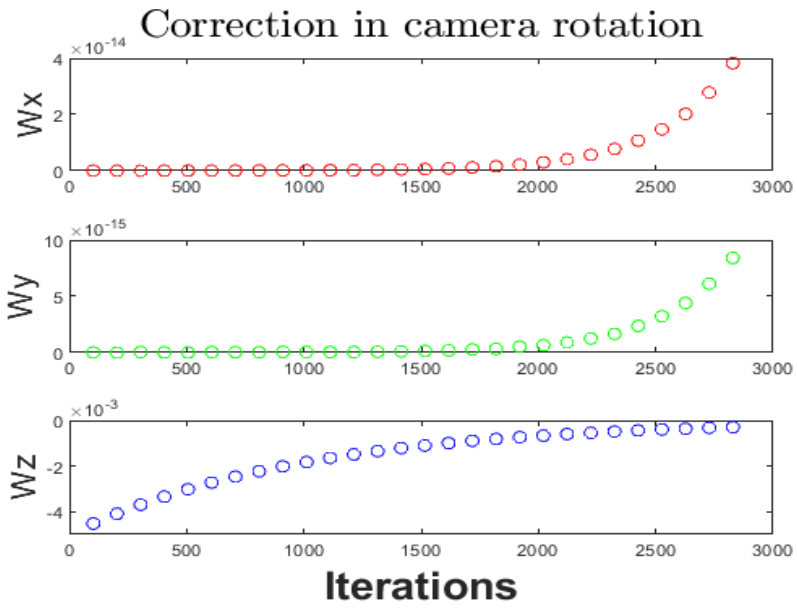

Figure 2.7: Results of the experiment with L_s represents camera speed in rotation.

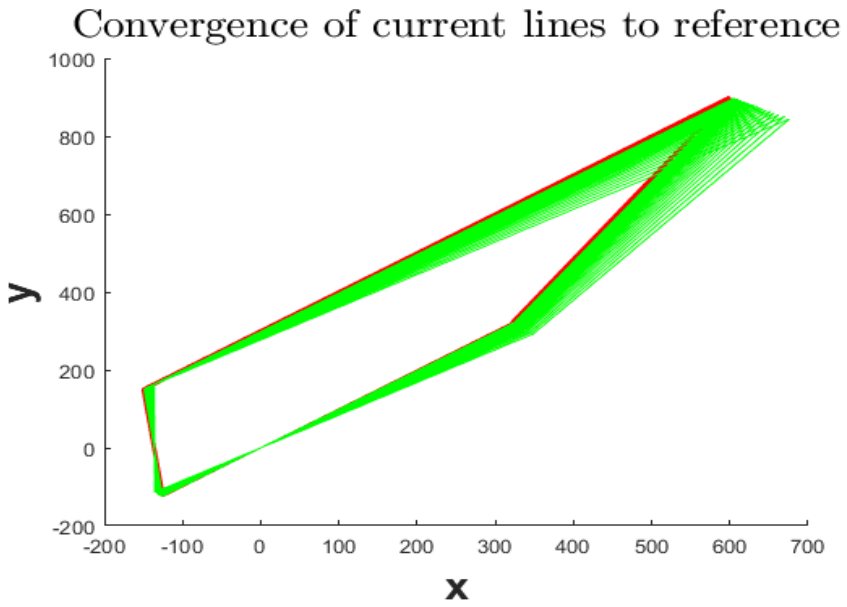

Figure 2.8: Test Results: The green lines represent the reference view, while the red lines indicate the corrected view, shown on the xy image plane in 2D.

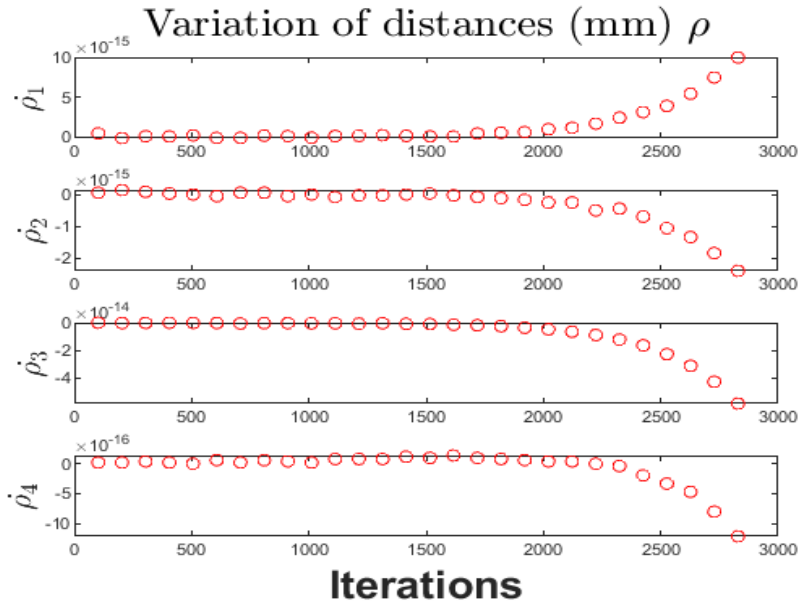

Figure 2.9: Results of variation of distances with L_{s^*} of four lines.

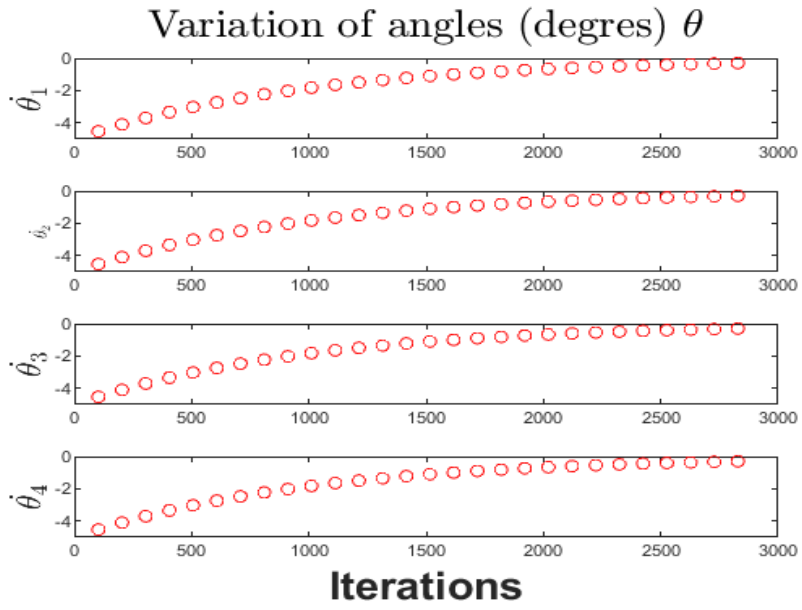

Figure 2.10: Results of variation angles with L_{s^*} of four lines.

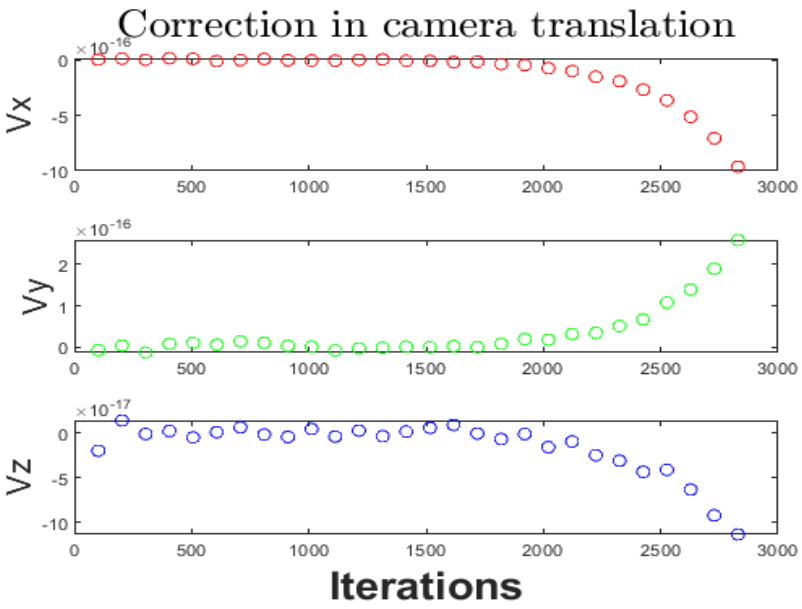

Figure 2.11: Results of the experiment with L_{s^*} represents camera speed in translation.

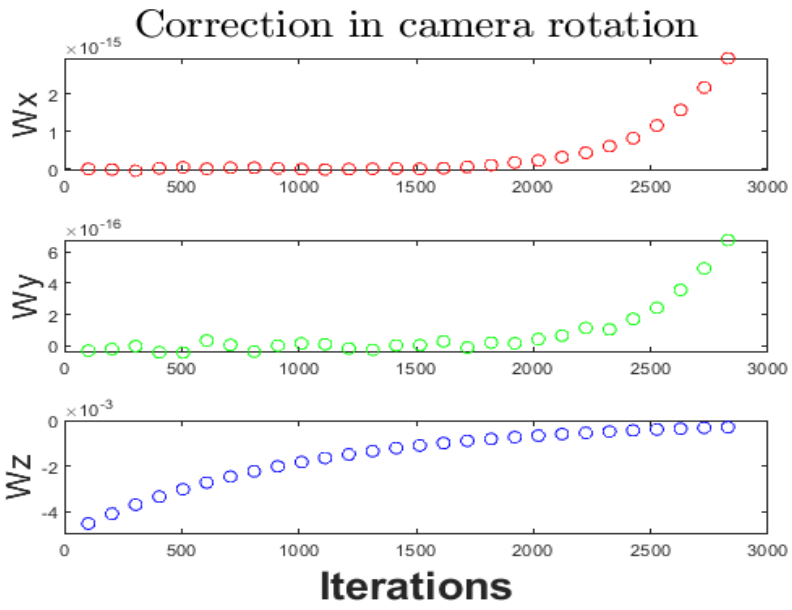

Figure 2.12: Results of the experiment with L_{s^*} represents camera speed in rotation.

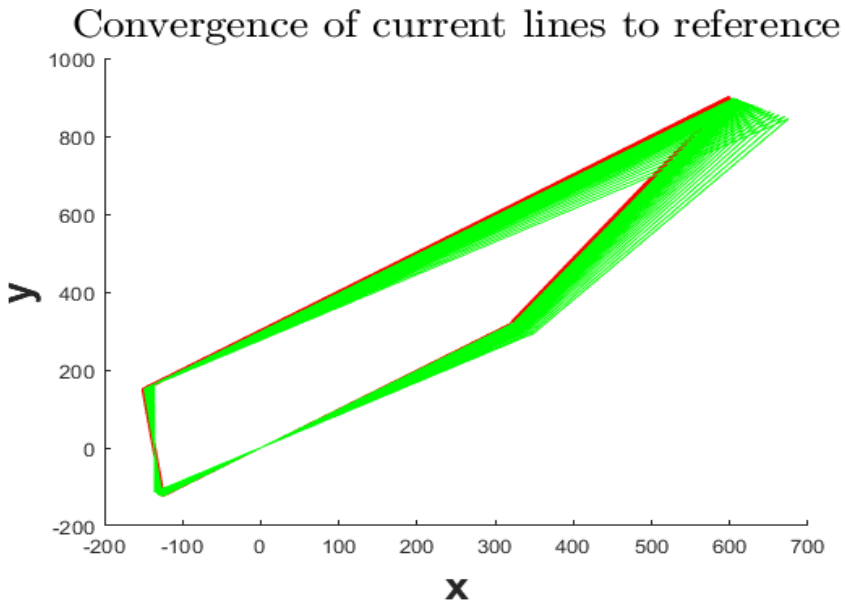

Figure 2.13: Green lines is the reference view. Red lines is the corrected view.

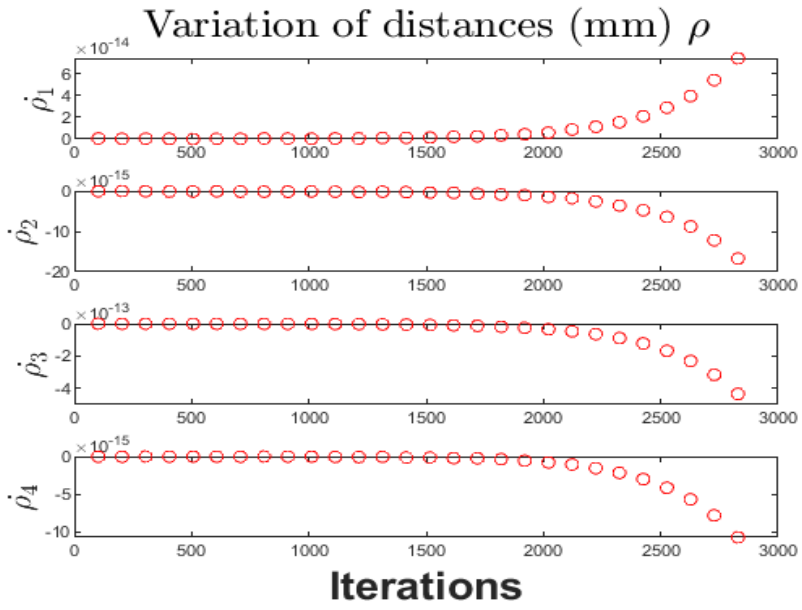

Figure 2.14: Results of variation of distances with $\frac{L+L^*}{2}$of four lines.

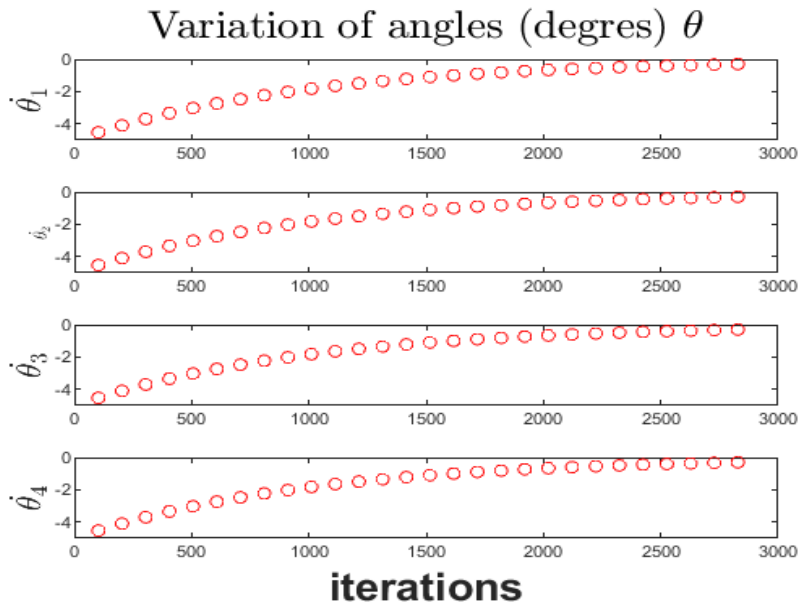

Figure 2.15: Results of variation angles with $\frac{L_s+L_{s^*}}{2}$ of four lines.

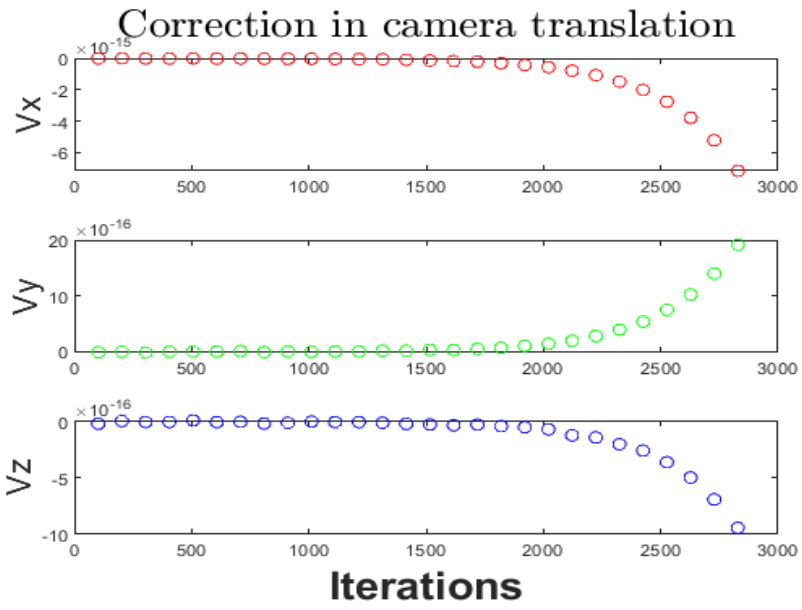

Figure 2.16: Results of the experiment with $\frac{L_s+L_{s^*}}{2}$ represents camera speed in translation.

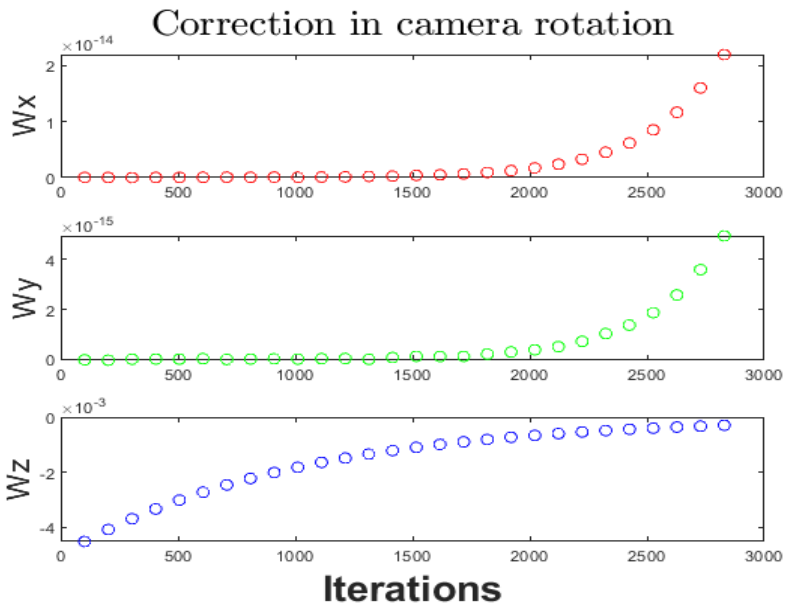

Figure 2.17: Results of the experiment with $\frac{L_s+L_{s^*}}{2}$ represents camera speed in rotation.

2.6 Results on non rigid polygon

The visual servoing method is evaluated using a freely moving camera with six degrees of freedom. Elastic deformation is applied to a configuration composed of four lines, with the deformation parameters specified by constants detailed in the appendix A.3 as it is shown in the figures 2.18, 2.19, 2.20, 2.21 and 2.22 . The deformation applied to the object

reaches a maximum of $250mm$ at its uppermost point. This deformation is observed using a camera equipped with a focal length of 1500, positioned fronto-parallel to the grid at a distance of $1.5meters$. The camera is manipulated in all six degrees of freedom, including linear displacements and rotational movements. v_x, v_y, v_z andw_x, w_y, w_z respectively). First we used the uniform correction method are constants defined in the appendix A.1 the camera is moved to the reference view. After $2,000$ iterations, we observe that the rate of convergence is modest. A higher prohibits convergence. Additionally, we've seen that starting states that are too distant from the reference state prevent us from going back there.

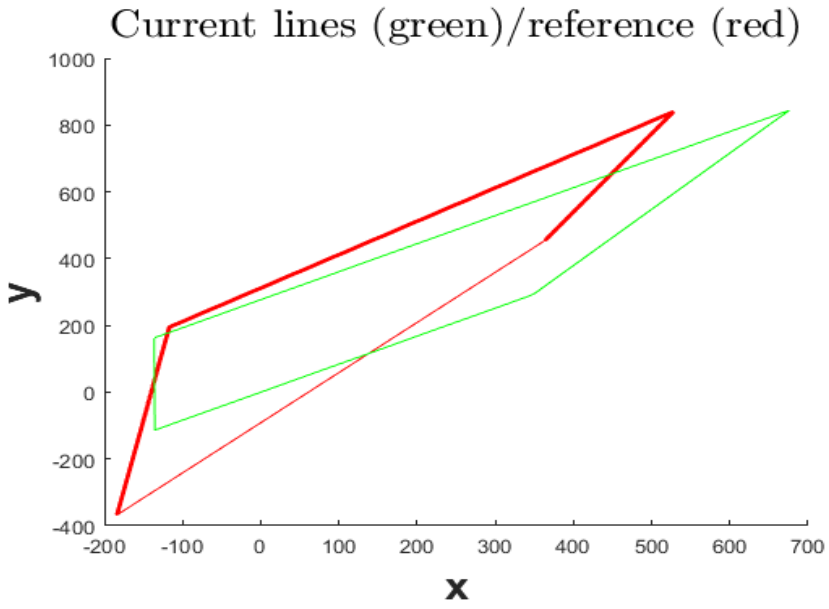

Figure 2.18: Initial configuration: The reference lines are depicted in red, while the green lines represent the observed lines in the initial state.

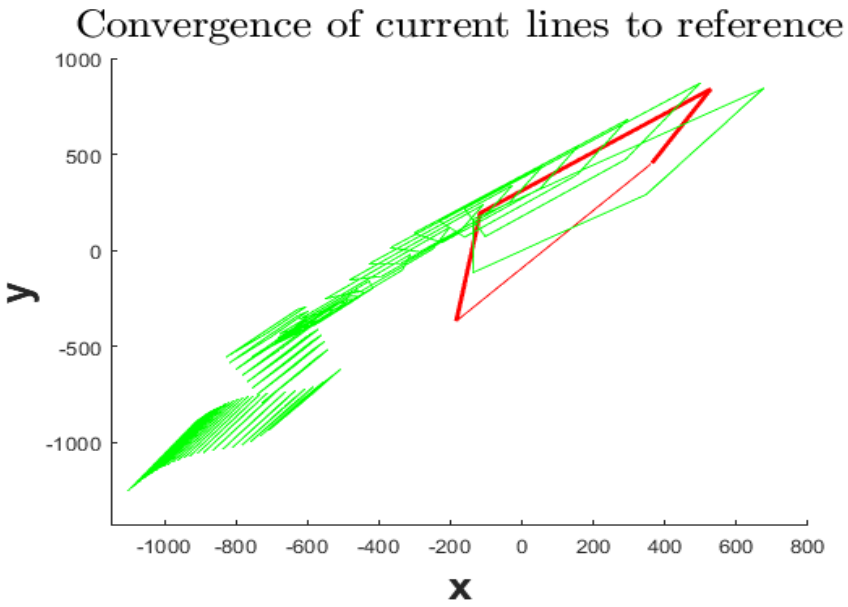

Figure 2.19: Test results: The green lines represent the four reference lines, while the red lines illustrate the corrected view obtained after applying the visual servoing approach.

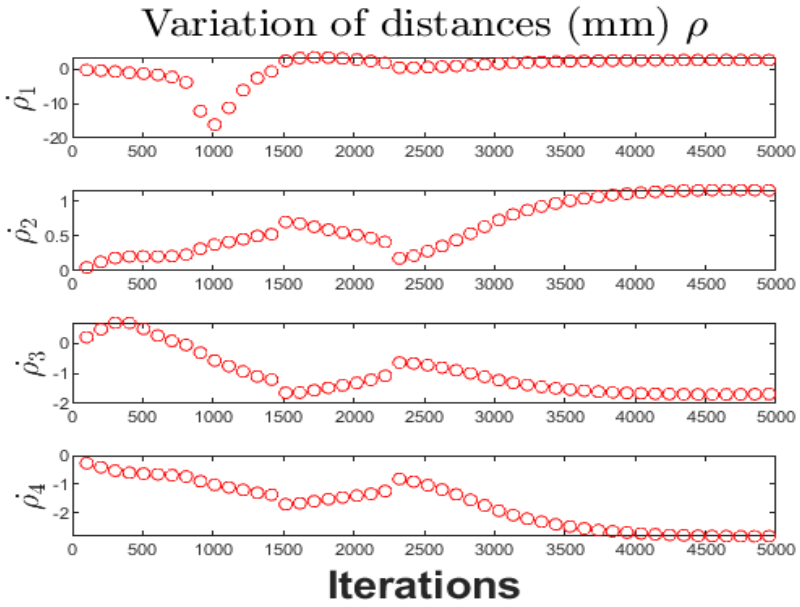

Figure 2.20: Results of distance variation with L_s for non-rigid deformation of four lines.

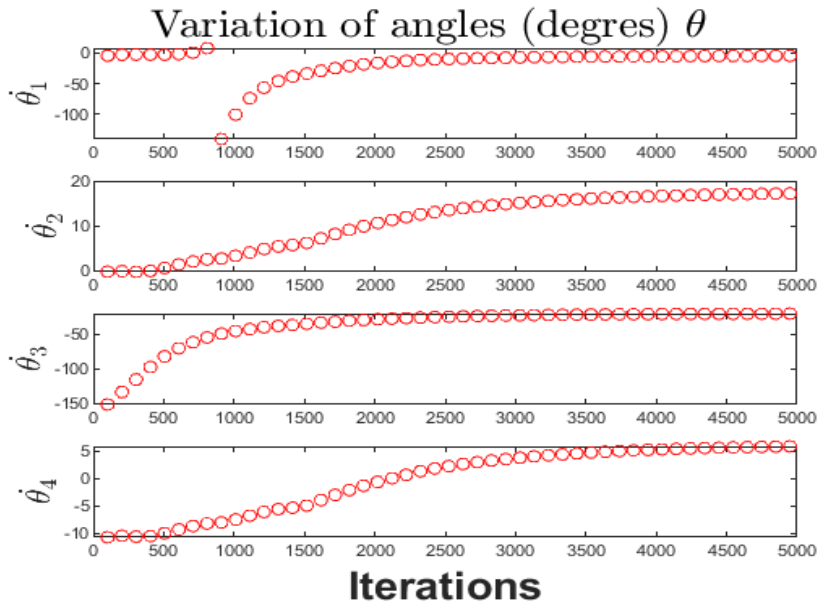

Figure 2.21: Results of angle variation with L_s for non-rigid deformation of four lines.

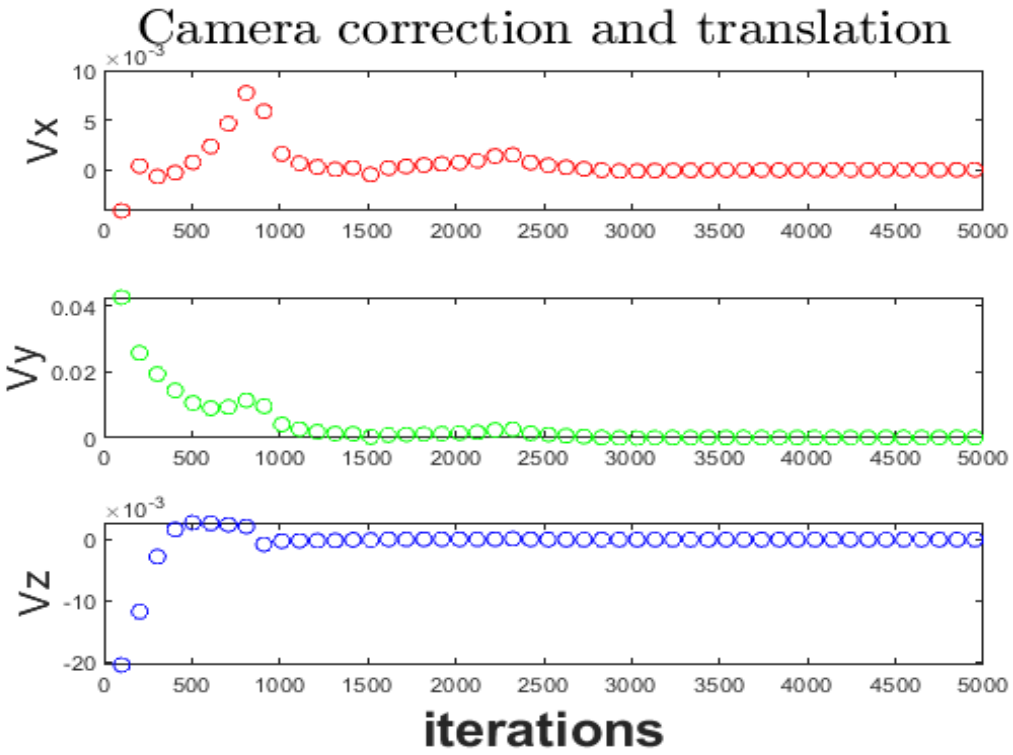

Figure 2.22: Results of the experiment with L_s non-rigid, representing camera translation speed.

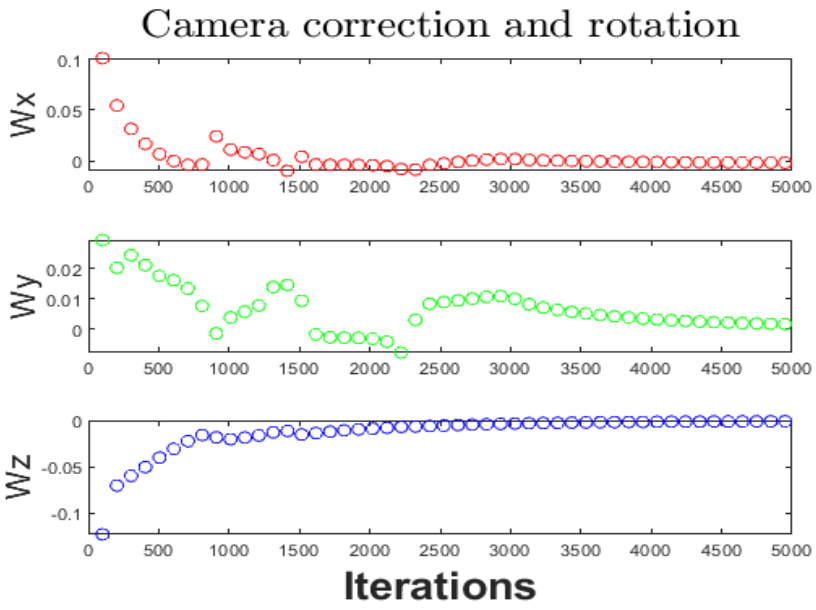

Figure 2.23: Experiment with L_s non-rigid, representing camera rotation speed

2.7 Conclusion

A usage of 2D lines as primitives in visual servoing on non-rigid polygon object is presented. The line's projection into the picture plane and its orthogonal separation from the reference frame origin make up the visual parameter. In the comparison of three visual servoing strategies, the interaction matrix is instantiated at distinct locations: the current position, the reference position, and a point midway between these two. Experimental results using four line features demonstrate that the method employing the interaction matrix computed at the midpoint between the current and reference states achieves superior performance in terms of accuracy and convergence speed. In contrast, using the interaction matrix at either the current or reference point individually yields less reliable results and is therefore not recommended for this scenario. We also observed that the 2D line primitive is not an appropriate feature to use in the case of non-rigid objects. In the next chapter we investigate if the image moment primitive is more reliable in this context.

Chapter 3

Image moment as feature for visual servoing

3.1 Introduction

Image moments are data that may be used to define a wide range of 2D forms since they are generic. Because they immediately reflect the area, barycentre, and primary orientation in the picture of the object of interest, low-order moments have an obvious geometric significance. These have the advantage of being less sensitive to measurement noise, as they are derived through integration, specifically through discrete summation over the observed area of the image. Image moments are commonly used in computer vision, especially in pattern recognition tasks, where translational and rotational invariant moment functions in the image have been constructed. They have also been presented in visual servoing, where 6 visual primitives built from a mixture of 2D moments have been offered in system control to execute the duty of camera placement [19] [83] [3].

3.2 Image moment-based visual servoing matrix computation

Let T the triangle visible in the image acquired by the camera at time t.
We denote $D(t)$ the area within the image corresponding to the projection of the 3D triangle, with $C(t)$ representing its contour edges at time $D(t)$. We analyze the triangle that has been segmented within the image. The moments m_{ij} of T in the image are defined by [90]:

$$m_{ij} = \int \int_{D(t)} x^i y^j dxdy \tag{3.1}$$

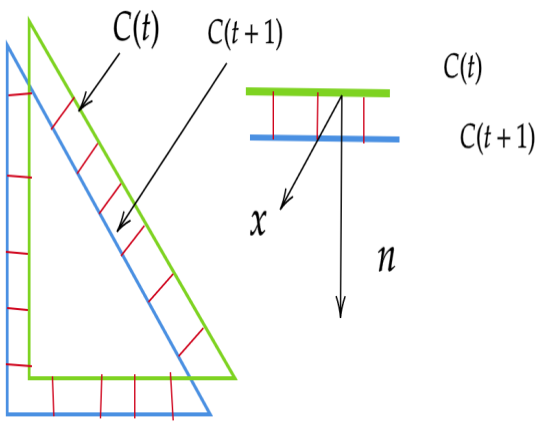

Figure 3.1: Image Plane Displacement of Triangular Features.

And its time derivative is

$$\dot{m}_{ij} = \oint_{C(t)} f(x,y)(\dot{x},\dot{y})^T \, \vec{n} \, dl \tag{3.2}$$

With $(\dot{x},\dot{y})^T$ the movement of the points along the edge contour at each time step $(x,y)^T \in C(t)$ and $f(x,y) = x^i y^j$. $\vec{n}$ the normal vector corresponds to the orientation perpendicular to the triangle edge (refer to Figure 3.1), and represents the integral computed along the edges of the triangle. The triangle is selected with the intention of extending the approach to triangular mesh representations of more intricate geometries. Our objective is to derive the analytical expression that captures the temporal variation $\dot{m}_{ij}$ of the moment m_{ij}. Observing that in the above equation, the only term that depends on time is the position of the contour points, we focus on analyzing how the motion of these points influences the overall variation of the integral t is $D(t)$ and by applying Green's theorem we have

$$\dot{m}_{ij} = \int \int_{D(t)} div[f(x,y)(\dot{x},\dot{y})^T] dxdy \tag{3.3}$$

Developing the divergence we find that

$$\dot{m}_{ij} = \int \int_{D(t)} [\frac{\partial f}{\partial x}\dot{x} + \frac{\partial f}{\partial y}\dot{y} + f(x,y)(\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y})] dxdy \tag{3.4}$$

This type of derivation, commonly applied in physics and mechanics, has also been pre-

viously utilized in the field of computer vision. Using the expression of f(x,y), we obtain:

$$\dot{m}_{ij} = \int\int_{D(t)} [ix^{i-1}y^j\dot{x} + jx^iy^{j-1}\dot{y} + x^iy^i(\frac{\partial\dot{x}}{\partial x} + \frac{\partial\dot{y}}{\partial y})]dxdy \tag{3.5}$$

The connection between the velocity of an image point and the movement of the camera is established as follows (3-translation and 3-rotation) [51] can be written as defined in the appendix A.

With $c = (v_x, v_y, v_z, \omega_x, \omega_y, \omega_z)^T \in R^6$, we considering the plane where the triangle lies is:

$$A_1X + B_1Y + C_1Z = D_1 \tag{3.6}$$

Therefore, the image projection can be expressed as a function of the depth of the 3D point Z as:

$$\frac{1}{Z} = Ax + By + C \tag{3.7}$$

with $A = A_1/D_1$, $B = B_1/D_1$ and $C = C_1/D_1$. Considering this, we substitute the inverse depth into the point interaction matrix, leading to the following expression for its variation:

$$\dot{x} = -v_x/(Ax + By + C) + xv_z/(Ax + By + C) + xy\omega_x - (1 + x^2)\omega_y + y\omega_z, \tag{3.8}$$

$$\dot{y} = -v_y/(Ax + By + C) + yv_z/(Ax + By + C) + (1 + y^2)\omega_x - xy\omega_y - x\omega_z. \tag{3.9}$$

$$\begin{pmatrix} \frac{\partial\dot{x}}{\partial x} \\ \frac{\partial\dot{y}}{\partial y} \end{pmatrix} = \begin{bmatrix} -A & 0 & 1/Z & y & -2x & 0 \\ 0 & -B & 1/Z & 2y & -x & 0 \end{bmatrix} * c \tag{3.10}$$

By differentiating the preceding equation with respect to the two-dimensional spatial coordinates, we obtain the following result x and y we find that:
For planar object $\frac{1}{Z} = Ax + By + C$ from which we deduce:

$$\begin{pmatrix} \frac{\partial\dot{x}}{\partial x} \\ \frac{\partial\dot{y}}{\partial y} \end{pmatrix} = \begin{bmatrix} -A & 0 & (Ax + By + C) & y & -2x & 0 \\ 0 & -B & (Ax + 2By + C) & 2y & -x & 0 \end{bmatrix} * c \tag{3.11}$$

Replacing equations (3.10) and (3.11) into (3.5), the interaction matrix, when expressed in terms of moments, can be formulated as follows

$$L_{m_{ij}} = \begin{bmatrix} m_{vx} & m_{vy} & m_{vz} & m_{wx} & m_{wy} & m_{wz} \end{bmatrix} \tag{3.12}$$

Where

$$\begin{cases} m_{vx} = -i(Am_{ij} + Bm_{i-1,j+1} + Cm_{i-1,j}) - Am_{ij} \\ m_{vy} - j(Am_{i+1,j-1} + Bm_{ij} + Cm_{i,j-1}) - Bm_{ij} \\ m_{vz} = (i+j+3)(Am_{i+1,j} + Bm_{i,j+1} + Cm_{ij}) - Cm_{ij} \\ m_{wx} = (i+j+3)m_{i,j+1} + jm_{i,j-1} \\ m_{wy} = -(i+j+3)m_{i+1,j} - im_{i-1,j} \\ m_{wz} = im_{i-1,j+1} - jm_{i+1,j-1} \end{cases} \tag{3.13}$$

$L_{m_{ij}}$ can be computed from moments of order less than $i+j+2$ and from plane parameters A, B and C for translational components.
The centered moments can be expressed as follows:

$$\mu_{ij} = \int\int_{D(t)} (x - x_g)^i (y - y_g)^j dxdy \tag{3.14}$$

$$L_{\mu_{ij}} = \begin{bmatrix} \mu_{vx} & \mu_{vy} & \mu_{vz} & \mu_{wx} & \mu_{wy} & \mu_{wz} \end{bmatrix} \tag{3.15}$$

The orientation of the triangle can then be described in terms of the second-order centered moments.

$$\theta = \frac{1}{2} arctan(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}) \tag{3.16}$$

The corresponding interaction matrix can then be expressed as follows

$$L_\theta = \begin{bmatrix} \alpha_{vx} & \alpha_{vy} & \alpha_{vz} & \alpha_{wx} & \alpha_{wy} & -1 \end{bmatrix} \tag{3.17}$$

$$\begin{cases} \alpha_{vx} = \theta A + \lambda B \\ \alpha_{vy} = -\gamma A - \theta B \\ \alpha_{vz} = -A\alpha_{wy} + B\alpha_{wx} \\ \alpha_{wx} = -\lambda x_g + \theta y_g + \delta \\ \alpha_{wy} = \theta x_g - \gamma y_g + \nu \end{cases} \tag{3.18}$$

With

$$\begin{cases} \theta = \mu_{11}(\mu_{20} + \mu_{02})/\Delta \\ \lambda = [2\mu_{11}^2 + \mu_{02}(\mu_{02} - \mu_{20})]/\Delta \\ \gamma = [2\mu_{11}^2 + \mu_{02}(\mu_{20} - \mu_{02})]/\Delta \\ \delta = 5[\mu_{12}(\mu_{20} - \mu_{02}) + \mu_{11}(\mu_{03} - \mu_{21})]/\Delta \\ \nu = 5[\mu_{21}(\mu_{02} - \mu_{20}) + \mu_{11}(\mu_{30} - \mu_{12})]/\Delta \\ \Delta = (\mu_{20} - \mu_{02})_2 + 4\mu_{11}^2 \end{cases} \tag{3.19}$$

$$\alpha_{vx} = \alpha_{vy} = \alpha_{vz} \tag{3.20}$$

1. When $A = B = 0$, $\alpha_{wx} = 0$ when $x_g = y_g$ and $\mu_{03} = \mu_{12} = \mu_{21} = 0$,

2. $\alpha_{wy} = 0$ when $x_g = y_g = 0$ and $\mu_{30} = \mu_{21} = \mu_{12} = 0$

The area of the triangle is given by the formula $a = m_{00}$

$$L_a = \begin{bmatrix} -aA & -aB & a(\frac{3}{Z_g} - C) & 3ay_g & -3ax_g & 0 \end{bmatrix} \tag{3.21}$$

The centroid of the triangle in the image can be expressed as

$$x_g = \frac{m_{10}}{m_{00}},\ y_g = \frac{m_{01}}{m_{00}} \tag{3.22}$$

$$L_{x_g} = \begin{bmatrix} \frac{-1}{Z_g} 0 \frac{x_g}{Z_g + \epsilon_1} x_g y_g + 4n_{11} - (1 + x_g^2 + 4n_{20)y_g} \end{bmatrix} \tag{3.23}$$

$$L_{y_g} = \begin{bmatrix} 0 \frac{-1}{Z_g} \frac{y_g}{Z_g + \epsilon_2} 1 + y_g^2 + 4n_{02} - x_g y_g - 4n_{11)-x_g} \end{bmatrix} \tag{3.24}$$

$$\begin{cases} \frac{1}{Z_g} = Ax_g + By_g + C \\ \epsilon_1 = 4(An_{20} + Bn_{11}) \\ \epsilon_2 = 4(An_{11} + Bn_{02}) \end{cases} \tag{3.25}$$

$n_{ij} = \frac{\mu_{ij}}{a}$ with

$$\begin{cases} \mu_{20} = m_{20} - ax_g^2 \\ \mu_{02} = m_{02} - ay_g^2 \\ \mu_{11} = m_{11} - ax_g y_g \end{cases} \tag{3.26}$$

Visual moments based on images can be calculated for a variety of geometric shapes such as circles, ellipses, triangles, rectangles, polygons, and so on. The computation of

visual moments is influenced by the object's shape, size, and orientation within the image.

3.3 Explicit image moments for a triangle

The visual moments of a triangle or polygon can be calculated using the enrolled polygon approach or a method of approximating the object's shape using a polynomial combination of functions.

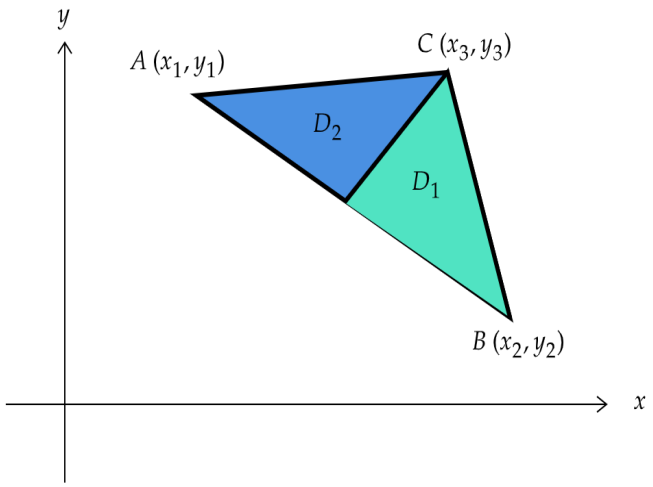

Figure 3.2: Triangle on the XY Image Plane with Integration Regions

Let ABC be a triangle in an image with coordinates $A(x_1, y_1)$, $B(x_2, y_2)$ and $C(x_3, y_3)$, see figure 3.2. The equations for each edge of the triangle are expressed as follows:
Line AB:

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1} = \alpha_{AB} \tag{3.27}$$

$$y = \alpha_{AB} x + y_1 - \alpha_{AB} x_1 \tag{3.28}$$

Line BC:

$$\alpha_{BC} = \frac{y_3 - y_2}{x_3 - x_2} \tag{3.29}$$

$$y = \alpha_{BC} x + y_2 - \alpha_{BC} x_2 \tag{3.30}$$

Line AC:

$$\alpha_{AC} = \frac{y_3 - y_1}{x_3 - x_1} \tag{3.31}$$

$$y = \alpha_{AC} x + y_1 - \alpha_{AC} x_1 \tag{3.32}$$

So,

$$m_{01} = \int\int_D ydxdy = \int\int_{D1} ydxdy + \int\int_{D2} ydxdy \tag{3.33}$$

$$m_{01} = \int_{x_1}^{x_3} (\int_{y=\alpha_{AB}x+y_1-\alpha_{AB}x_1}^{y=\alpha_{AC}x+y_1-\alpha_{AC}x_1} ydy)dx + \int_{x_3}^{x_2} (\int_{y=\alpha_{AB}x+y_1-\alpha_{AB}x_1}^{y=\alpha_{BC}x+y_2-\alpha_{BC}x_2} ydy)dx \tag{3.34}$$

$$I_1 = \int_{x_1}^{x_3} \Big(\alpha_{AC} - \alpha_{AB}\Big) x - \Big(\alpha_{AC} - \alpha_{AB}\Big) x_1 dx \tag{3.35}$$

$$I_1 = \Big(\alpha_{AC} - \alpha_{AB}\Big) \left[\left(\frac{x_3^2 - x_1^2}{2}\right) - x_1\Big(x_3 - x_1\Big)\right] \tag{3.36}$$

$$I_1 = \Big(\alpha_{AC} - \alpha_{AB}\Big)\Big(x_3 - x_1\Big)\left(\frac{x_3 + x_1}{2} - x_1\right) \tag{3.37}$$

Change equation of AB :

$$\alpha_{BA} = \frac{y_1 - y_2}{x_1 - x_2} \tag{3.38}$$

$$y = \alpha_{BA}x + y_2 - \alpha_{BA}x_2 \tag{3.39}$$

$$I_2 = \int_{x_3}^{x_2} \int_{y=\alpha_{BA}x+y_2-\alpha_{BA}x_2}^{y=\alpha_{Bc}x+y_2-\alpha_{BC}x_2} \tag{3.40}$$

$$I_2 = \Big(\alpha_{BC} - \alpha_{BA}\Big)\Big(x_2 - x_3\Big)\left(\frac{x_2 + x_3}{2} - x_2\right) \tag{3.41}$$

Then,

$$\begin{aligned} m_{01} = \alpha_{AC} - \alpha_{AB}x_3 - x_1\frac{x_3 + x_1}{2} - x_1 \\ +\alpha_{BC} - \alpha_{BA}x_2 - x_3\frac{x_2 + x_3}{2} - x_2 \end{aligned} \tag{3.42}$$

3.4 Visual feedback control of a triangle using a servoing loop

We will now demonstrate how to develop a control strategy using a multiple triangle configuration. For this purpose, we consider the vector $im = (a,\ x_g,\ y_g,\ \theta)$ which represents the area, center of gravity, and orientation extracted from the image using image moments for a specific triangle, as outlined in the previous sections and in [90]. The following equation enables the definition of the servoing scheme representation for a triangle.

$$\begin{pmatrix} \dot{a} \\ \dot{x_g} \\ \dot{y_g} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} L_a \\ L_{x_g} \\ L_{y_g} \\ L_\theta \end{pmatrix} \dot{c} = L_m \, c_m \tag{3.43}$$

In this work, we compute the matrix L at the desired pose of the triangular primitive. This last equation provides four equations to control the velocity and the angular rotation around the camera axis.To gain control over the remaining two rotational angles, it is necessary to incorporate two additional moments derived from the fourth- and fifth-order image moments [16].

3.5 Closed-loop visual control using multiple triangular markers

This study focuses on the manipulation of general non-rigid objects modeled using triangular meshes. In cases involving n triangles, a standard approach consists of assembling the individual interaction matrices by row-wise concatenation, followed by the computation of a unified control command for the camera, as detailed below

$$c_n = -\lambda_n \begin{pmatrix} L_1 \\ \vdots \\ L_n \end{pmatrix}^+ \begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix} \tag{3.44}$$

In this study, we propose calculating an individual correction for each triangle independently.

$$c_i = -\lambda L_i^+ e_i, \; i = 1, \cdots, n \tag{3.45}$$

Subsequently, a global correction is obtained by averaging the individual corrections computed for each triangle(see Figure 3.3).

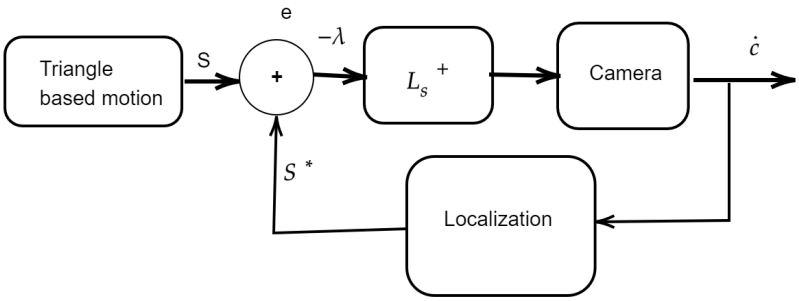

Figure 3.3: Block diagram of a visual servoing of image-moment visual cues

$$c_n = \frac{1}{n}\sum_{i=1}^{n} c_i \tag{3.46}$$

We demonstrate that this method achieves improved convergence compared to the naive approach, offering greater consistency and enhanced stability in the results.
Indeed, when the individual triangles experience varying motions, the global correction scheme becomes less realistic. This is because it attempts to compute a single rigid transformation for the entire object, whereas in reality, each triangle may undergo distinct rigid or even non-rigid deformations, as previously observed in earlier sections [90].

3.6 Results on simulated data

3.6.1 Results for one triangle

A triangular visual primitive is employed to assess the performance of the visual servoing strategy, utilizing the four previously defined image moments. These moments are used to control and regulate four degrees of freedom of the camera motion (velocity camera axis angular speed: vx, vy, vz, and ωz, respectively). In the desired condition, the triangle is placed in front of the camera with $A = B = 0$ so that the camera is lined up with the world marker. The triangle's orthogonal distance from the center of the camera is 2 meters. The camera is first translated by 0.5 meters along its X axis and turned by 5 degrees about its optical axis (Z). The camera's focal length is $f = 1500$; Unless stated

otherwise, distances are expressed in meters, and angles are measured in degrees. We use $\lambda = 1$ for the servoing gain, we define a stop condition for the servoing loop whenever the average difference between the visual characteristics falls below a certain threshold. 0.01. We compare the moment-based method with the earlier line-based approach, which employed the three lines extending from the edges of the triangle as visual elements [90]. Additionally, we examine the approach's stability using random-centered Gaussian noise and standard 2 and 3 pixel sizes. At each iteration, the visual feature was given more noise. Figure 3.4 shows the initial (green) and desired (red) configurations of the triangle in the image plane. The visual discrepancy highlights the initial positioning error that the servoing controller must correct. Figure 3.5 illustrates the convergence process: the green triangle represents the reference configuration, while the red triangle shows the corrected view after applying moment-based visual servoing. The close alignment demonstrates the controller's ability to minimize visual error. Figure 3.6 presents a qualitative comparison of convergence under Gaussian image noise. The top row (2-pixel noise) and bottom row (3-pixel noise) both show successful alignment, confirming the method's robustness to measurement uncertainty. The quantitative performance is detailed in Figures 3.9 and 3.10, which plot the camera's translational and rotational velocities over iterations. The smooth decay of velocities toward zero indicates stable and well-damped convergence without oscillatory behavior. Figures 3.11 and 3.12 provide further insight into noise sensitivity by showing the evolution of visual error under 3-pixel and 2-pixel noise, respectively. The error curves converge steadily, with only a modest increase in iteration count ($\approx$ 10 additional steps) for the higher noise level, underscoring the method's practical reliability. For comparison, Figure 3.15 displays results using the line-based method. The corrected view (red) shows poorer alignment with the reference (green), and the convergence is less stable observations consistent with the higher variability noted in line-based quantitative results (not shown). Finally, Figure 3.18 contrasts the moment-based and line-based approaches directly, illustrating the superior smoothness and precision of the moment-based strategy. The moment-based error decreases monotonically, while the line-based error exhibits fluctuations and slower convergence. We conducted our simulations on a multi-core processor i3 lap-top with 4Go RAM and Matlab2015a.

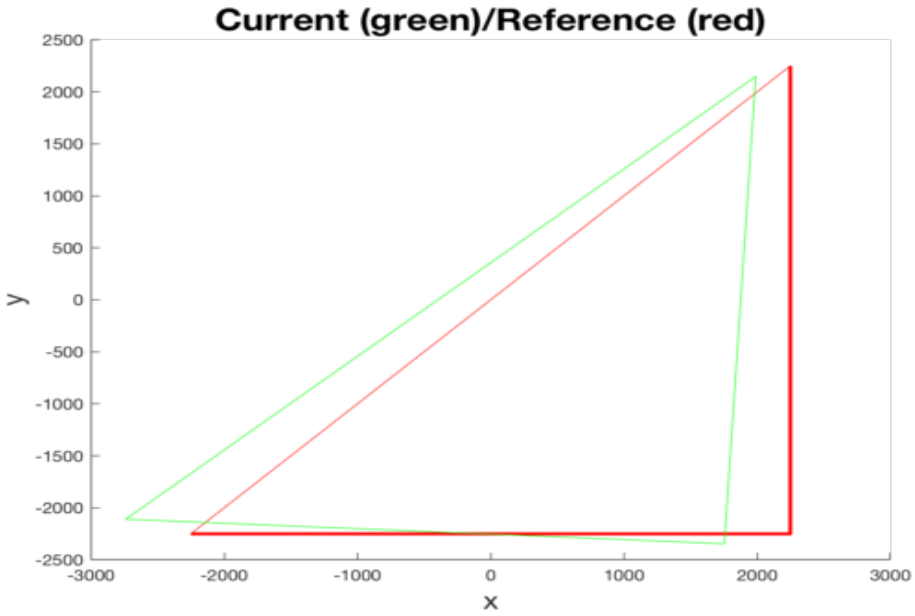

Figure 3.4: The target and actual positions of the triangle.

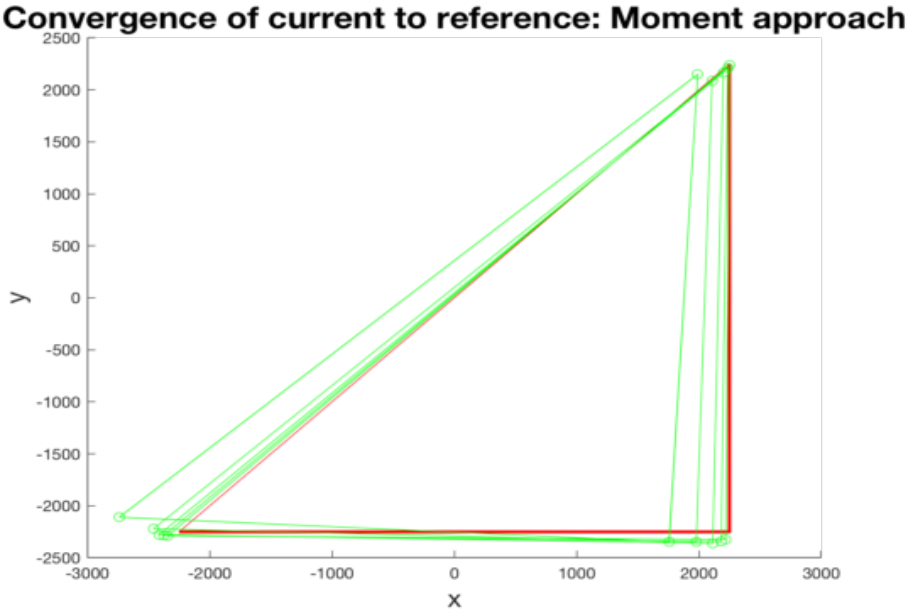

Figure 3.5: The test results are displayed. The green triangles represent the reference, while the red triangles indicate the corrected view.

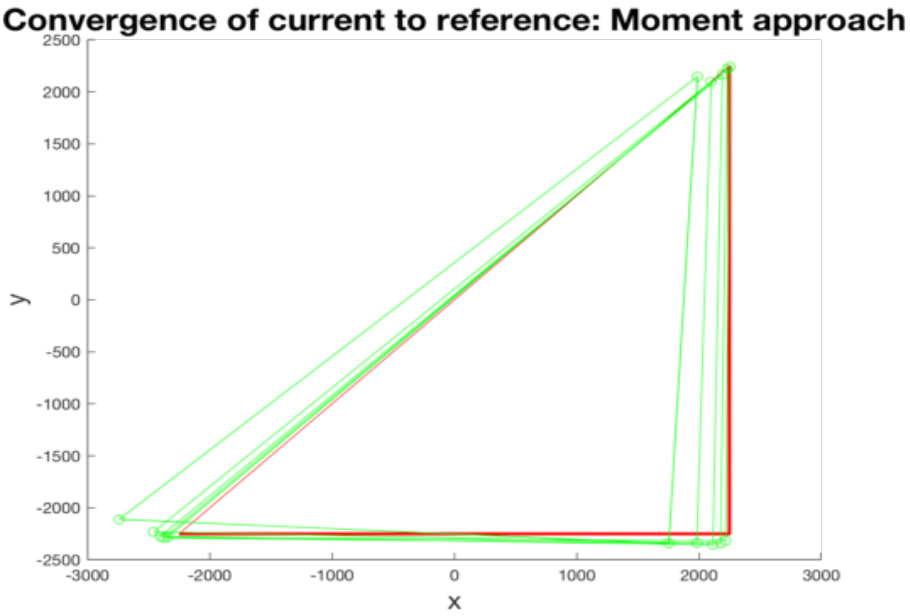

Figure 3.6: A qualitative analysis is conducted to assess the influence of Gaussian noise centered on the image. Results corresponding to a standard deviation of 2 pixels are shown in the top row, while those with a standard deviation of 3 pixels are displayed in the bottom row.

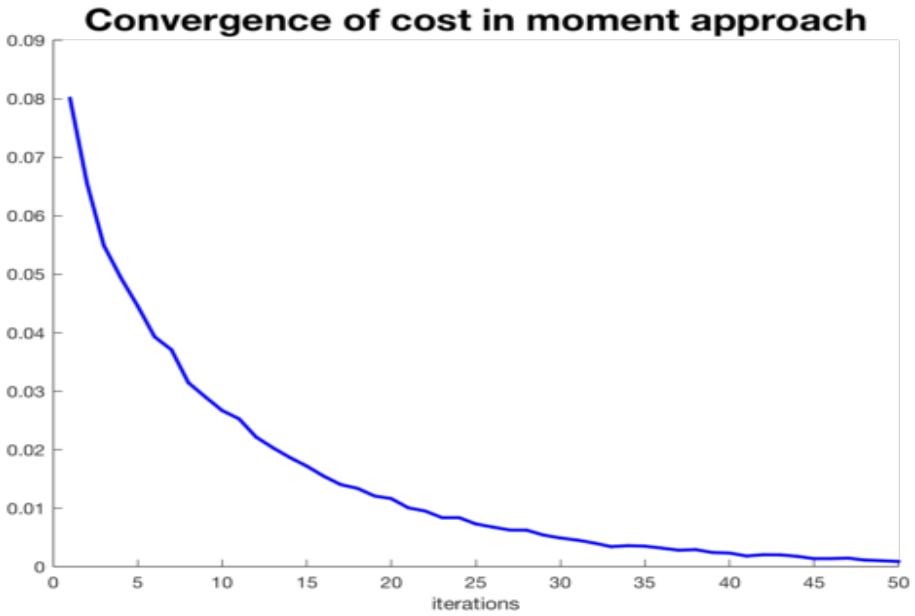

Figure 3.7: Cost function minimization over iterations for image moment visual servoing

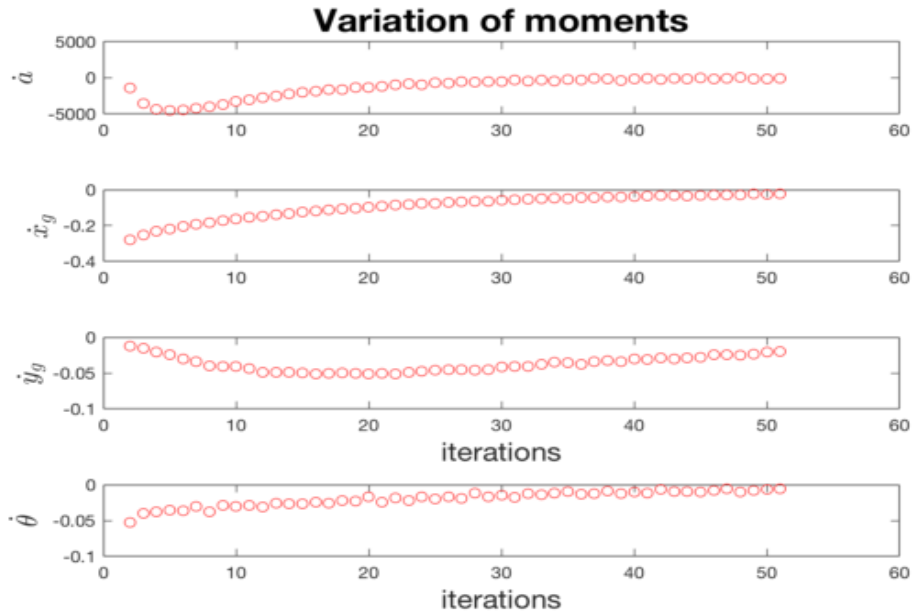

Figure 3.8: Results of applying the moment-based visual servoing technique, compared to the line-based visual servoing method.

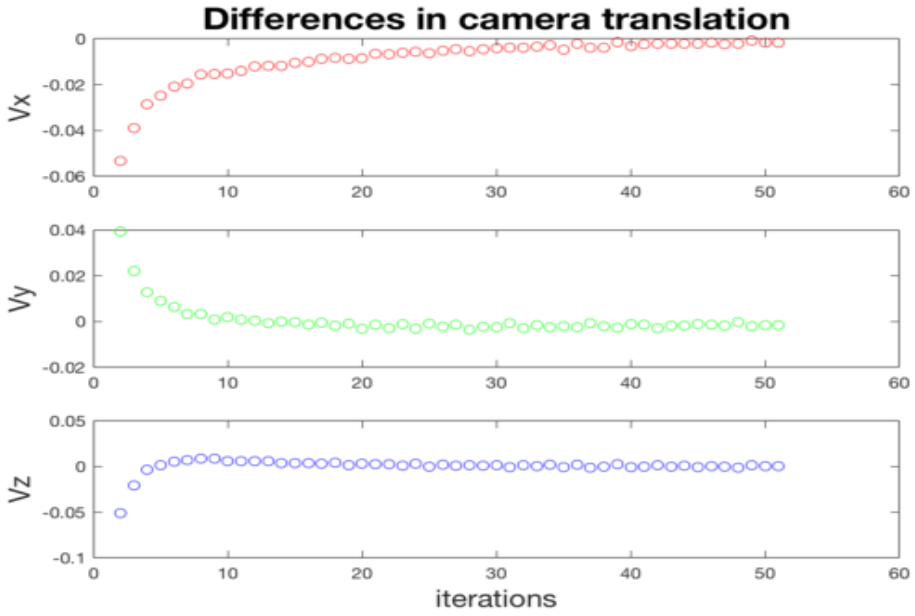

Figure 3.9: Results of the experiment with L represents camera speed in translation.

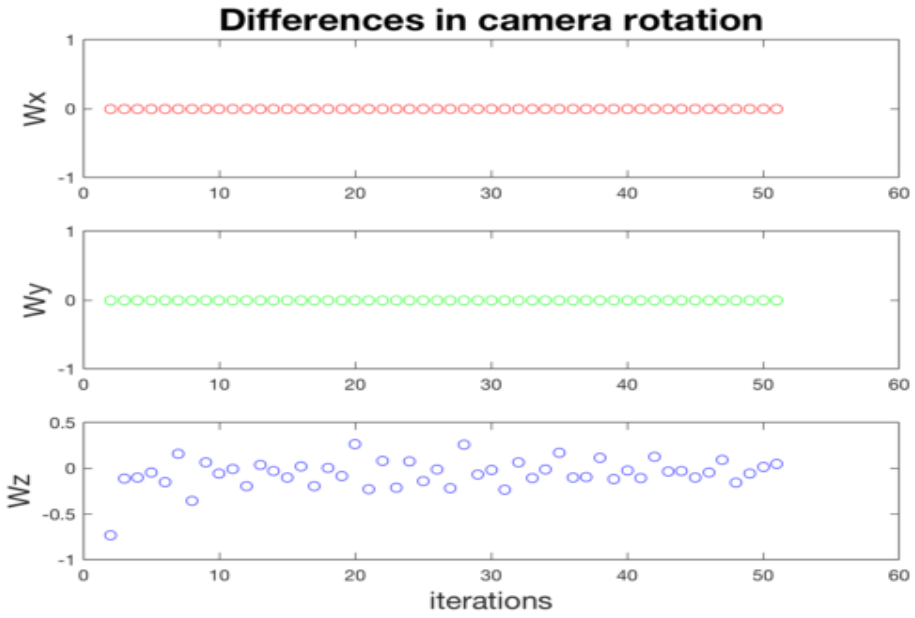

Figure 3.10: Results of the experiment with L represents camera speed in rotation.

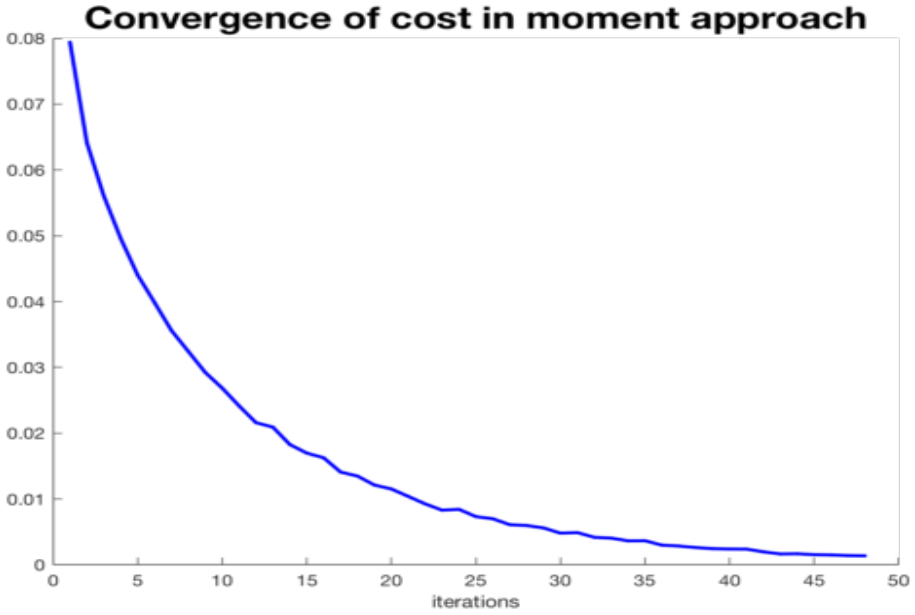

Figure 3.11: The impact of centered Gaussian image noise with a standard deviation of 3 pixels.

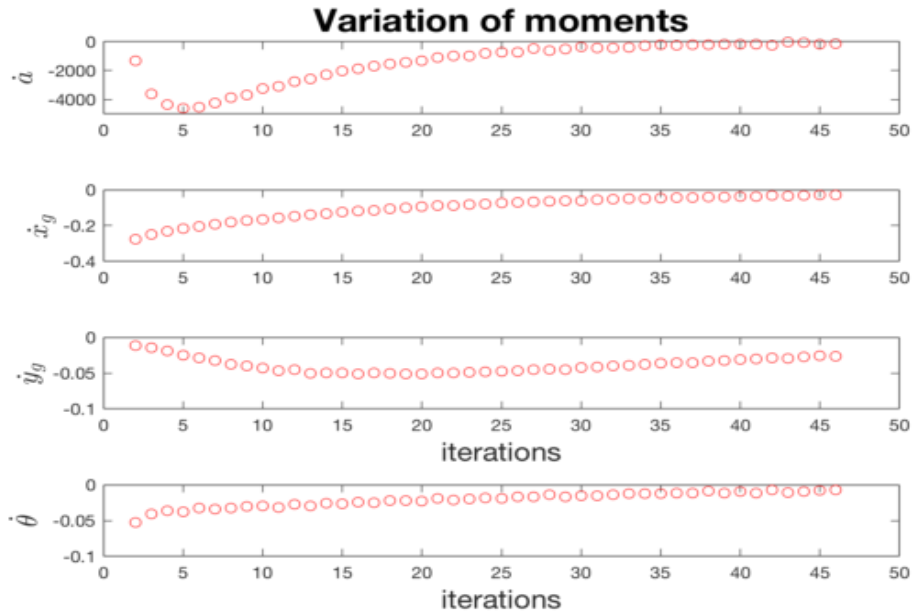

Figure 3.12: The impact of centered Gaussian image noise with a standard deviation of 2 pixels.

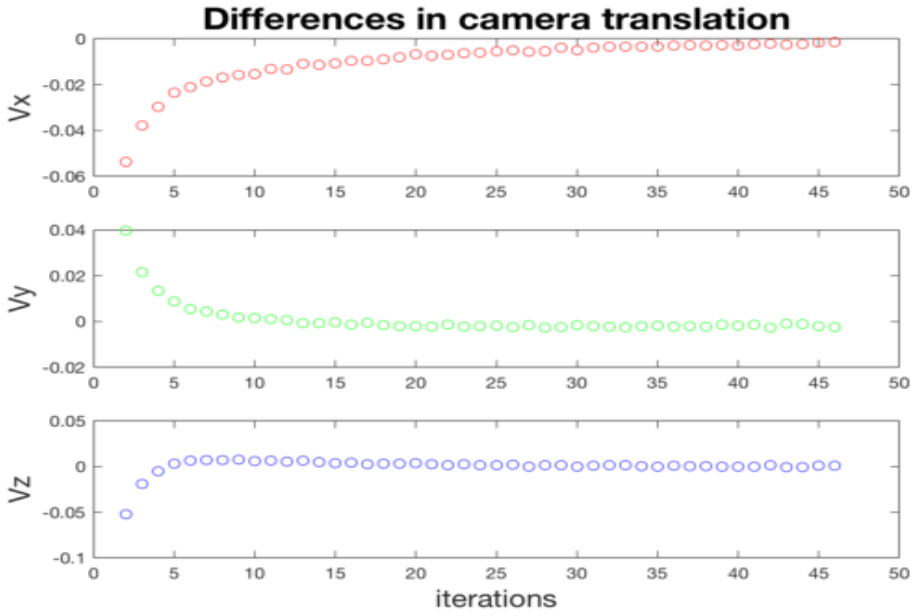

Figure 3.13: Results of the experiment with L represents camera speed in translation.

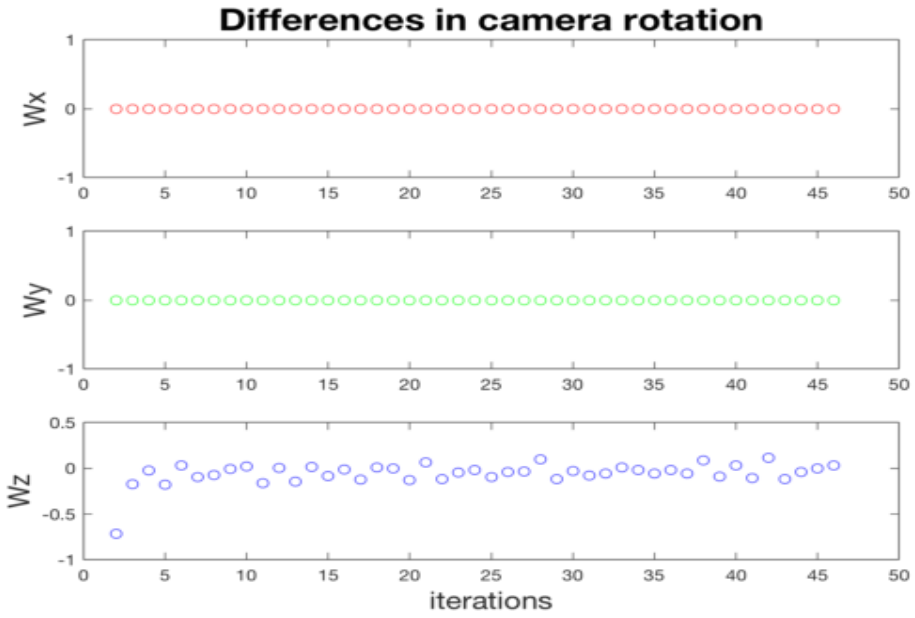

Figure 3.14: Results of the experiment with L represents camera speed in rotation.

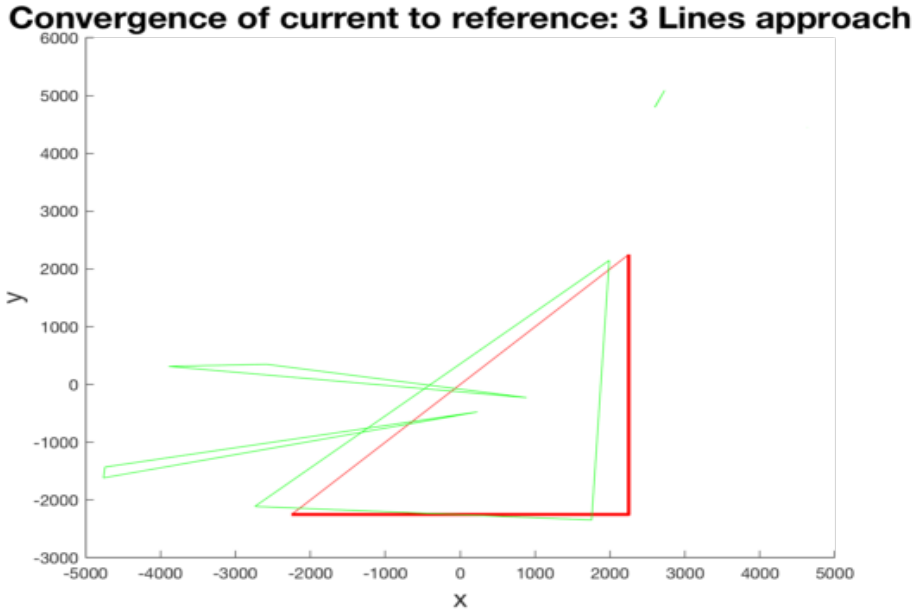

Figure 3.15: The test results are displayed. The green triangles represent the reference, while the red triangles indicate the corrected view.

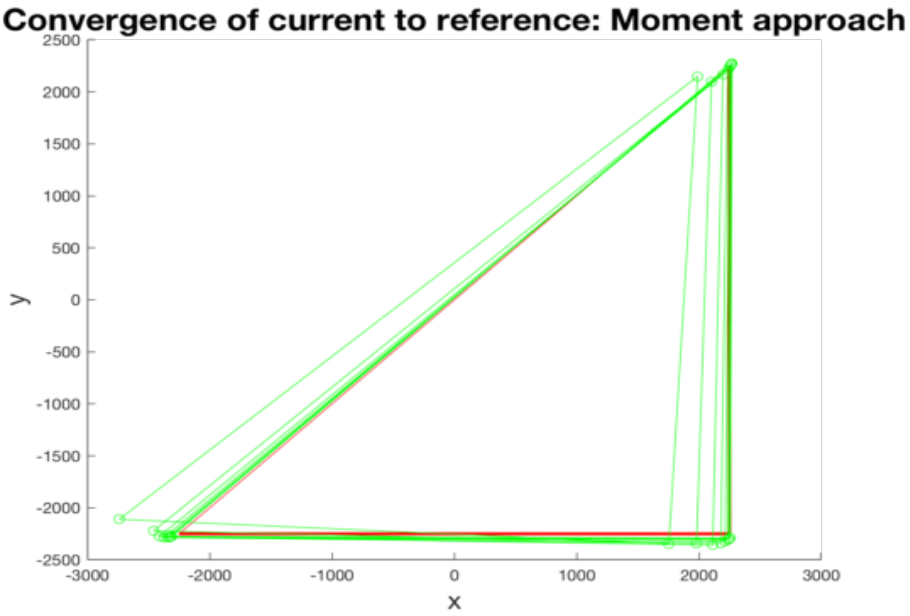

Figure 3.16: The results of the test are shown, with the green triangles representing the reference and the red triangles depicting the corrected view.

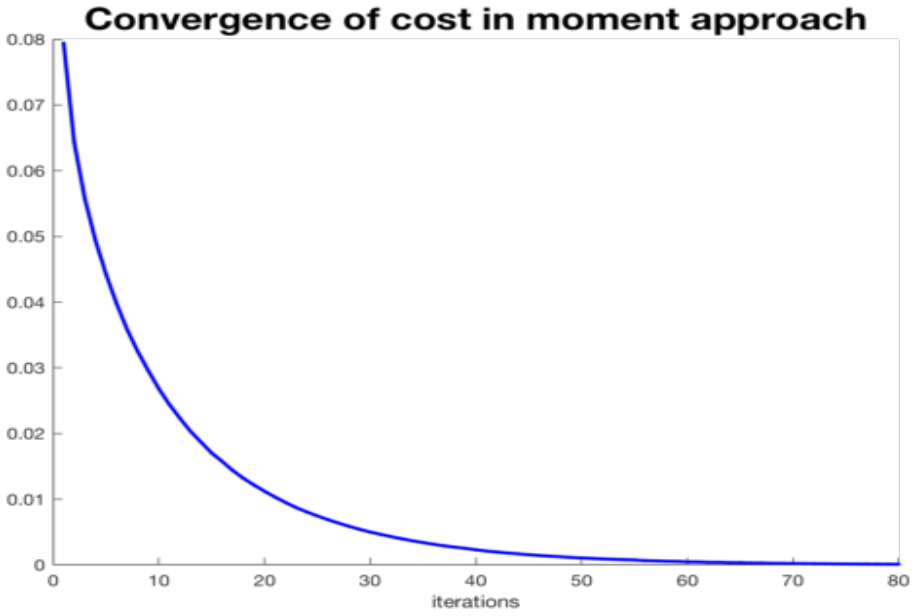

Figure 3.17: Evolution of the tracking error during moment-based visual servoing

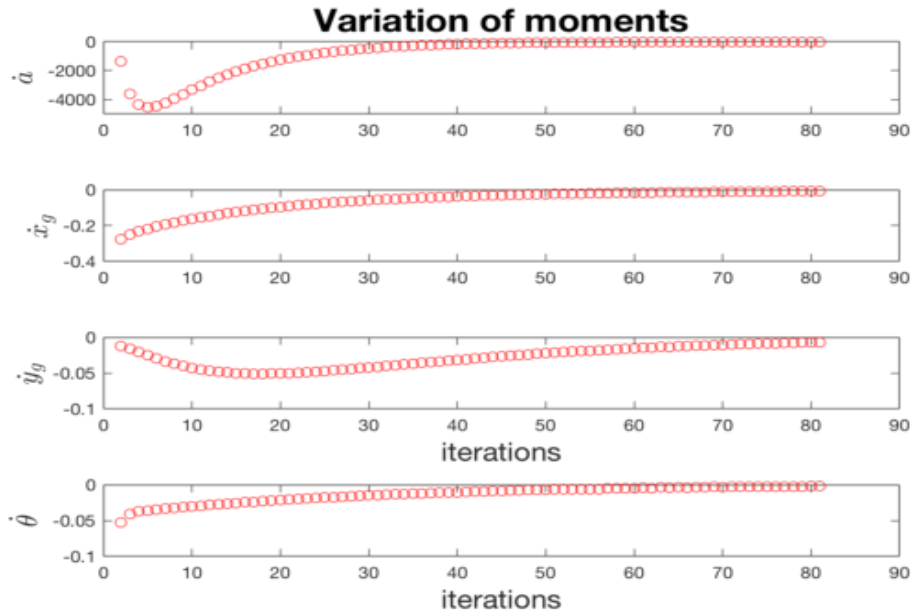

Figure 3.18: The results of applying the moment-based visual servoing method are shown, compared with the line-based visual servoing approach.

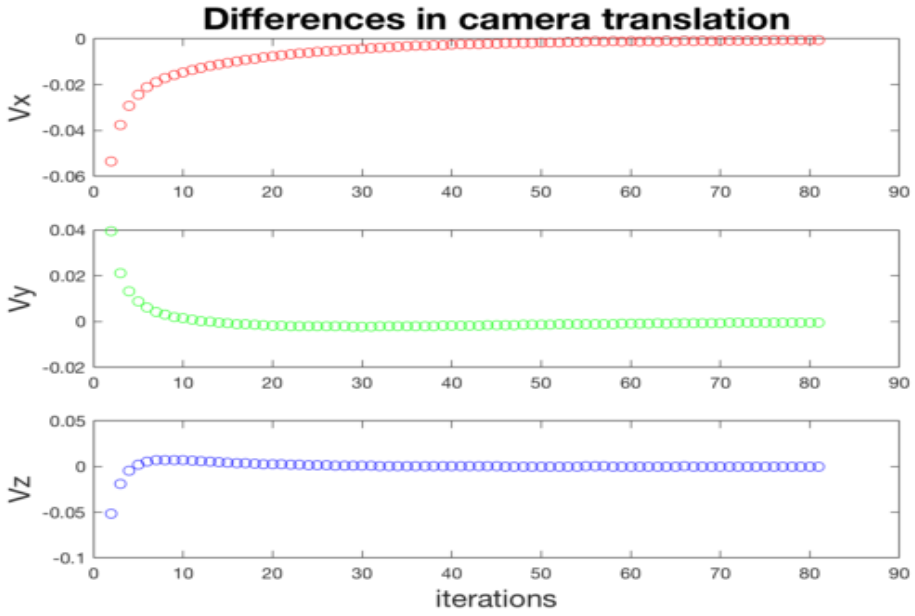

Figure 3.19: Test of the experiment with L represents camera speed in translation.

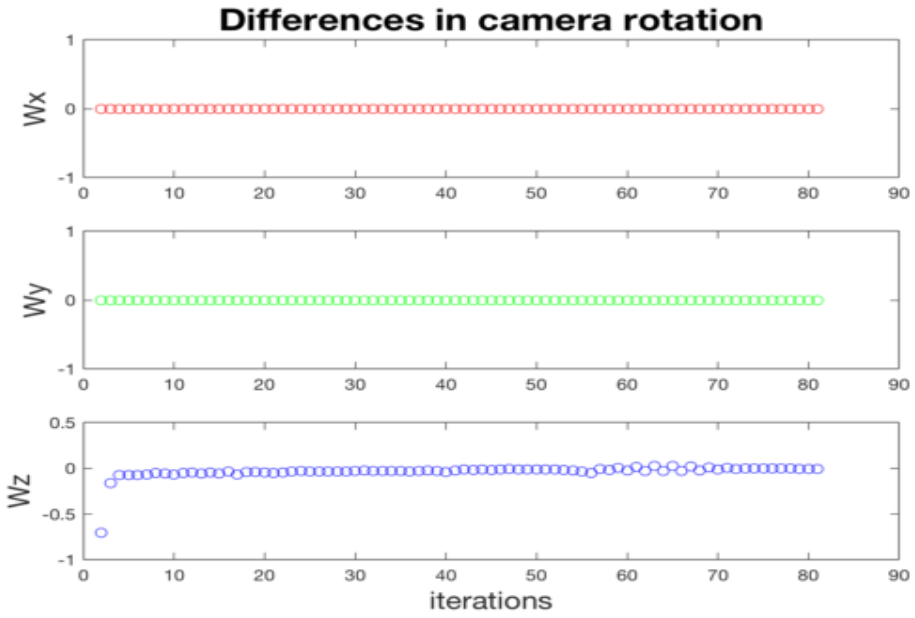

Figure 3.20: Test of the experiment with L represents camera speed in rotation.

3.6.2 Results for multiple triangles

This study explores a visual servoing approach based on image moments applied to multiple triangular primitives, with the objective of controlling four degrees of freedom of the camera, namely its translational velocity along the optical axis and its angular velocity around three orthogonal axes: v_x, v_y, v_z and ω_z respectively),and so two triangles are placed at distinct positions. In the target configuration, the camera is aligned with the world marker, and each triangle is oriented directly in front of the camera to ensure optimal visibility $A = B = 0$. The perpendicular distance between the triangle and the optical center of the camera is given by 2 meters.At the initial state, the camera undergoes a rotation of 5 degrees around its optical axis Z and translated by 0.5 meters along its X axis.The camera's focal length is set to$f$. We have chosen two triangles: one triangle on the (oxy) plane with vertices at position $p0(:,1) = [3;3;2;1]$; $p0(:,2) = [3;-3;2;1]$; $p0(:,3) = [-3;-3;2;1]$ (in Matlab notation). We applied a transformation $T1$ which represents a translation followed by a rotation. The second triangle has vertices at the points $p1(:,1) = [3;6;2;1]$;$p1(:,2) = [3;0;2;1]$; $p1(:,3) = [-3;0;2;1]$.
We applied another transformation T_2 different than T_1. From the experiments conducted, it appears that the average control-based approach is more stable than the approach based on the use of a single interaction matrix for all triangles. It can be observed that the speed of convergence is fast enough for the instant of approach (after 150 iterations). We also ran similar set-up on three triangles and the obtained control converges as well as can be seen in figure 3.23.
Figure 3.21 illustrates the visual servoing process when a single interaction matrix is used for both triangles. The corrected view (red) attempts to align with the reference

(green), but residual misalignment can be observed, indicating that a global rigid correction cannot fully compensate for independent triangle motions. Figure 3.22 presents the proposed averaged control approach, where an individual correction is computed per triangle and then averaged. The result shows significantly improved alignment, confirming that this strategy better handles independent motions. Figure 3.23 extends the validation to three triangles, demonstrating that the averaged approach remains stable and converges within approximately 150 iterations, even as the number of primitives increases. The experiments demonstrate that the average control-based approach is more stable than using a single interaction matrix for all triangles. This is because the averaging strategy respects the independent motions of each triangle, whereas a single matrix forces an unrealistic global rigid transformation. The convergence remains efficient, with visual alignment achieved in a practical number of iterations. We ran our simulations on core i3 lap-top with 4Go RAM and Matlab2015a.

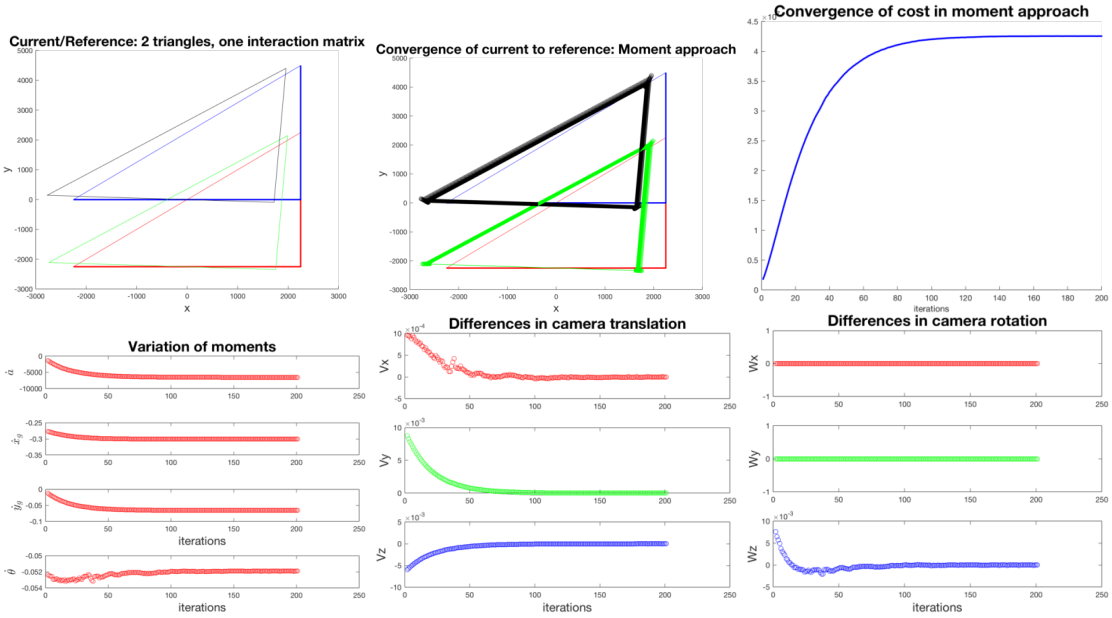

Figure 3.21: The results demonstrate the application of moment-based visual servoing on two triangles, utilizing transformations of the triangles with a single interaction matrix.

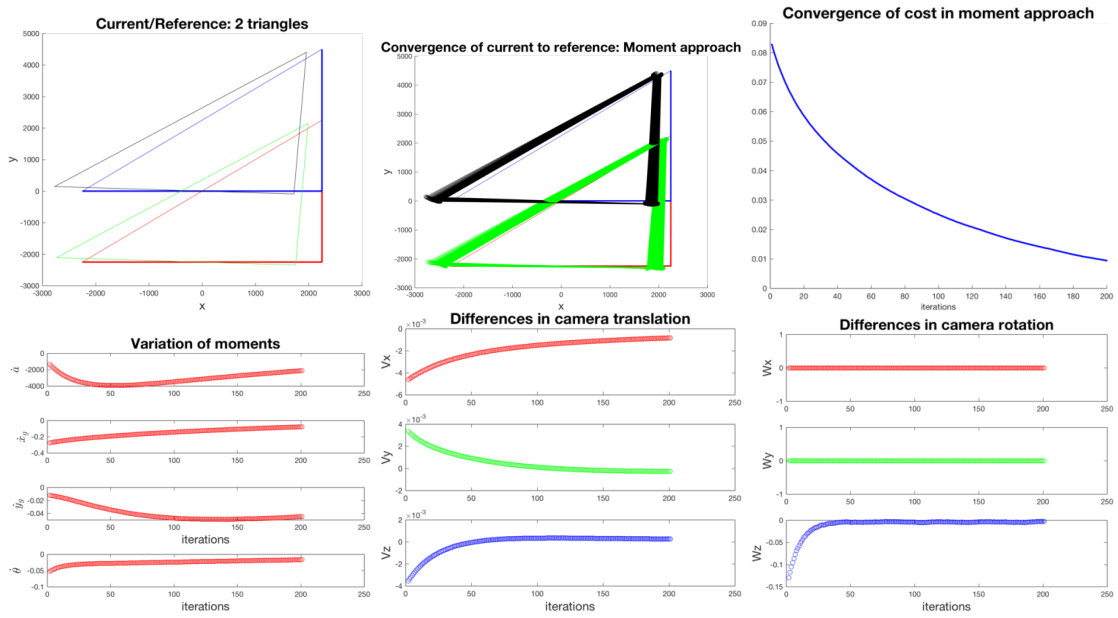

Figure 3.22: The results demonstrate the application of moment-based visual servoing on two triangles, utilizing transformations of the triangles with a single interaction matrix test 2.

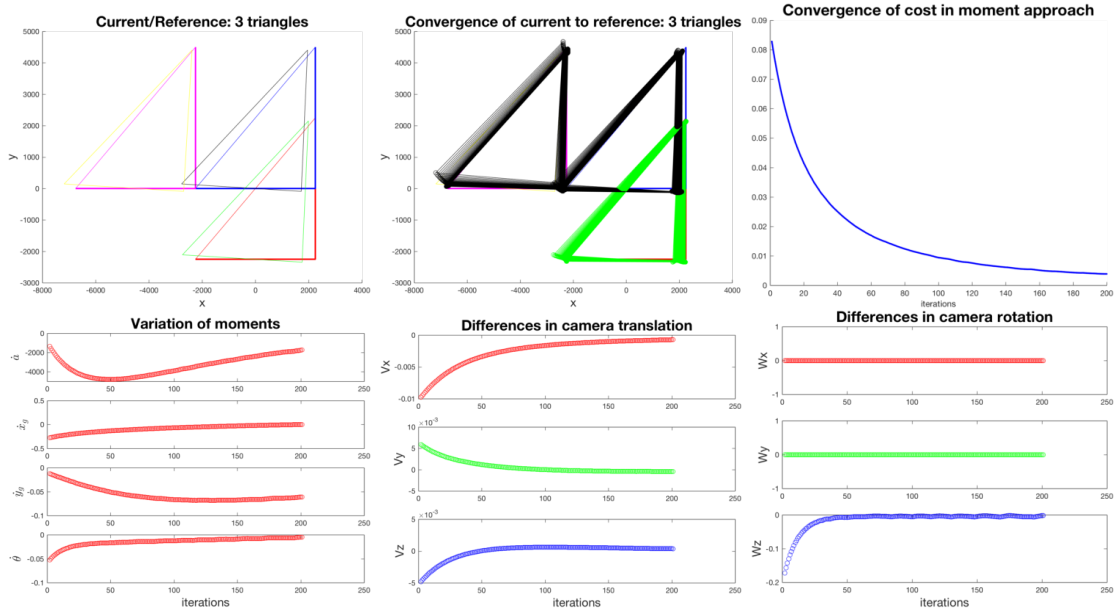

Figure 3.23: The results showcase the application of moment-based visual servoing across multiple triangles, highlighting the performance and stability of the approach in controlling the camera's degrees of freedom.

3.6.3 Results for one deformed triangle

This subsection evaluates the performance of the moment-based visual servoing strategy when applied to a single deformable triangle. The triangle undergoes elastic deformation

along the z-axis, with a maximum displacement of 0.5 meters at its highest point, as illustrated in Figure 3.24.
Figure 3.25 displays the camera translation velocities (v_x, v_y, v_z) over iterations. The profiles show a smooth, adaptive response as the camera adjusts its position to compensate for the deformation. Figure 3.26 presents the camera rotation velocities $(\omega_x, \omega_y, \omega_z)$. The controller primarily adjusts ω_z (rotation around the optical axis) to align the triangle orientation, while rotations around x and y remain minimal, consistent with the planar nature of the deformation. Figure 3.27 provides a direct comparison between moment-based and line-based servoing for the deformed triangle. The moment based approach (top) shows a stable decrease in visual error, while the line-based method (bottom) exhibits oscillations and slower convergence. This reaffirms the superiority of moments in handling non-rigid deformations, thanks to their global and integrative nature.

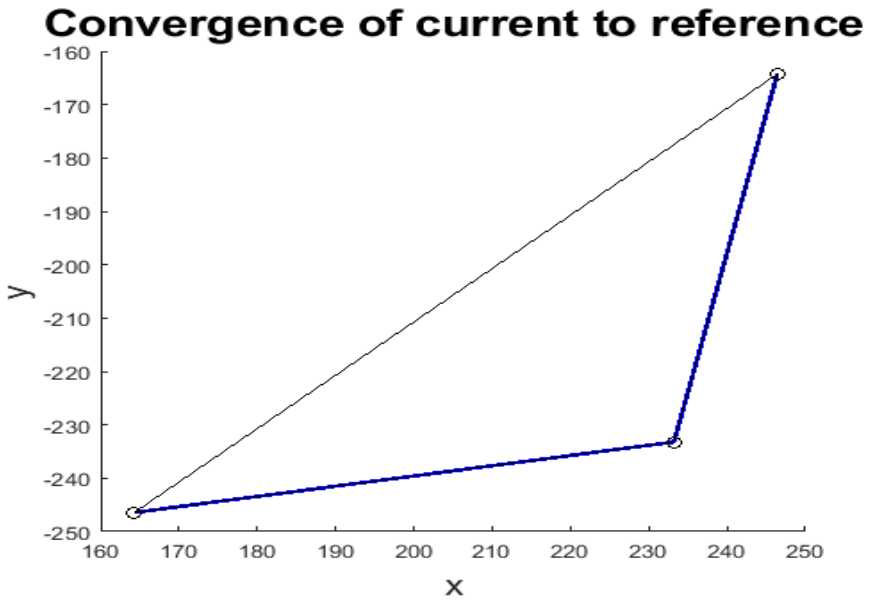

Figure 3.24: The desired and current positions of the deformed triangle illustrate the system ability to track and align the triangle under deformation, showcasing the effectiveness of the visual servoing strategy

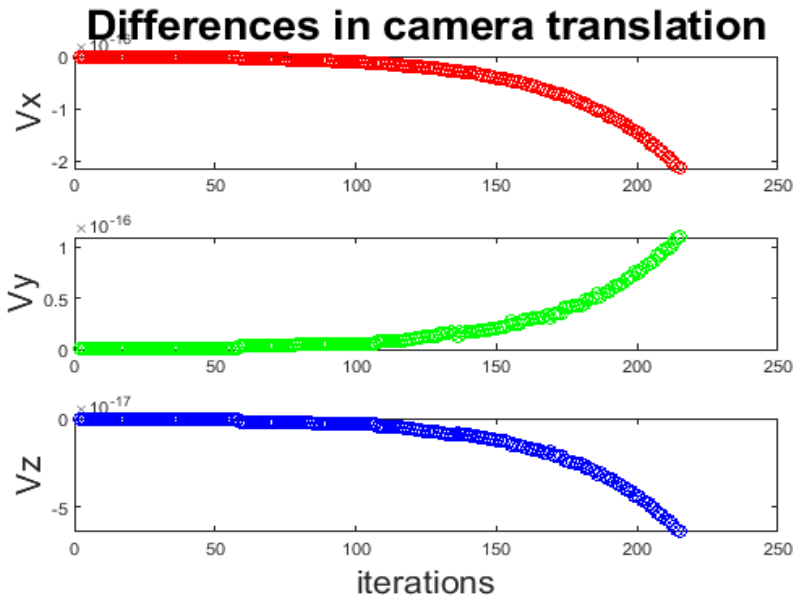

Figure 3.25: Test of the experiment with L represents camera speed in translation.

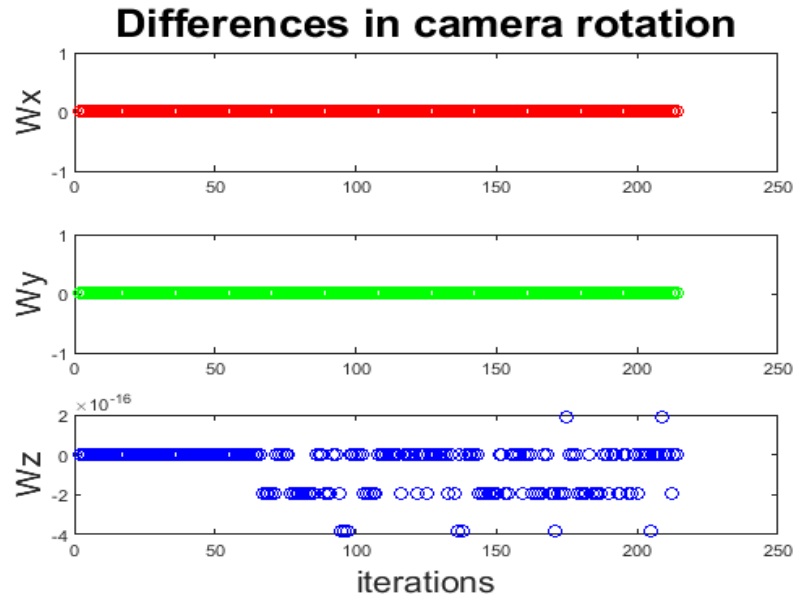

Figure 3.26: Test of the experiment with L represents camera speed in rotation.

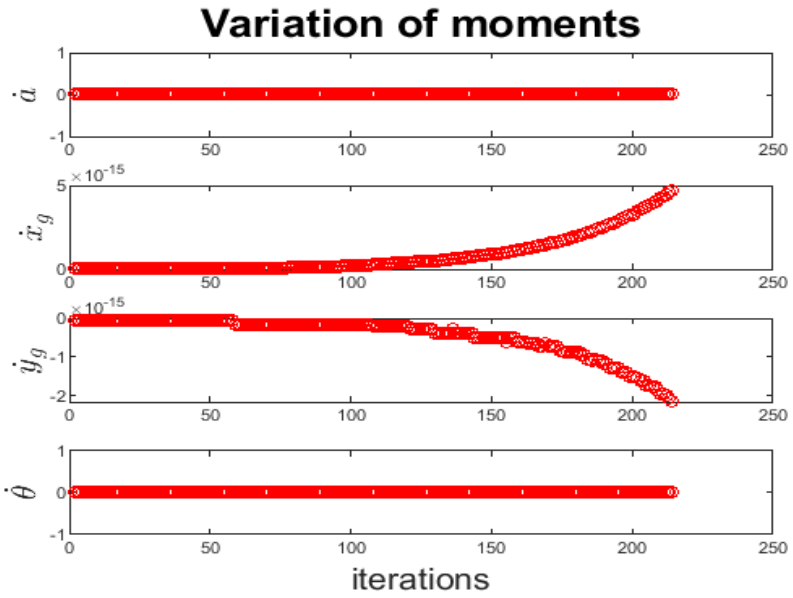

Figure 3.27: The results of applying moment-based visual servoing, compared to the line-based approach using interaction matrix L, for a single deformed triangle, demonstrate the variation in visual moments. These variations highlight the robustness of the moment-based method in capturing geometric deformations and ensuring stable servoing performance under non-rigid transformations.

3.7 Conclusion

We show how we obtain the visual servoing by using the method of image moments, where we derived explicitly formulas of the method by using analytic integrals for the area, able to handle simple geometrical. The visual moments of a triangle of order 0 which represents the area of the triangle , order 1 which represents the center of gravity of a triangle and order 2 represents the orientations of this triangle with these parameters we can control our camera on 4 degrees of freedom. Our approach is to find a better camera position correction so we can see the triangles as the initial view despite applying any form of rigid movement on the triangles. This is achieved by computing individual corrections for each triangle and subsequently applying their average to adjust the camera motion. We achieve this by calculating the correction for each triangle and then apply an average correction on the camera. We showed that this approach is more stable and gives results better than using a single interaction matrix for all triangles. On a rigid triangle and a deformable triangle, a set of visual perception tests was conducted. Through these testing, it was shown that visual characteristics based on moments of order 1 and 2 are less reliable than visual characteristics based on points.

Chapter 4

A general framework for visual servoing on non-rigid deformations

4.1 Introduction

Due to the ease of the equations governing linear elasticity and the potential for computing time optimization, it is frequently used to describe deformable materials. When the deformations and displacements of an elastic tissue are sufficiently minimal, its physical behavior can be thought of as linear. The method is known as "kinematic" when it applies to elastic deformations in solid mechanics. Kinematically acceptable displacements are those that minimize the total potential energy and satisfy a stable equilibrium condition. The majority of the research on visual servoing has been done in conjunction with optical imaging. Typically, a monocular video camera that detects light reflections on the scene's objects produces the image [45]. We highlight two medical applications of visual servoing that must handle deformation: Robotic needle guidance can enhance diagnostic accuracy and therapeutic outcomes, while ultrasound imaging is utilized to monitor object deformation during procedures[15]. The majority of ultrasonic imaging equipment have the benefit of producing video-quality images, making it possible to utilize them for real-time robotic applications, control strategies based on visual servoing are employed to regulate the motion of dynamic systems[42]. Over the past ten years, the use of robotic needles with ultrasound guidance has attracted significant interest from the scientific community[15]. For both diagnosis and therapy, needle insertion methods under ultrasonography are often employed [22]. The challenge of planning and trajectory control is used to formulate the insertion of the needle into a deformable tissue. By integrating digital models that account for needle deflection and soft tissue deformation, a new concept of needle steering has been introduced, along with a Jacobian-based approach for needle manipulation [28]. When the real goal that has to be attained is moving, as is typically the case with disruptions brought on by a patient's physiological movements, the positioning inaccuracy is likewise increased. Our article's issue is how to

position the camera to stay as close as possible to the visual cues of the reference position for each mesh triangle given the deformation of each triangle. Using our method, considering the deformation observed in each triangle, we first computed the camera correction based on the difference in the triangle's area. Subsequently, a global correction for the camera was derived by aggregating the individual adjustments[57]. The needle is a common instrument employed in various medical procedures, including aspiration biopsies that aid in the diagnosis of malignant conditions. However, in order to correctly direct a needle and achieve the objective, the physician must possess a wide range of abilities [1]. Guidance based on imaging techniques presents significant potential for enabling a broad spectrum of applications[15]. Various imaging modalities have been utilized in the medical field to support both diagnostic and surgical procedures. Among these, ultrasound (US) imaging provides notable advantages in terms of integration, safety, and portability. Specifically, conventional two-dimensional ultrasound is widely adopted due to its real-time imaging capabilities, high spatial resolution, and broad availability in clinical settings. This work aims to achieve accurate cross-sectional imaging of a target structure through the use of a robot-actuated 2D ultrasound probe, guided by a visual servoing approach. The control strategy directly leverages ultrasound images acquired by the probe for feedback. Rafik Mebarki et al [61] proposed a visual servoing approach that utilizes two-dimensional ultrasound images. The primary goal is to control a robotic system equipped with a 2D ultrasound probe in order to acquire a specific cross-sectional view of a target structure. This method relies on visual features derived from combinations of image moments extracted from the ultrasound data [16]. An emerging group of researchers is actively exploring robotic assistance techniques, with a particular focus on the steering and control of flexible needles featuring tapered tips [7] These needles are employed in procedures such as aspiration biopsies, which are commonly used for cancer diagnosis. [23] A method has been proposed for guiding a flexible needle with a tapered tip to a target located within moving tissue [90].

4.2 Piecewise Jacobian approach to global image-to-camera interaction modeling

The interaction matrix associated with a point in terms of its coordinates $p = (x, y)$ in the image space, with a depth Z relative to the camera reference, the interaction matrix is expressed by constants outlined in the appendix.

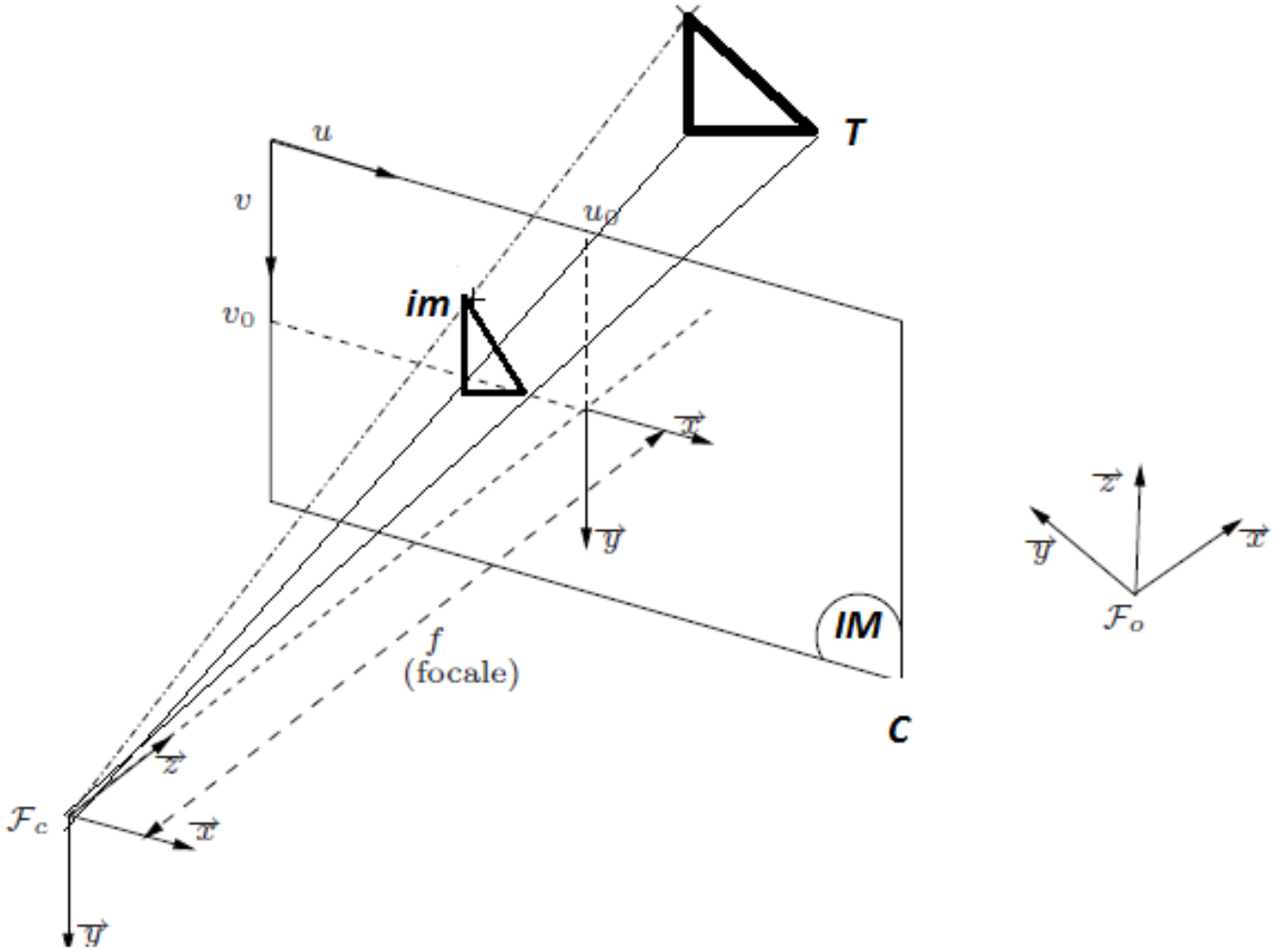

Figure 4.1: Geometric analysis of triangle projection under perspective imaging.

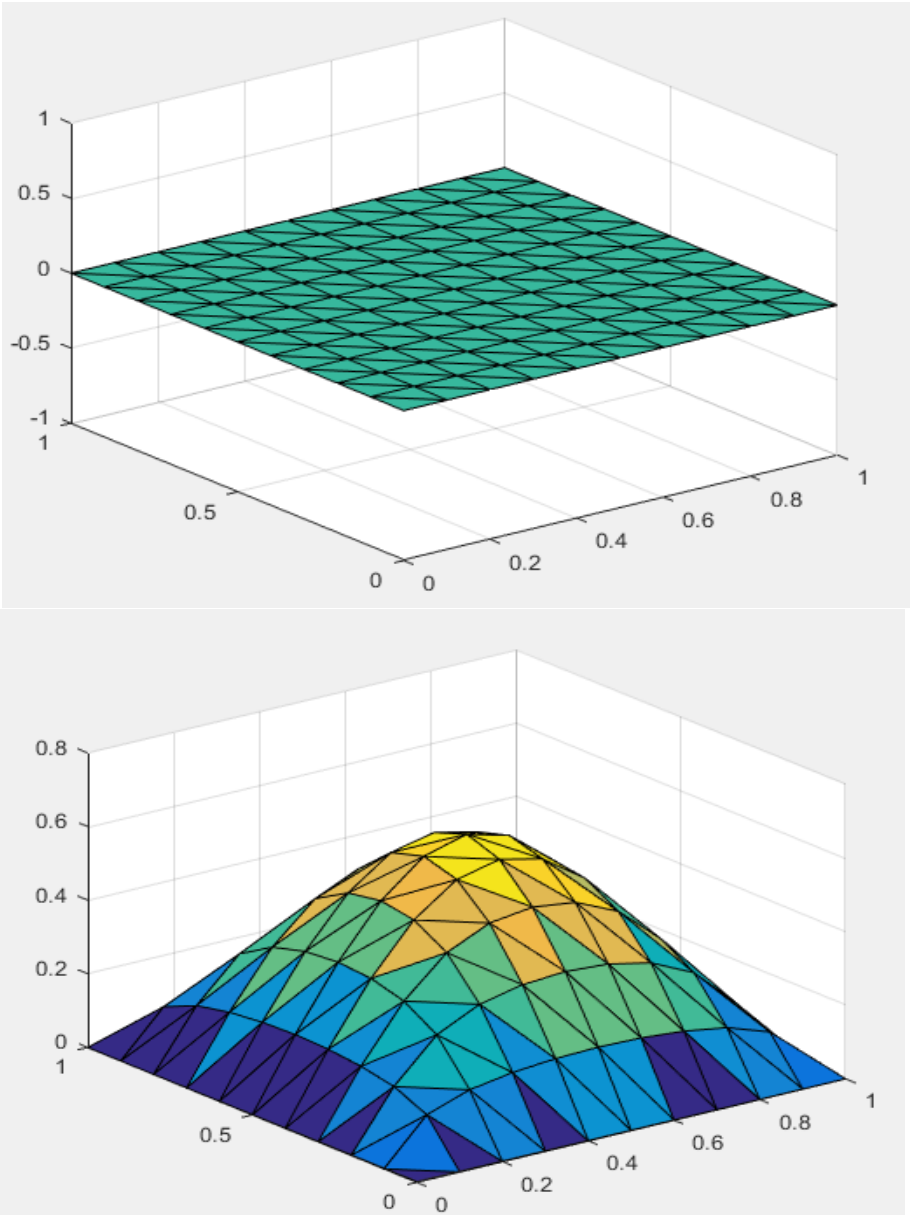

Figure 4.2: Comparison of pre-deformation and post-deformation configurations.

4.3 Visual feedback control for non-rigid deformation compensation

4.3.1 Visual feedback control of a triangle using servoing loop dynamics

We will now demonstrate how to develop a control strategy using a mesh configuration composed of multiple triangles.

To do this, we consider the vector $s_D = (x_1, y_1, x_2, y_2, x_3, y_3)$ and $z_D = (z_1, z_2, z_3)$ which represents the from the image of vertices for agiven triangle and their depths.
The final equation defines the servoing scheme used in this study.
The subsequent relation allows for the representation of the servoing scheme for a triangle.

$$\begin{pmatrix} \dot{x1} \\ \dot{y1} \\ \dot{x2} \\ \dot{y2} \\ \dot{x3} \\ \dot{y_3} \end{pmatrix} = \begin{pmatrix} L_{x1} \\ L_{y1} \\ L_{x2} \\ L_{y2} \\ L_{x3} \\ L_{y3} \end{pmatrix} \dot{c_D} = L\,\dot{c_D} \tag{4.1}$$

In this study, we compute the matrix L_D at the desired pose of the triangular primitive. This equation yields six independent equations that govern both the velocity and the angular rotation around the camera's axis.

4.3.2 Multi-feature servoing loop with triangular shapes

For a setup involving n triangles, a straightforward and conventional approach consists of arranging the interaction matrices row-wise and computing a unified correction for the camera as shown below.

$$\dot{c_D} = -\lambda_D \begin{pmatrix} L_1 \\ \vdots \\ L_n \end{pmatrix}^{+} \begin{pmatrix} e_1 \\ \vdots \\ e_n \end{pmatrix} \tag{4.2}$$

It has been demonstrated that this method does not yield satisfactory results [27]
In this work, a single correction per triangle has been proposed.

$$\dot{c}_i = -\lambda_D L_i^{+} e_i,\ i = 1, \cdots, n \tag{4.3}$$

An average correction is then computed from all the per-triangle corrections proposed.

$$\dot{c_D} = \frac{1}{n} \sum_{i=1}^{n} \dot{c}_i \tag{4.4}$$

Although the uniform method demonstrates convergence, it fails to preserve the reference visual features.
In contrast, the proposed approach effectively manages deformation and aligns the rigid camera pose with the most relevant deformed triangle.

We consider the most deformed triangle to be the most significant.
Therefore, we propose assigning weights to each triangle based on their level of deformation, which is related to the surface area of each deformed triangle in the mesh.

$$W_i = \frac{(a - a_0)}{a_0} \tag{4.5}$$

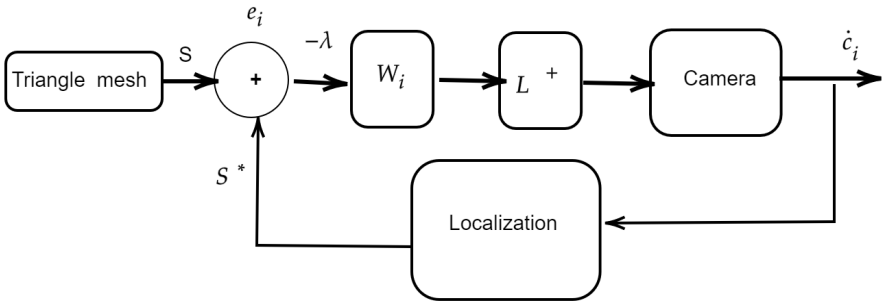

Figure 4.3: Block diagram of a visual servoing of traingle mesh

$$a(t) = |\frac{1}{2}\overrightarrow{P_1(t)P_2(t)} \bigwedge \overrightarrow{P_1(t)P_3(t)}| \tag{4.6}$$

Where a : Area of triangle when deformed.
a_0 : Area of triangle at rest.
In order to have smooth corrections it is necessary to have continuous weights [52].
This approach is justified, as the proposed weights are based on triangle areas, which vary continuously under deformation.
The global correction is then computed as follows:

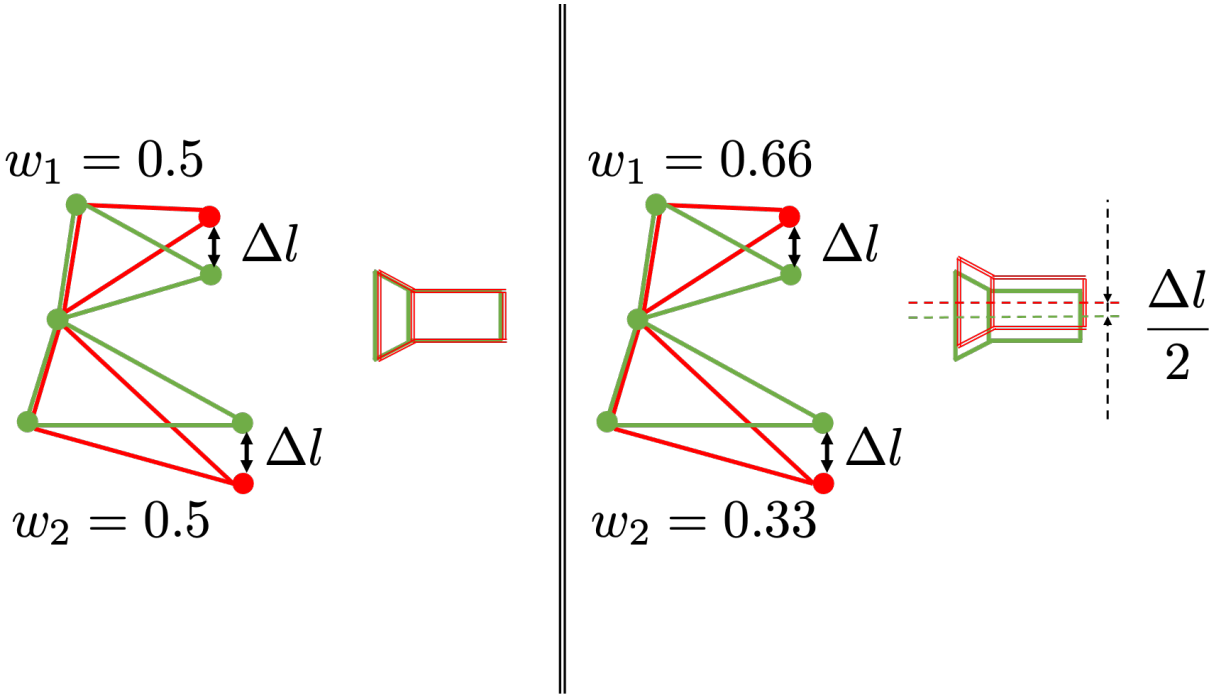

Figure 4.4: In this figure, a reference camera (in green) observes two reference triangles, also depicted in green.

Illustration of Weighting Effects In this figure, a reference camera (in green) observes two reference triangles, also depicted in green. Following deformation, the red vertices are displaced in opposite directions along the vertical axis by a distance of Δl, resulting in a non-rigid transformation of the two triangles. We analyze two scenarios:

Left Panel: Both triangles are assigned equal weights. In this situation, the correction computed using Equation 4.7 results in no motion. This outcome occurs because the opposing displacements cancel each other out due to equal weighting.

Right Panel: The upper triangle is assigned a weight twice that of the lower one. Here, Equation4.8 yields a vertical displacement equivalent to half of Δl. If the weight of the lower triangle (w_2) is set to zero, the resulting correction corresponds to a full vertical displacement of Δl upward.

$$\dot{c}_i = -\lambda W_i L_D^+ e_i \tag{4.7}$$

$$\nu = \frac{1}{\sum_i^n w_i} \sum_i^n \dot{c}_i \tag{4.8}$$

4.4 Results on simulated data

We investigate visual servoing on a deformable square grid composed of 121 points and 200 uniformly sized triangles, as illustrated in Figure4.2. In its undeformed (rest) state, the grid forms a square with each side measuring 1 meter. An elastic deformation is

applied to this triangular mesh along the z-axis, as depicted in the figure 4.2.The maximum deformation reaches 0.5meters at the top of the grid. The scene is observed by a camera with a focal length of 1500, positioned frontoparallel to the grid at a distance of 1.5 meters. The camera's motion is controlled across all six degrees of freedom, including both translational and rotational velocities v_x, v_y, v_z and w_x, w_y, w_z respectively). Two control strategies are evaluated: the uniform correction method [27] with equation 4.3 and the proposed weighted correction method 4.7. The uniform method fails to align the deformed view with the reference, as seen in Figure 4.7, where significant misalignment persists. The corresponding camera velocities (Figures 4.8 and 4.9) show uncoordinated translation and oscillatory rotation, leading to divergence and a pixel error peak of ~ 2400 pixels at iteration 70 (Figure 4.11). In contrast, the weighted method assigns importance based on deformation magnitude, resulting in a corrected trajectory that brings the deformed view (Figure 4.14, green) notably closer to the reference (red). The associated velocity profiles (Figures 4.15 and 4.16) are smooth and stable, with translation primarily vertical to compensate for z-axis deformation and minimal rotational adjustments. The pixel error (Figure 4.17) decreases steadily, stabilizing around 600 pixels within ~ 40 iterations. While a rigid camera cannot fully cancel non-rigid deformation, this approach achieves optimal alignment under the constraints. Figure 4.5 illustrates the 3D geometric relationship between reference and corrected trajectories, while Figures 4.6, 4.12, and 4.13 offer multiple perspectives on the object's deformation and alignment. Figure 4.4 schematically explains the weighting mechanism, demonstrating how differential weighting prevents motion cancellation when triangles deform in opposing directions. Overall, the proposed method significantly outperforms uniform correction by weighting visual features according to deformation relevance, enabling effective rigid-camera compensation for non-rigid object deformations. All simulations were conducted using MATLAB 2015a on an Intel Core i3 laptop with 4 GB of RAM.

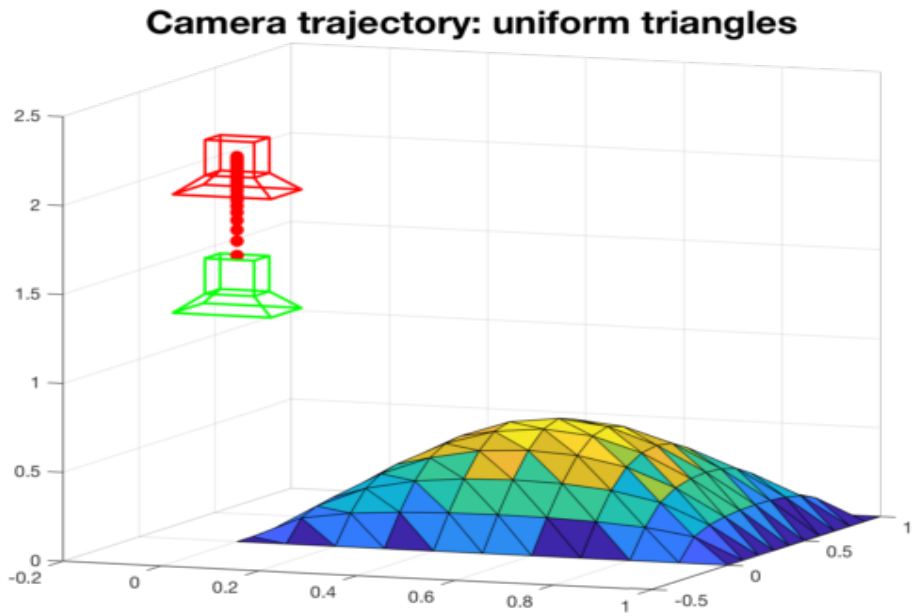

Figure 4.5: The results of Test (2) are presented. The green trajectory represents the reference camera path, while the red trajectory shows the corrected camera movement.

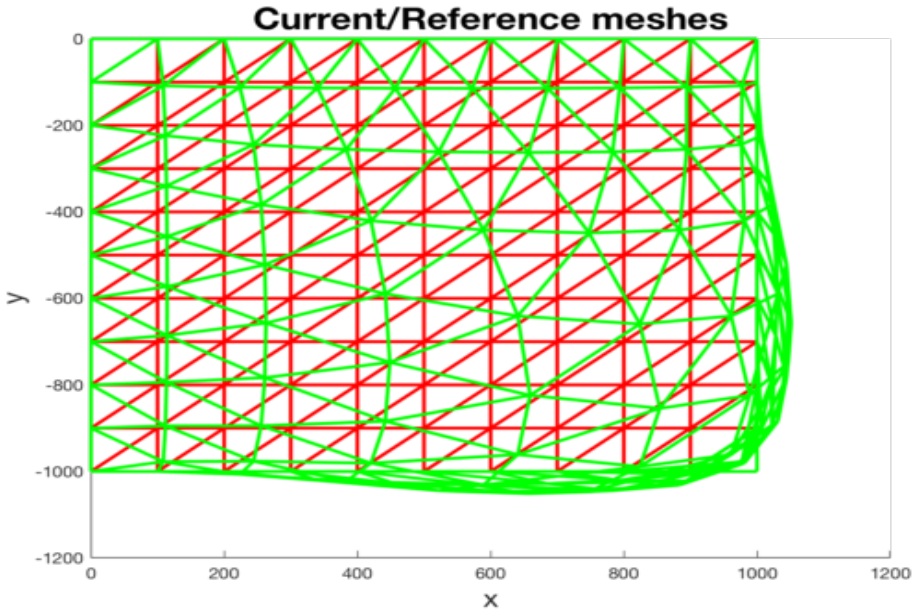

Figure 4.6: The reference object (in red) and the deformed object (in green) are displayed. The corrected trajectory aims to align the view of the object as closely as possible with the reference view.

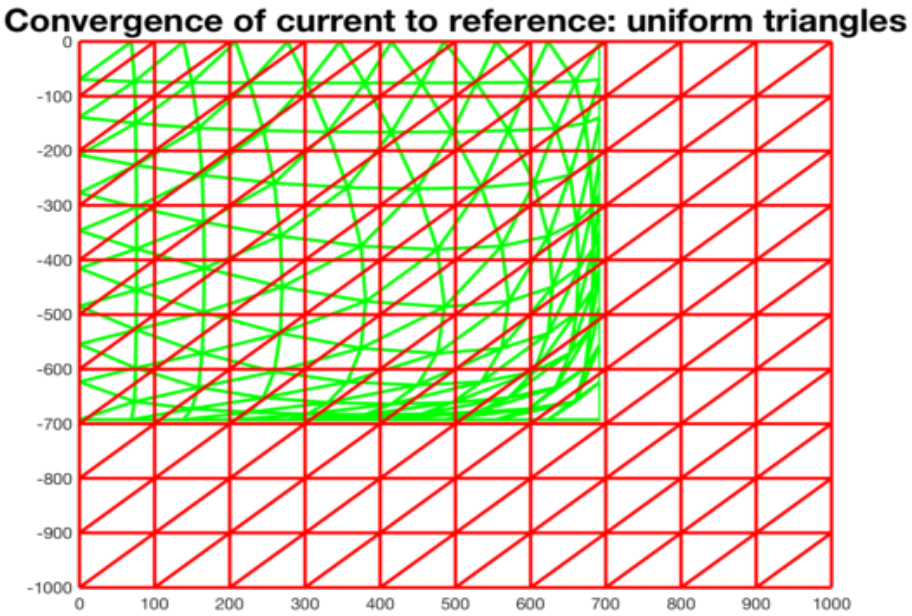

Figure 4.7: The reference object (in red) and the deformed object (in green) are shown. The corrected trajectory seeks to match the reference view of the object as closely as possible.

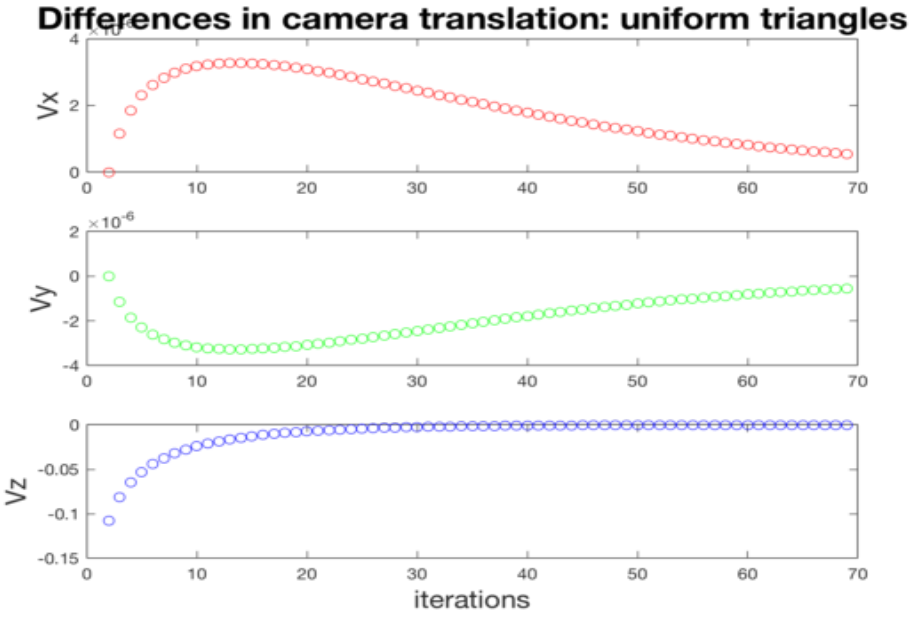

Figure 4.8: Test with L represents camera speed in translation.

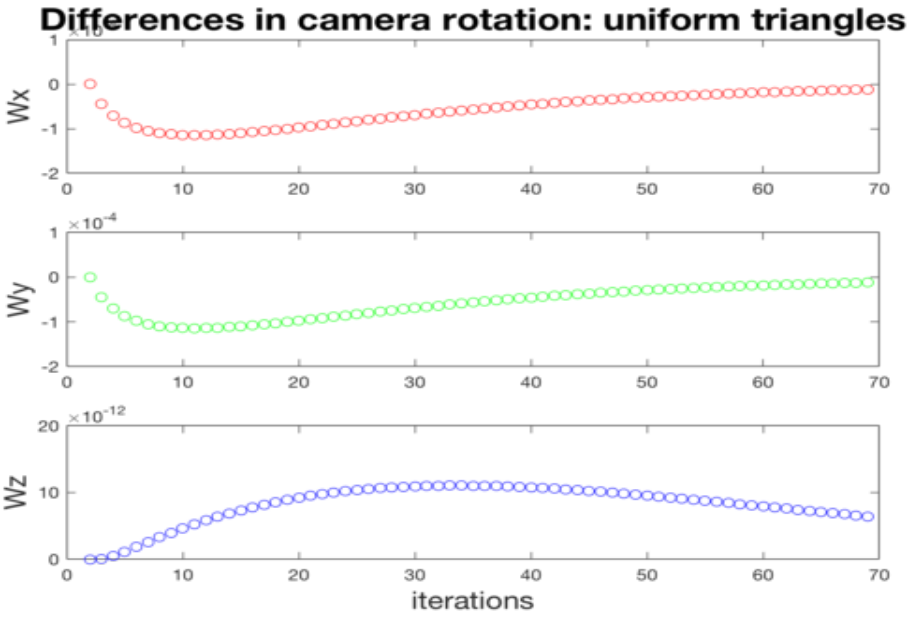

Figure 4.9: Test with L represents camera speed in rotation.

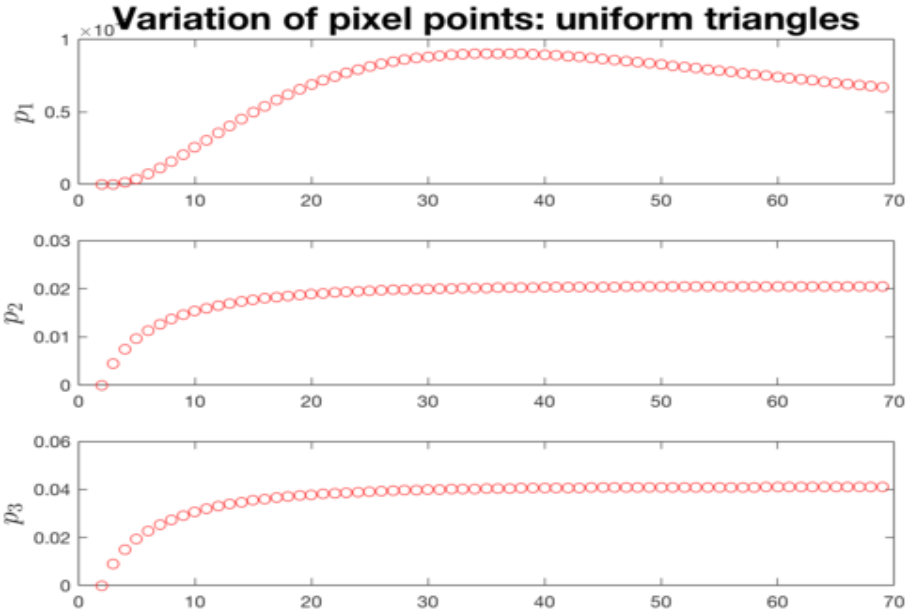

Figure 4.10: Variation of pixel error during visual servoing with uniform triangle weights

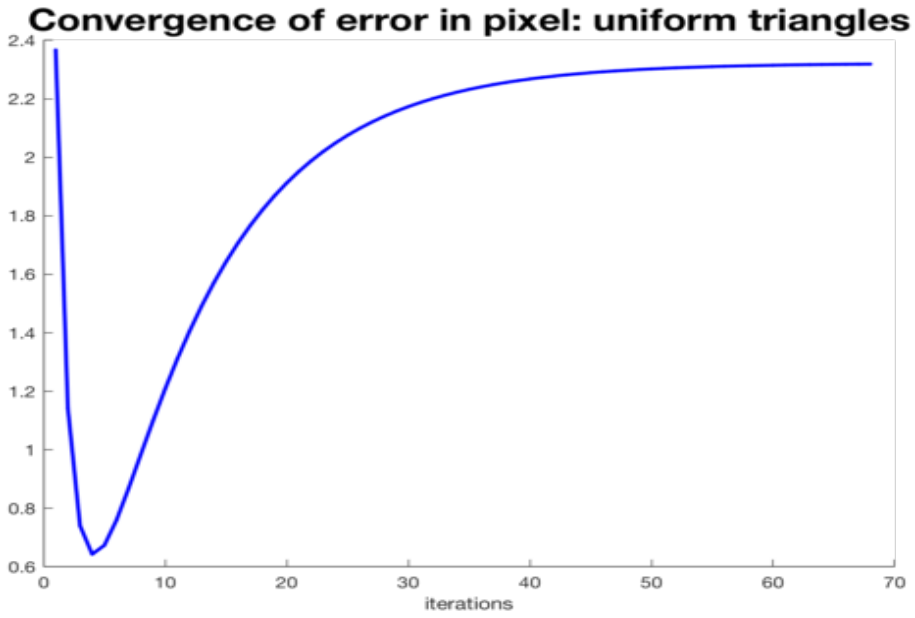

Figure 4.11: The results from applying visual servoing with elastic deformation on multiple triangles using equation 4.3 are shown. The global correction is calculated using uniform weighting.

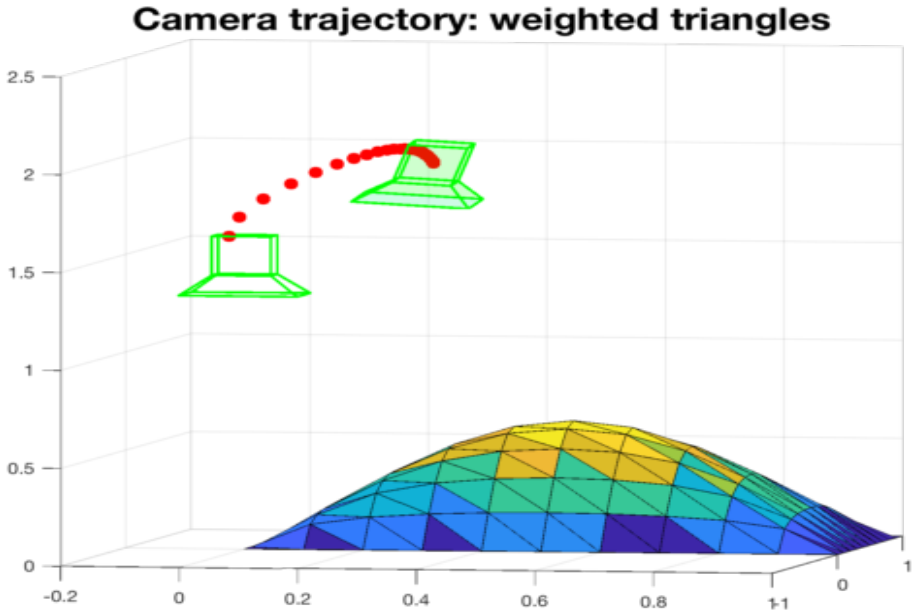

Figure 4.12: The reference viewed object is shown in red, while the deformed viewed object is depicted in green. The corrected trajectory aims to align as closely as possible with the reference view of the object.

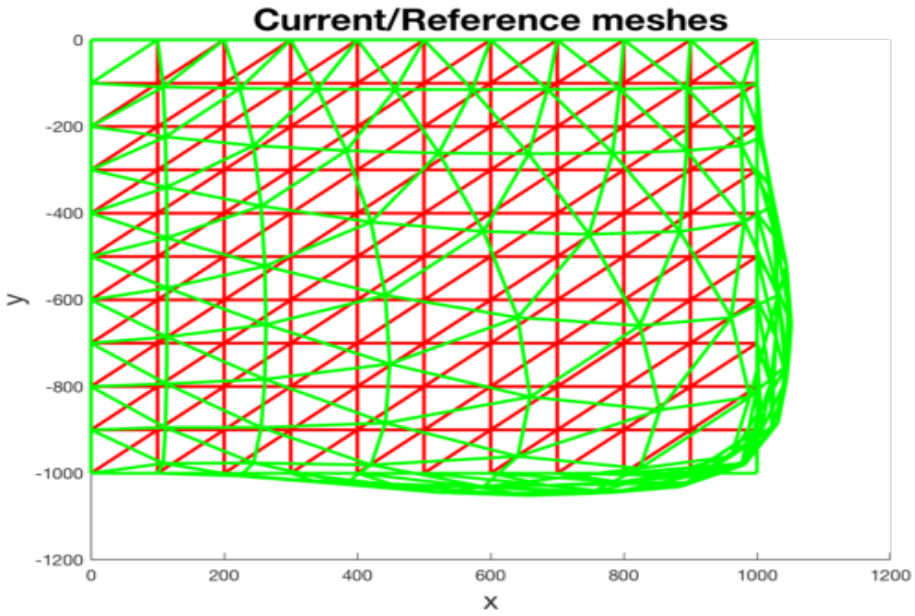

Figure 4.13: The reference object is displayed in red, and the deformed object is shown in green. The corrected trajectory strives to align the deformed view as closely as possible with the reference view.

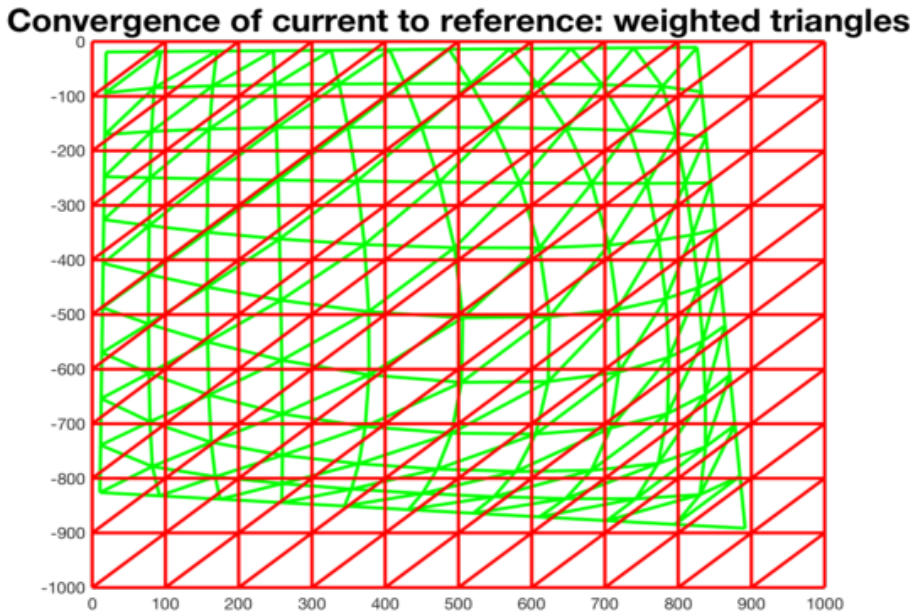

Figure 4.14: The object in the reference view is depicted in red, while the deformed object is shown in green. The corrected trajectory attempts to align the deformed object as closely as possible with the reference view.

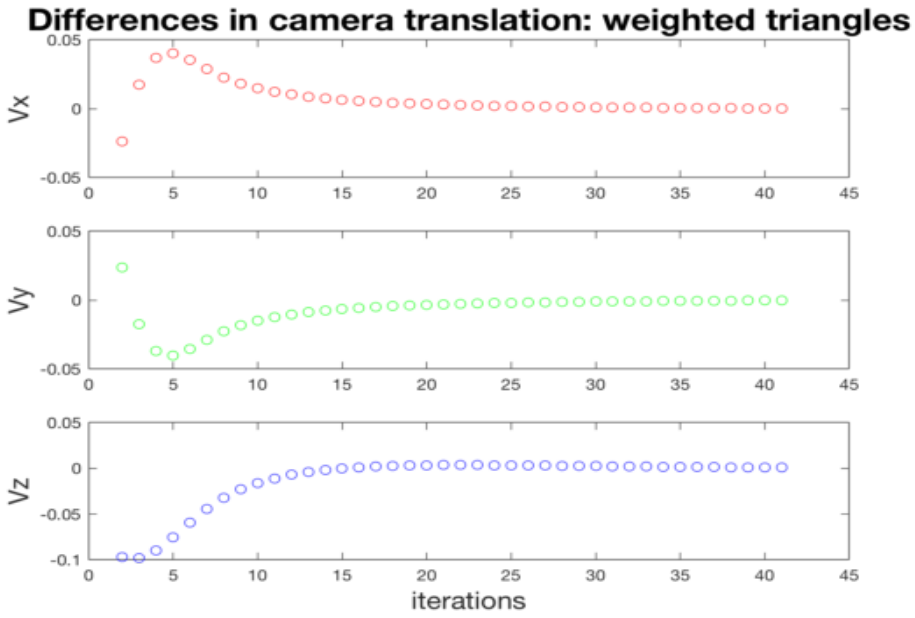

Figure 4.15: Test with L represents camera speed in translation.

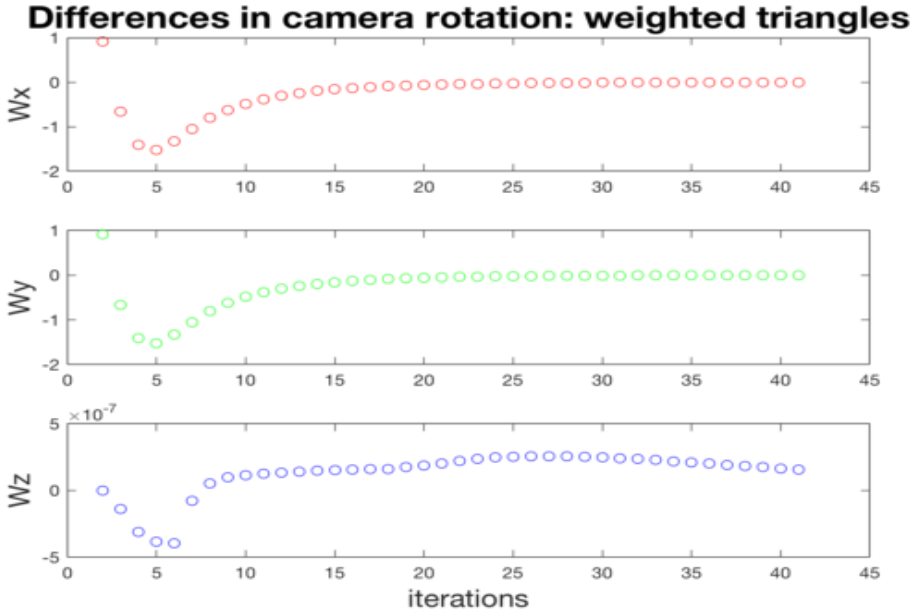

Figure 4.16: Test with L represents camera speed in rotaion.

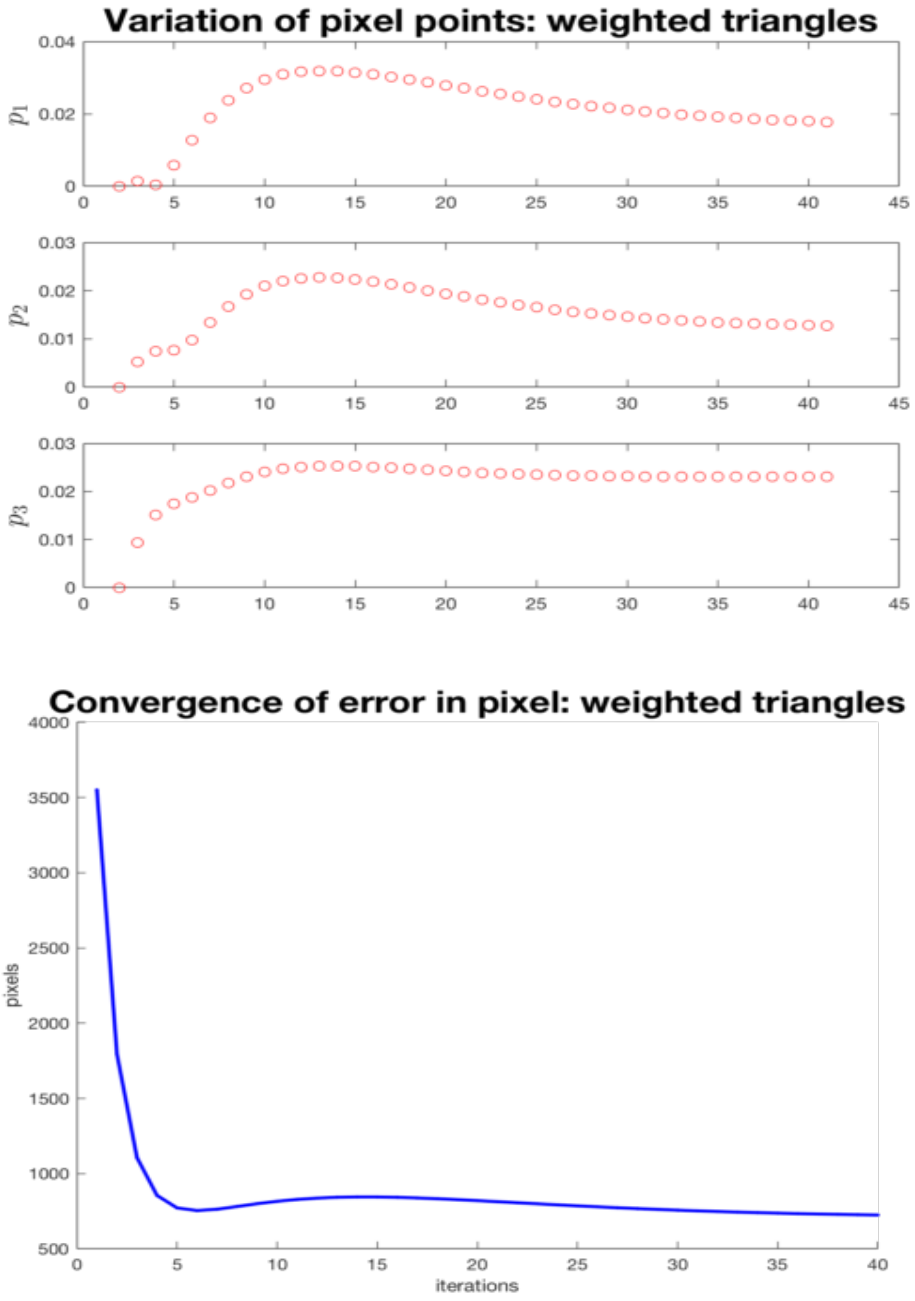

Figure 4.17: The outcomes of applying visual servoing based on elastic deformation to multiple triangles, as detailed in equation 4.5, are presented. The global correction is computed through area-weighted averaging

4.5 Conclusion

Visual servoing was applied to a non-rigid object modeled by a triangular mesh. Compared to uniform positioning, our approach resulted in a more precise camera correction, determined by averaging the scaling factors of all relative repositionings. A proportional gain, which adapts according to the area of each triangle, was implemented, alongside

a switching strategy that ensures smooth corrections throughout the trajectory. Experimental results demonstrated the camera's capability to adjust its configuration according to the planned geometric path. By averaging the scaling of all relative repositionings, we achieved more accurate camera corrections, effectively compensating for deformations and aligning closely with the reference view.

Chapter 5

Non-rigid object motion planning using a triangle mesh representation

5.1 Introduction

We offer a robust method for tracking object deformation via optimum visual servoing. We presume that a reference camera trajectory observing the targeted item has been preplanned. We will suppose that this item is characterized by a mesh made up of triangular geometric primitives.We make no assumptions about prior knowledge or information regarding the nature or mechanics of the deformation. In our study, the liver was used as a deformable reference object. The problem can be framed as follows: Consider a camera observing a reference trajectory while continuously tracking a live, deformable object. If the object undergoes deformation, how should the camera trajectory be adjusted according. In the context of laparoscopy, this is a common difficulty that is frequently assigned to an operator controls the camera (laparoscope) via a joystick, with guidance provided by the surgeon. Given the resolution of two significant difficulties, this task can be automated. First, a suitable liver tracking system that allows one the objective is to follow the 2D characteristics of the liver. Additionally, an automation criterion is introduced to adapt to deformations and adjust the camera trajectory accordingly.

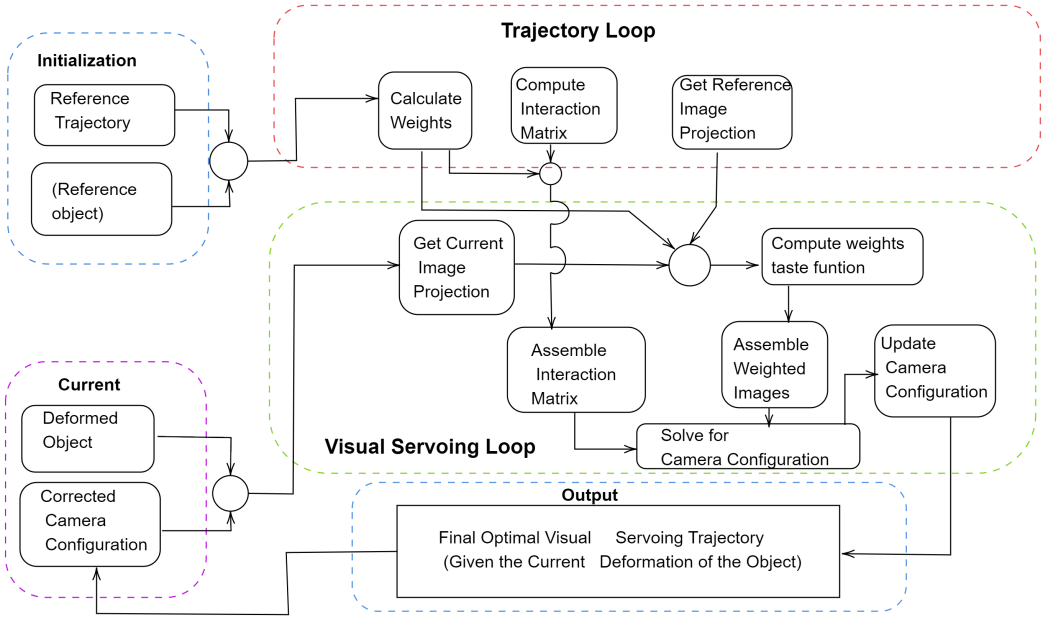

Figure 5.1: Flowchart of the triangle-mesh-based visual servoing algorithm for deformable objects.

This diagram 5.1 illustrates the flow of a triangle-mesh-based visual servoing algorithm used for controlling a camera's motion when observing a deformable object. The process begins by initializing the reference trajectory and mesh data, followed by iterating through each point on the trajectory. For each point, the algorithm processes each triangle in the object's mesh, calculating key parameters such as weights and interaction matrices. A visual servoing loop then iteratively adjusts the camera's position and orientation to minimize the error between the current image of the deformed object and the reference image, ultimately producing an optimized trajectory that accounts for the object's deformation.

5.2 Trajecotry-based global interaction matrix

We consider the triangle in the reference frame $\Re$ contain tree points $P_1(x_1, y_1), P_2(x_2, y_2), P_3(x_3, y_3)$ we represent the interaction matrix for each points.

$$\begin{pmatrix} \dot{x_1} \\ \dot{y_1} \\ \dot{x_2} \\ \dot{y_2} \\ \dot{x_3} \\ \dot{y_3} \end{pmatrix} = \begin{bmatrix} L_{x1} \\ L_{y1} \\ L_{x2} \\ L_{y2} \\ L_{x3} \\ L_{y3} \end{bmatrix} * \dot{c} \tag{5.1}$$

We consider the triangle with the greatest deformation as more relevant. To account for this, we propose weighting the triangles based on the amount of deformation they undergo W_i.

Incorporating weighting schemes into object localization

If the number of triangles is insufficient, the system may not provide reliable visual feedback or accurate corrections eq(5.18) If the number of triangles is too large, the system may become over-determined, potentially leading to numerical instability or increased computational complexity during the optimization and control processes (5.18) when the number of triangles becomes excessively high, the system may no longer have an exact solution, leading to what is known as an over-constrained localization problem. In such scenarios, standard localization techniques typically involve modeling both the triangle features and sensor observations as Gaussian random variables. The robot configuration is then estimated by identifying the one that maximizes the likelihood of the observed measurements.

Each triangle is assigned a weight based on the positional variance, as discussed in [23, 24]. In our approach, these weights are integrated directly into the specification of the motion control task. As a result, they influence the resolution of the linear system that governs the control strategy (5.18) We formulate a linear system of equations in which each equation is scaled by a corresponding positive real-valued weight w_i.

The significance of these coefficients will be clarified in a subsequent section. At this stage, they lead to the formulation of the following linear system:

$$W(c - c_0) = IM_i - IM_{0i}, \tag{5.2}$$

where W is matrix:

$$\begin{bmatrix} w_0 \dfrac{\partial \Pi_{C,T_0}}{\partial C}(c, l_0)(c - c_0) \\ . \\ . \\ w_m \dfrac{\partial \Pi_{C,T_m}}{\partial C}(c, l_m)(c - c_0) \end{bmatrix} \tag{5.3}$$

and

$$IM = \begin{pmatrix} w_1 im_1 \\ . \\ . \\ w_m im_m \end{pmatrix} \qquad IM_0 = \begin{pmatrix} w_1 im_{01} \\ . \\ . \\ w_m im_{0m} \end{pmatrix} \tag{5.4}$$

The optimal solution minimizing the residuals of this system, in the least-squares sense, is given by

$$c = c_0 + W^+(IM - IM_0), \tag{5.5}$$

where W^+ is the pseudo-inverse of W, it approximates, to first order, the minimization of the weighted sum of squared Euclidean distances between the expected images from configuration C and the actual observed image. im_i :

$$\| \sum_{i=0}^{m} w_i^2 \Pi_{C,T_i}(C) - im_i \|^2 \tag{5.6}$$

Camera localization on a triangle involves solving a system of equations that link the robot configuration to the observed images of the triangle. In practice, this system is often over-constrained, meaning there are more equations than unknowns, and solutions are found by minimizing the weighted sum of residuals. The choice of weights is crucial when the map used for localization deviates from reality, appropriate weighting can significantly improve accuracy. This sensitivity to weighting directly impacts the effectiveness of motion planning, as accurate localization is essential for reliable navigation and path execution.

5.3 Mesh-based path planning using triangular elements

Based on the preceding analysis, we introduce a motion model for the camera that is derived from a triangular mesh representation. Given a camera C, an environment with m traingle mesh $T_1, ..., T_m$, respectively associated to sensor C, a motion strategy based on triangular mesh representation is defined as follows:
(i) A reference trajectory is defined as the predefined path that the camera must follow,

typically aimed at positioning the camera in relation to the object or scene of interest.

$$\gamma : [0, U] \rightarrowtail C \tag{5.7}$$

$$u \rightarrowtail \gamma(u) \tag{5.8}$$

where $[0, U]$ is an interval.
(ii) w_i Continuous positive real-valued functions are mathematical functions that are defined on a given domain, where for every point in the domain, the function returns a positive real number. Additionally, these functions are continuous, meaning there are no abrupt jumps or breaks in their values within the domain $w1, ..., wm$:

$$w_i : [0, U] \rightarrowtail R^+ \tag{5.9}$$

$$u \rightarrowtail w_j(u) \tag{5.10}$$

such that $w_i\ (u) = 0$ for any u such that T_i is not visible by camera C when the camera is in configuration $\gamma(u)$. The triangle mesh-based motion for the mobile robot involves estimating the robot's current configuration within the closed-loop control task by solving a system of equations that accounts for the deformation of the object observed. The system leverages the geometric relationships between the robot's position, the deformed object, and the reference trajectory to calculate the necessary corrections. This approach ensures that the robot adapts to the real-time changes in the object deformation while maintaining the desired trajectory in a controlled manner system (eq.(5.20)) about the reference configuration $\gamma(u)$ and with values of the weights $w_i(u)$ for abscissa u along the trajectory.
Consider the triangle that undergoes greater deformation as being more significant. We suggest assigning weights to the triangles based on the amount of deformation. To ensure smooth corrections, it is essential to use continuous weights. This is achieved by the proposed weighting system, as the area of the triangle changes continuously during deformation. The global correction is then calculated as follows.

5.4 Visual servoing for triangle-mesh-based motion on non-rigid deformations

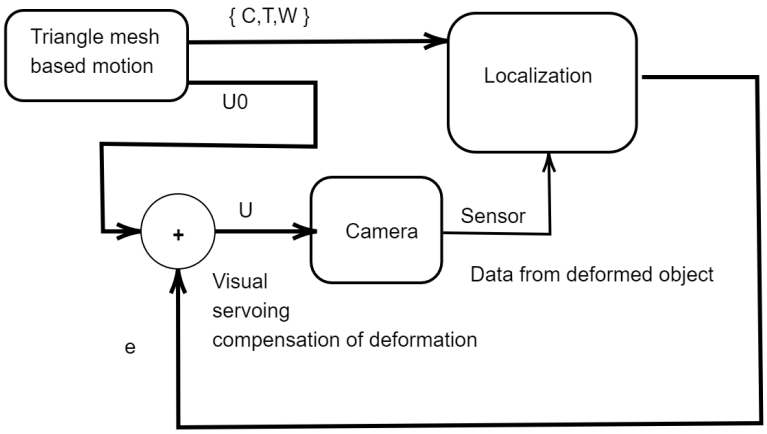

Figure 5.2: Optimal visual control following a Triangle-mesh based planned motion.

The control loop, represented by the feedback block in Figure 5.2 is feedback control law, adjusted through a set of weights, is used to correct errors in the robot configuration during localization. To ensure that corrections remain feasible and do not introduce instability, the weight function is constrained to be Lipschitz continuous. Without this constraint, corrections could become excessively large or erratic, making the current geometric path invalid and requiring replanning. However, the decision process for such replanning falls outside the scope of this work.

Triangle

Consider a triangle, which serves as the fundamental geometric unit for analyzing the system's behavior T as a basic geometric primitive, the triangle is composed of three vertices connected by edges, which serve as the fundamental elements for analysis in the system 3 points in the 3D workspace. Let $L = \mathbf{R}^9$ be the configuration space of this primitive. We denote by
$l = (X_1, X_2, X_3, Y_1, Y_2, Y_3, Z_1, Z_2, Z_3)^\top$ the configuration of T in L.
Where $(X_i, Y_i, Z_i)^\top$, $1 \leq i \leq 3$ the coordinates of the three vertices forming the triangle are denoted as T.Index i represents the number of the vertex in the considered triangle.

Camera "sensor"

Define C consider a camera that projects one or more triangular primitives onto a 2D feature in the image plane.
Let $C = \mathbf{SE}(3)$Let this represent the configuration space of the sensor.
We denote by c the spatial configuration of the camera C.

Triangle-based feature sensing

Detecting a triangle's features T by a camera C Can be represented through a continuous mapping:

$$\Pi : C \times T \to I_{c,l}, \tag{5.11}$$

$$(c, l) \mapsto \Pi(c, l), \tag{5.12}$$

Which associates each camera and triangle configuration with a corresponding feature in the image space $I_{c,l} \subset \mathbf{R}^6$.The image space is formed by projecting the three points that define the triangle of interest (see figure 4.2). Remark that Π is usually defined over a subset of $C \times T$ corresponding to configurations pairs of the camera and the triangle that must be in the field of view of the camera. In the case of a perspective camera, Π can be written as

$$\Pi(c, l) = \Big(x_1 = X_1/Z_1, \quad x_2 = X_2/Z_2, \quad x_3 = X_3/Z_3, \quad y_1 = Y_1/Z_1, \quad y_2 = Y_2/Z_2, \quad y_3 = Y_3/Z_3 \quad \Big)^\top \tag{5.13}$$

Joint localization and visual servoing framework

Equation-based approach to localization

Consider a camera positioned at a specific configuration $c \in C$. Let us consider m triangles $T_1, ..., T_m$of known configurations $l_1, ..., l_m \in L$, where each of them is visible from a camera C. Therefore, each pair (C, T_i) a given camera-triangle configuration leads to a system of localization equations:

$$\Pi(c, l_i) = im_i, \tag{5.14}$$

Where $im_i \in I_{c,l_i}$ is the projection of T_i in the image space. im_i is observed and therefore assumed to be known, and is denoted by $im_i = (x_1, x_2, x_3, y_1, y_2, y_3)^\top$. The knowledge of c and l_i will be discussed in the following section.

Reference-based visual servoing strategy

When the camera is required to follow a reference trajectory while observing a deformable reference object, the outcome derived from equation 5.14 It is assumed that the system

remains close to a reference configuration for both the camera and the object. By disregarding the higher-order terms of the linearization, equation 5.14 can be linearized as follows, providing a first-order approximation of the system's behavior around the reference configuration:

$$\frac{\partial \Pi}{\partial c}(c_0, l_{0i})(c - c_0) + \frac{\partial \Pi}{\partial l}(c_0, l_{0i})(l_i - l_{0i}) = im_i - im_{0i}. \tag{5.15}$$

Where c_0 denotes the reference configuration of the camera along the predefined trajectory l_{0i} represents the expected reference configuration of the observed triangle, with its associated reference features located at im_{0i} in the image space I_{c,l_i}. At execution time, both the reference camera and object configurations may differ from their runtime counterparts due to two main factors: camera drift and object deformation. These considerations are valid under the assumption that localization is performed at a high frequency faster than both the camera control loop and the dynamics of the object's deformation [56]. im_i and im_{0i}, respectively, represent the image of triangle T_i in sensor C and The expected image refers to the one that would be observed from configuration c_0, assuming the absence of both camera drift and object deformation. $\frac{\partial \Pi}{\partial c}$ It is the Jacobian matrix of order 6, which characterizes the variation of the image features with respect to changes in the camera configuration. This matrix is derived from the geometric relationship between the camera and the triangle and corresponds to the so-called interaction matrix in classical visual servoing terminology. $\frac{\partial \Pi}{\partial l}$ is the Jacobian matrix of order 6×9 represents the Jacobian that captures the variation of the image features with respect to changes in the triangle configuration. In the case of non-deforming objects, this Jacobian reduces to a zero matrix. Previous studies have typically modeled this matrix using either physics-based deformation models or data-driven approaches based on machine learning algorithms. [80], [30], [59]. In this work, we adopt an optimal visual control framework to integrate the control loop within the visual servoing algorithm. This formulation enables us to implicitly account for object deformation without requiring explicit numerical computation or regression of the deformation behavior. To begin, we examine the analytical formulation of the interaction matrix. It is well established that, for a single 3D point with coordinates (X, Y, Z) projected onto 2D image coordinates (x, y), the interaction matrix also known as the image Jacobian that relates infinitesimal variations in the camera pose to changes in the image coordinates is given by [16]:

$$L(x, y, Z) = \begin{bmatrix} L_u(x, y, Z) \\ L_v(x, y, Z) \end{bmatrix} = \begin{bmatrix} \frac{1}{Z} & 0 & \frac{-y}{Z} & -xy & 1 + x^2 & -y \\ 0 & \frac{1}{Z} & \frac{-y}{Z} & -(1 + y^2) & xy & x \end{bmatrix}. \tag{5.16}$$

The intrinsic parameters of the camera are assumed to be known, and the image coordinates are normalized accordingly to remove the effect of pixel scaling. Considering a triangle as the visual primitive T at reference coordinates $l_0 = (X_1^0, X_2^0, X_3^0, Y_1^0, Y_2^0, Y_3^0, Z_1^0, Z_2^0, Z_3^0)^\top$ (see for instance figure 4.2), the corresponding interaction matrix for the triangle is represented as

$$\frac{\partial \Pi}{\partial c}(c_0, l_{0i}) = \begin{bmatrix} L_u(x_1^{0i}, y_1^{0i}, Z_1^{0i}) \\ L_u(x_2^{0i}, y_2^{0i}, Z_2^{0i}) \\ L_u(x_3^{0i}, y_3^{0i}, Z_3^{0i}) \\ L_v(x_1^{0i}, y_1^{0i}, Z_1^{0i}) \\ L_v(x_2^{0i}, y_2^{0i}, Z_2^{0i})) \\ L_v((x_3^{0i}, y_3^{0i}, Z_3^{0i}) \end{bmatrix}, \tag{5.17}$$

Where $im^0 = (x_1^0, x_2^0, x_3^0, y_1^0, y_2^0, y_3^0, z_1^0, z_2^0, z_3^0)^\top$ The reference projection of the triangle onto the camera plane, with the focal length and camera center assumed to be known and accounted for, is provided. For further details, we refer the reader to the appendix A.1 for more details on the definition of c and $\frac{\partial \Pi}{\partial c}(c_0, l_{0i})$.

Rigid object servoing via weighted triangular features

If the object is considered rigid, then $l_i = l_{0i}$ for all the m triangles and equation 5.15 simplifies to:

$$\frac{\partial \Pi}{\partial c}(c_0, l_{0i})(c - c_0) = im_i - im_{0i}. \tag{5.18}$$

Traditional visual servoing methods for rigid objects involve determining the camera pose$c$ such an equation for the whole set of triangles.This equation defines and solves the local task of aligning the current triangle's view with the expected one. Additionally, positive real weights can be assigned to the triangles of the object that are relevant to the current task. The equation above is then modified as follows:

$$w_i \frac{\partial \Pi}{\partial c}(c_0, l_{0i})(c - c_0) = w_i(im_i - im_{0i}). \tag{5.19}$$

Here w_i is the assigned weight to triangle L_i. By taking into account the entire set of triangles that constitute the object, along with their respective weights, we formulate the following system of equations.

$$W(c - c_0) = IM - IM_0. \tag{5.20}$$

Where

$$W = \begin{bmatrix} w_0 \dfrac{\partial \Pi}{\partial c}(c, l_0) \\ \vdots \\ w_m \dfrac{\partial \Pi}{\partial C}(c, l_m) \end{bmatrix}, \tag{5.21}$$

$$IM = \begin{pmatrix} w_1 im_1 \\ \vdots \\ w_m im_m \end{pmatrix} \text{ and } IM_0 = \begin{pmatrix} w_1 im_{01} \\ \vdots \\ w_m im_{0m} \end{pmatrix} \tag{5.22}$$

To solve equation 5.20 in the case of a rigid object, several considerations can influence the estimation of the camera pose. These include noise in the measurement of triangle positions within the image and inaccuracies in the 3D object model obtained from scanning or prior modeling. Furthermore, if the number of triangles is too large, the resulting system of equations may become over-constrained, and equation 5.20 may not have an exact solution, making the servoing task over-constrained. On the other hand, if the system of equations is underdetermined, meaning there are infinitely many possible camera poses that satisfy it, a common approach is to compute the minimum-norm solution to identify the most plausible configuration [56]. In the following section, we present our third contribution: the integration of the weighted triangle-based visual servoing formalism into an optimal visual control framework specifically designed for non-rigid objects. This integration enables the computation of corrective camera motions required to follow a reference trajectory on a deformable target. Notably, the proposed method operates without relying on prior knowledge or any parametric deformation model of the object.

5.5 Optimal visual control for vision-based servoing

As part of our contribution, we have shown that visual servoing can be formulated as a quadratic optimization problem, enabling effective optimization of the object's deformation (eq. (5.28)), to control the camera path. Recall that a general optimization problem is defined as the process of finding x such that:[8]

$$\widehat{argmin} f_o(x) \tag{5.23}$$

$$\text{Subjet } f_c(x) \leq 0 \tag{5.24}$$

where $f_0(x)$ is the objective function and the inequality $f_c(x) \leq 0$his constraint is of paramount importance significantly outweighing the objective itself. A constrained quadratic programming (QP) problem can be formulated when:

$$f_o(X) = \frac{1}{2} X^T Q X + C^T X \tag{5.25}$$

$$f_c(X) = AX - B \tag{5.26}$$

Visaul error on mesh primitives on triangles.

$$err_i = \ e_v \tag{5.27}$$

Quadratic programming (QP) is particularly useful as it allows objectives to be expressed in terms of Euclidean norms. A common example is the cost function associated with a tracking control objective:

$$f_o(x) = \frac{1}{2}\|e_v + \alpha \dot{e_v}\|^2 \tag{5.28}$$

Here, α represents tunable gain parameters. It is important to note that this is a design choice, and $\alpha = 1$.

$$f_o(c) = \frac{1}{2}\|e_v + \dot{e_v}\|^2 \tag{5.29}$$

In the 3D reconstruction of deformable objects from monocular views, assuming that any deformed shape minimizes stretching and compressing energy has proven to be an effective approach [54, 53, 55, 13]. In our work, we do not aim to infer the 3D shape from the 2D view. Rather, our objective is to compensate for the deformation through camera motion. Accordingly, we define this motion as the one that minimizes the stretching and compressing energy. Using equation 5.29, this approach results in minimizing the deformation as observed through perspective projection, which can be formalized as follows:

$$\begin{aligned} \hat{c} =& \underset{\tilde{c}\in C}{\operatorname{argmin}} \frac{1}{2}\sum_{i=1}^{m}\left\|\frac{\partial \Pi}{\partial l}(c_0, l_{0i})(l_i - l_{0i})\right\|_2^2, \\ \text{s.t. } & \frac{\partial \Pi}{\partial l}(c_0, l_{0i})(l_i - l_{0i}) + \frac{\partial \Pi}{\partial c}(c_0, l_{0i})(\tilde{c} - c_0) = (im_i - im_{0i}). \end{aligned} \tag{5.30}$$

In our work, we aim to prioritize specific spatial deformations over others. This objective is formalized by assigning real positive weight values to triangular primitives. As a result, the previously defined criterion can be expressed as follows:

$$\begin{aligned} \hat{c} =& \underset{\tilde{c}\in C}{\operatorname{argmin}} \frac{1}{2}\sum_{i=1}^{m}\left\|w_i\frac{\partial \Pi}{\partial l}(c_0, l_{0i})(l_i - l_{0i})\right\|_2^2, \\ \text{s.t. } & w_i\frac{\partial \Pi}{\partial l}(c_0, l_{0i})(l_i - l_{0i}) = -w_i\frac{\partial \Pi}{\partial c}(c_0, l_{0i})(\tilde{c} - c_0) + w_i(im_i - im_{0i}). \end{aligned} \tag{5.31}$$

This criterion facilitates the estimation of a rigid camera motion c that minimizes the discrepancy between the reference and the current deformed shapes. The weights enable us to assign greater importance to specific regions, as shown in figure 4.4. The following expression is used to compute the correction to the camera pose.

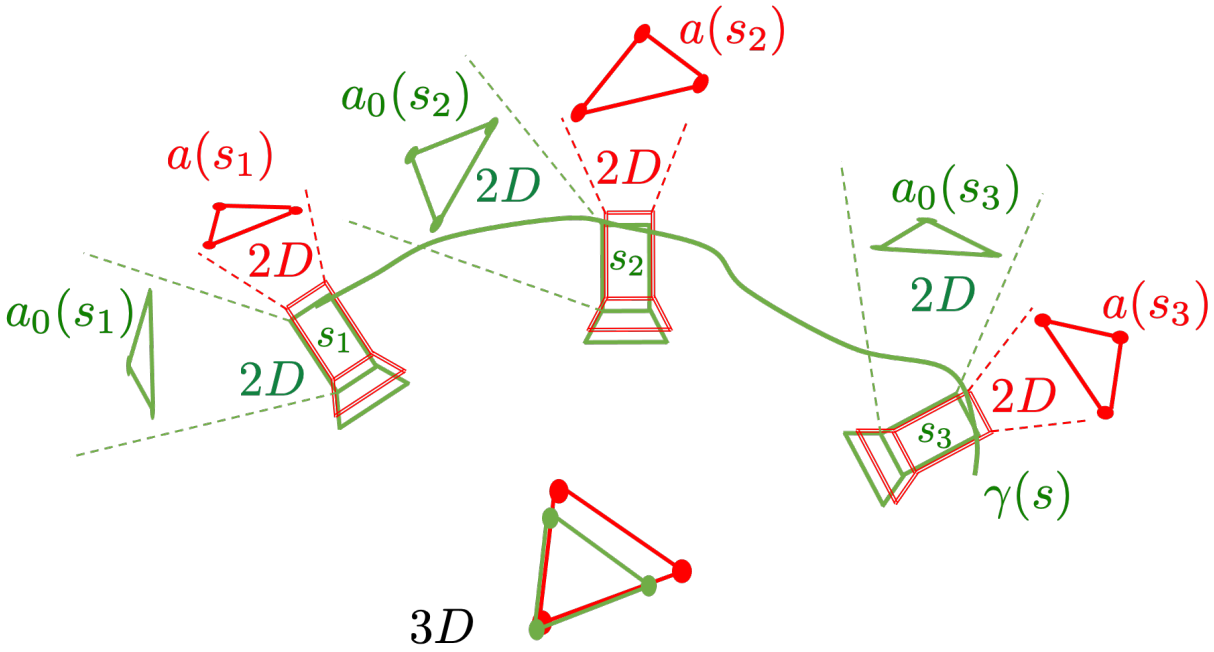

Figure 5.3: This figure provides an example of weight computation based on the area of the 2D projections of triangles. A reference camera, depicted in green, observes a reference triangle also shown in green along a predefined reference trajectory $\gamma(s)$, $0 \leq s \leq 1$. Following deformation, the deformed triangle is shown in red within the 3D space, with the corrected camera trajectory also depicted in red. Three sample points along the trajectory are chosen for analysis s_1, s_2 and s_3 Three pairs of views are presented, showing both the reference triangle from the reference camera and the deformed triangle from the corrected camera pose. As observed in the 2D image planes of both cameras, the area of the viewed triangles changes continuously along the trajectory. The weight is computed as follows $w(s) = a_0(s)\text{abs}(a_0(s) - a(s))/(1 + a_0(s))$.

Proposition 1 *Consider a camera placed in a reference configuration, where its position and orientation are defined relative to a fixed coordinate system $c_0 \in C$. The camera is viewing a deformed shape that is assumed to be at its minimal stretching/compressing energy. The rigid camera motion compensation $\hat{c}$, which satisfies criterion of equation* 5.31*, can be calculated as*

$$\hat{c} = c_0 + W^{+}(IM - IM_0). \tag{5.32}$$

W^{+} being the pseudo-inverse matrix of W. Taking into account equatio 5.31 and using the linearization equation 5.15, We express the optimal criterion as follows

$$\hat{c} = \underset{\tilde{c} \in C}{\text{argmin}} \frac{1}{2} \sum_{i=1}^{m} \left\| w_i \frac{\partial \Pi}{\partial c}(c_0, l_{0i})(c - c_0) - w_i(im_i - im_{0i}) \right\|_2^2, \tag{5.33}$$

This represents the weighted sum of the squares of the Euclidean norms of the differences between each image as seen from $c \in C$ and the image observed at runtime. This

minimization can be expressed in the following concatenated formula

$$\hat{c} = \underset{\tilde{c} \in C}{\operatorname{argmin}} \frac{1}{2} \| W(\tilde{c} - c_0) - (IM - IM_0) \|_2^2 \tag{5.34}$$

If W If the matrix has full column rank, solving for $c$in this weighted least-squares problem yields the following solution

$$\hat{c} = c_0 + (W^\top W)^{-1} W^\top (IM - IM_0). \tag{5.35}$$

In our work, we assume that, at every instant, there are sufficient visible triangles to compute the correction for the 6 degrees of freedom of the camera. To ensure this, we require at least two triangles to be visible, preventing ambiguous situations [55]. Given a mesh of N points and a camera tracking full regions of this mesh, it is reasonable to assume that this condition holds most of the time. In edge cases where this may not be the case (e.g., when the camera is too close to the object), we implement a safety test to verify the number of visible triangles, allowing us to compute the solution with minimum norm. If W has full row rank, the camera correction is computed as the minimum norm of the camera motion, satisfying the following criterion.

$$\hat{c} = \underset{\tilde{c} \in C}{\operatorname{argmin}} \frac{1}{2} \| \tilde{c} - c_0 \|_2 \text{ s.t. } W(\tilde{c} - c_0) = (IM - IM_0).$$

Given that W has full row rank, the solution to this under-determined least-squares problem is given by

$$\hat{c} = c_0 + W^\top (W W^\top)^{-1} (IM - IM_0).$$

In certain rare degenerate cases where W lacks full column rank and full row rank, the camera is not corrected from the reference configuration and we keep $\hat{c} = c_0$. The developments conducted in this section can be summarized as follows. Rigid compensation with camera motion of a nonrigid object motion consists of solving a system of equations establish a relationship between the camera configuration and the images of the object's triangles. In cases where the system is over-constrained, the solution involves determining a configuration that minimizes a weighted sum of residuals. When there is no deformation and the triangles remain unchanged, the choice of weights does not significantly affect the outcome (up to first-order approximation). However, as the object's triangles deform, the weight selection plays a critical role. Therefore, in our approach, we advocate for using these weights as a tool to plan motions based on the triangle mesh.

5.6 Triangle-mesh-based motion planning to compensate non-rigid deformations

We consider a scenario in which a camera follows a predefined trajectory while observing a deformable object modeled as a triangular mesh m traingles $T_1, ..., T_m$. A triangle mesh-based motion consists of two primary components:

(i) The first component is a reference trajectory, which defines the desired motion of the camera relative to the object

$$\gamma : [0, 1] \to C : \ s \mapsto \gamma(s), \tag{5.36}$$

Where $[0, 1]$ is the second component is a normalized interval of the trajectory temporal parameter, used to define the progression of motion over time s.

(ii) w these components are governed by continuous, positive, real-valued functions that modulate the motion over time$w_1, ..., w_m$:

$$w_i : [0, 1] \to R^+ : s \mapsto w_j(s), \tag{5.37}$$

such that $w_i\ (s) = 0$ for any s such that T_i is not visible by camera C when the each point along the trajectory, the camera is at configuration $\gamma(s)$. The continuity of w_i is essential to prevent undesirable discontinuities during the correction of the camera motion [56].

The triangle-mesh-based motion described above enables the compensation of the camera's rigid motion by correcting its current configuration within a closed-loop control framework. This correction is computed using the formulation presented in the referenced proposition 1 about the reference configuration $\gamma(s)$ and by assigning specific values to the weights $w_i(s)$ for abscissa s Along the trajectory, the correction formula can be defined for a specific abscissa s as:

1. If $W(s)$ Possesses full column rank

$$\hat{c}(s) = \gamma(s) + (W(s)^\top W(s))^{-1} W(s)^\top (IM(s) - IM_0(s)). \tag{5.38}$$

2. If $W(s)$ It does not have full column rank, but it has full row rank

$$\hat{c}(s) = \gamma(s) + W(s)^\top (W(s)\, W(s)^\top)^{-1} (IM(s) - IM_0(s)). \tag{5.39}$$

3. Else

$$\hat{c}(s) = \gamma(s). \tag{5.40}$$

Where $W(s)$, $IM(s)$ and $IM_0(s)$ are extensions of the notation used in the equations 5.21 and 5.22 when the reference camera setup follows a reference trajectory $c_0(s) = \gamma(s),\ s \in [0, 1]$.

The case 3 of equation 5.40 is an exceptional case, occurring rarely. We applied it as a safeguard correction in situations where Wdoes not possess full column rank or full row rank. The formulas provided above correspond to the corrected trajector $\hat{c}(s)$ from equations (5.38- 5.39-5.40) may be understood as the configuration that ensures optimal visual alignment, representing the minimum value of the integral, evaluated across the entire trajectory, of the difference between the reference view and the current deformed view, as shown below:

$$J = \int_0^1 \frac{1}{2} \| W(c(s) - \gamma(s)) - (IM(s) - IM_0(s)) \|_2^2 \, ds. \tag{5.41}$$

Continuous weights are crucial to achieving smooth corrections [56]. This occurs, for example, when the weights are proportional to the 2D area of the viewed triangle in the image plane. If the object is an organ and the goal is to compensate for deformations observed through a laparoscope, the weights could ideally be set manually by a technician prior to surgery. In this case, the surgeon, with the technician's assistance, The primary target area of the organ that must remain stable when viewed by the laparoscope can be identified. The technician can then assign greater weights to these areas compared to others. This semi-automatic process can be performed during the surgical planning phase.

5.7 Proposed algorithm and implementation details

The visual servoing algorithm 1 described in this section uses a deformable object's mesh to achieve precise camera positioning relative to the object, even as the object deforms. The key steps are:

1. The trajectory loop (lines 5:27) iterates through the reference camera trajectory, and for each camera pose, the algorithm computes:
   (a) for each triangle, using the reference object, the weights, the reference image projection, and the interaction matrix;
   (b) an overall weighted interaction matrix and reference and current object weighted images;

2. The visual servoing loop (lines 17:26) then iteratively solves for the optimal camera pose update that minimizes the difference between the reference and current object weighted images, updating the camera pose accordingly.

Algorithm 1 Triangle-Mesh-Based Motion for Visual Servoing on Deformable Object

Require: Reference trajectory $\gamma : [0,1] \to C : s \mapsto \gamma(s)$;
Require: Deformable object meshed with m triangles $T_1, \ldots, T_m$;
Require: step_size for sampling reference trajectory;
Require: max_iter for maximum number of iterations;
Require: ϵ_{im} stopping error on image difference;
1: **Initialization:**
2: Load reference trajectory γ for the camera;
3: Load data mesh of the reference object;
4: Load data mesh of the same object in its deformed state;
5: **Trajectory Loop:**
6: **for** $s \in [0,1]$ **with** step_size **do**
7: Get current camera configuration $c(s) \leftarrow \gamma(s)$;
8: **Triangle-Mesh Loop:**
9: **for** each triangle i from 1 to m **do**
10: Get current weight w_i by applying formulas in equations (B14) and (B15);
11: Get current interaction matrix by applying formula of eq (7);
12: Get the reference image projection im_i^0 of the triangular primitive;
13: Compute the current task function of the triangle using eq (10);
14: **end for**
15: $error \leftarrow \epsilon_{im} + 1$;
16: $iter \leftarrow 0$;
17: **Visual Servoing Loop:**
18: **while** $iter < max\_iter$ **or** $error > \epsilon_{im}$ **do**
19: Get the current image projection primitive im_i from the current camera pose;
20: Assemble object weighted interaction matrix W using equation (11);
21: Assemble reference and current object weighted image IM, IM_0 using equation (12);
22: Solve the optimal servoing function using equations (24), (25), (26);
23: Use the obtained $\hat{c}(s)$ to update the current camera configuration;
24: $iter \leftarrow iter + 1$;
25: $error \leftarrow \|W(\hat{c}(s) - c(s)) - (IM - IM_0)\|_2$;
26: **end while**
27: **end for**
28: **Output:**
29: Final optimal visual servoing trajectory given the current deformation of the object;

This approach allows the visual servoing to adapt to deformations of the object, using the mesh representation to correct the camera trajectory accordingly. This algorithm was implemented using Matlab version 2020b. It was run on a lapatop computer with an

Intel Core i7-8750H processor and 16 GB of RAM. The object model used was a triangle mesh with 10,000 vertices and 20,000 faces. The reference trajecotry was sampled at a step_size of 10 Hz. The maximum number of iterations, max_iter, was set to 100, and the stopping error, ϵ_{im}, was set to 0.01 pixels. The overall algorithm runs at an average speed of 40 Hz, enabling real-time visual servoing of the deformable object. This algorithm was tested across various scenarios, including multiple object types, reference trajectories, deformation modes, and levels of image noise. The following experimental section describes the results obtained in these diverse settings.

5.8 Simulated data results

This section presents representative examples of our verification tests, performed through simulations with two deformable objects: a planar object deformed into a bump-like shape and a model of the human liver, deformed at the left and right lobes. We conducted experiments involving both linear and nonlinear deformations, applying our method to two types of geometric trajectories: linear and circular. To evaluate the robustness of our approach, noise detection was simulated on the 2D mesh vertices. A perspective camera with a focal length of was utilized for these simulations 1500. We ran our simulations on core i3 lap-top with 4Go RAM and Matlab2015a. In the shown figures, distances are in meters, translation speeds of the camera is in meter per second and rotation speeds are in radian per seconds. The errors in image space are in pixels.

5.8.1 Planar Object Control Along a Linear Geometric Path

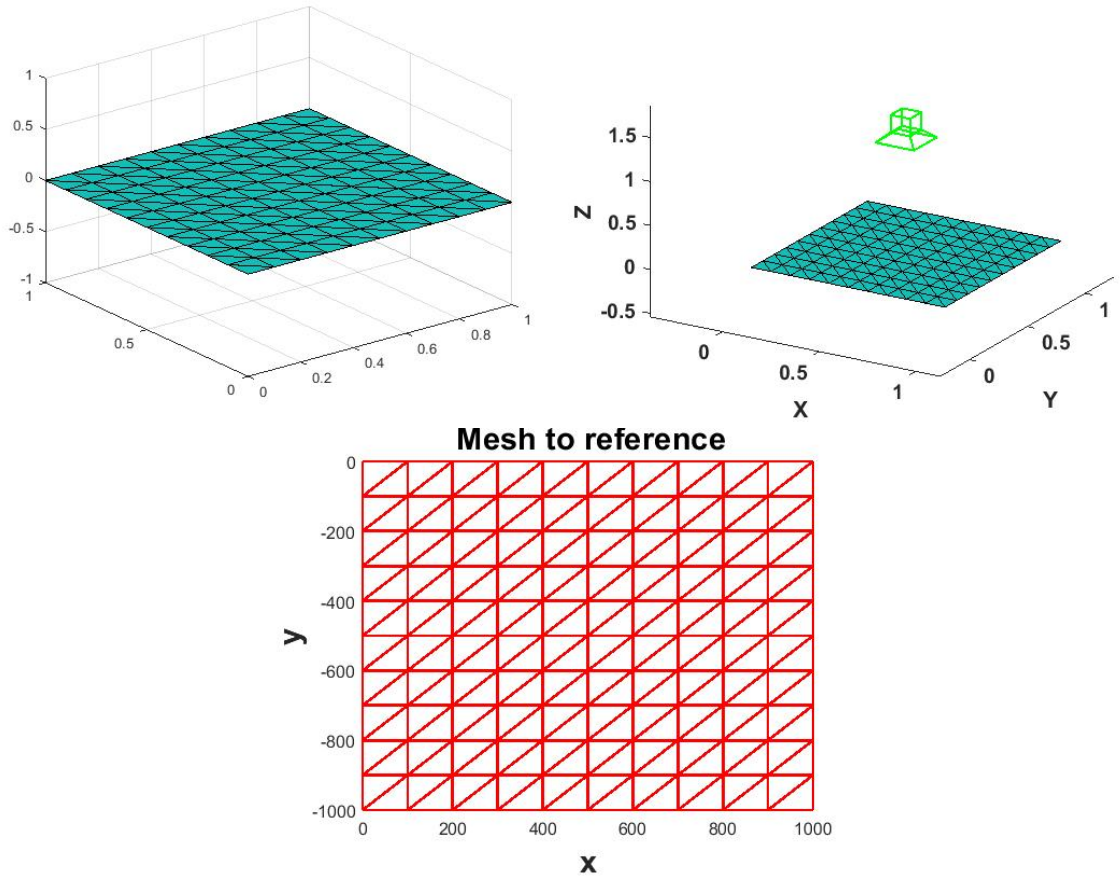

Figure 5.4: Top right: The reference object is represented as a 3D planar triangular mesh. Top left: The reference object, along with the camera configuration, is displayed in 3D space. Bottom: The view of the planar mesh as seen on the camera plane.

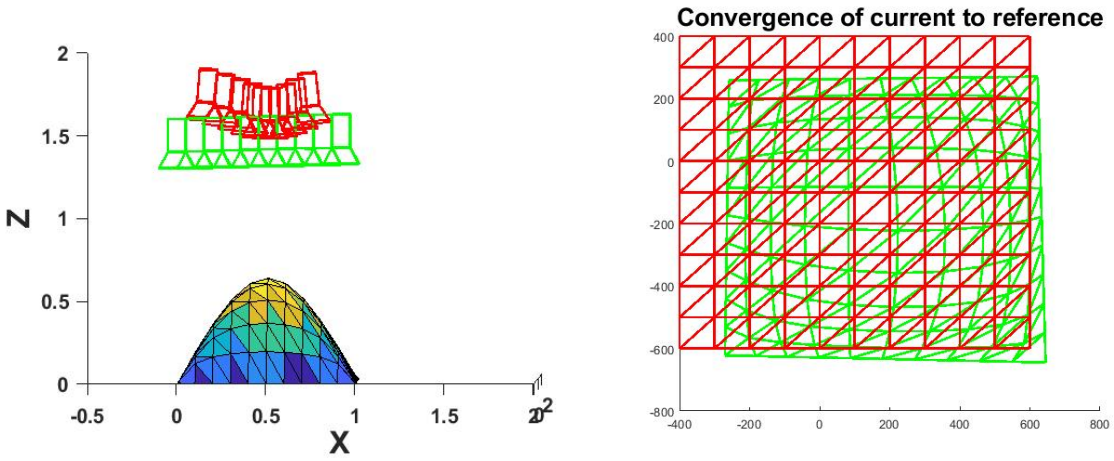

Figure 5.5: Left: The reference object in its deformed state, with the camera trajectories represented in green (reference) and red (deformed). The reference camera follows a pre-defined linear path, with the green trajectory indicating the reference and the red trajectory representing the corrected one. All weights are set to one. Right: The reference object (red) and the deformed object (green) as seen from the camera. The goal of the corrected trajectory is to closely match the reference view of the object.

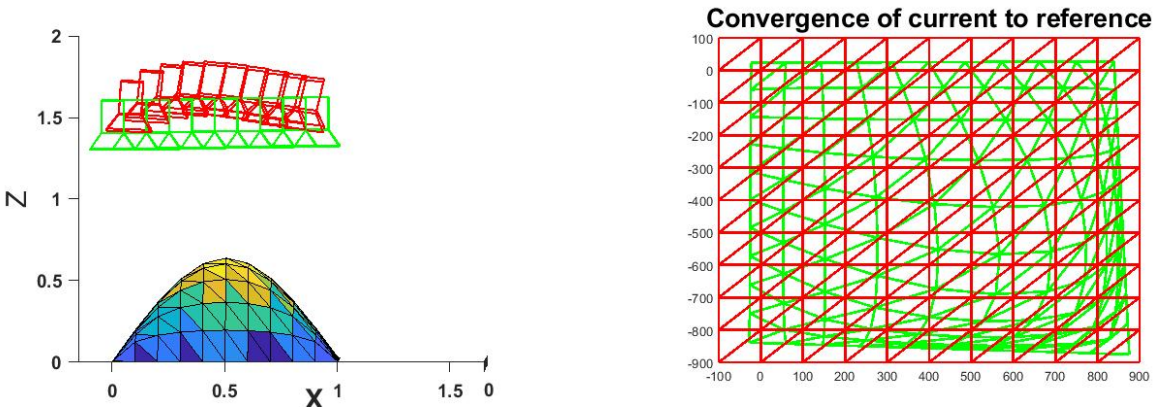

Figure 5.6: Left: The deformed 3D planar object with the camera following a linear planned trajectory. The green camera represents the reference trajectory, while the red camera indicates the corrected trajectory. In this case, all weights are set to one, except for the weights associated with the triangles at the top of the hill (yellow triangles), which are set to 4. Right: The reference object (in red) and the deformed object (in green) as viewed by the camera. The corrected trajectory aims to align the camera's view as closely as possible with the reference view of the object.

Figure 5.4 the object is represented in 3D space along with its projection onto the camera plane. Both the object and its deformation are depicted using a triangular mesh, as shown in the figure5.5. Throughout the deformation process, the camera path is determined by the constraints imposed on the camera, ensuring that the target features remain within

and dynamically adjust to the planned trajectory. This is demonstrated in the figures 5.6 the camera movement is controlled based on the weights along the path, with particular emphasis on the triangles that experience the greatest deformation In figures 5.5 and 5.6, the yellow, orange, green, and blue regions represent the deformation heatmap, with deformation intensity decreasing from high to low. In this experiment, constant weights are applied, with the yellow region assigned the highest weight of 4, while the other regions are given a weight of 1. There are no fixed limits for the weight values; only the relative size of the weights influences the tracking focus. Regions of greater importance during runtime are allocated higher weight values to enhance the tracking of the most deformed areas.

5.8.2 Case study: The liver as organ of interest

The liver, being the largest organ in the human body and a key component of the digestive system, carries out a variety of crucial functions. In this study, we utilize a liver mesh to assess the performance of our method. Taking the result of Saidi's et al work[78]. They deformed the liver in the superior medial segment while fixing the lower posterior segment. The liver mesh contains 130382 triangle and 12226 vertex see figure 5.7. We have two types of deformation for the same mesh of the liver, we moved the selected deformed part of 6 mm in the z-axis, which is considered as a linear deformation, and another deformation, we moved 825 vetrex exists on the liver superior medial by 22 mm, we consider as a non-linear deformation.

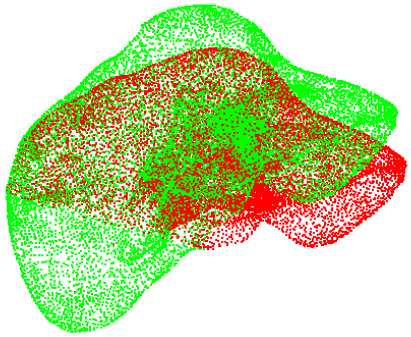

Figure 5.7: liver before and after elastic deformation.

5.8.3 Results on linear deformation and linear trajectory

For the linear deformation, a command was used to control the camera movement, with its trajectory following a horizontal straight line, as illustrated in the figure 5.7. This figure shows the difference between two tests performed where the first test is for a deformation with a weight equal to 1 ($w_i(t) = 1$) for the whole triangular mesh (we can

called test 1) it is represented in the left figure. The second test is for giving an important weight $w_i(t)$ for the deformed part of the triangular mesh (we can called test 2) it is represented in the right figure. The results demonstrate that the camera successfully tracked the deformation of the liver while maintaining the specified trajectory. Figure 5.8 show the trace of the deformed liver network in the striatum. Figure 5.9 show he analysis of camera control performance during the dynamic tracking of an object in a similar environment They show the translation and rotation of the camera during the tracking of the liver malformation.

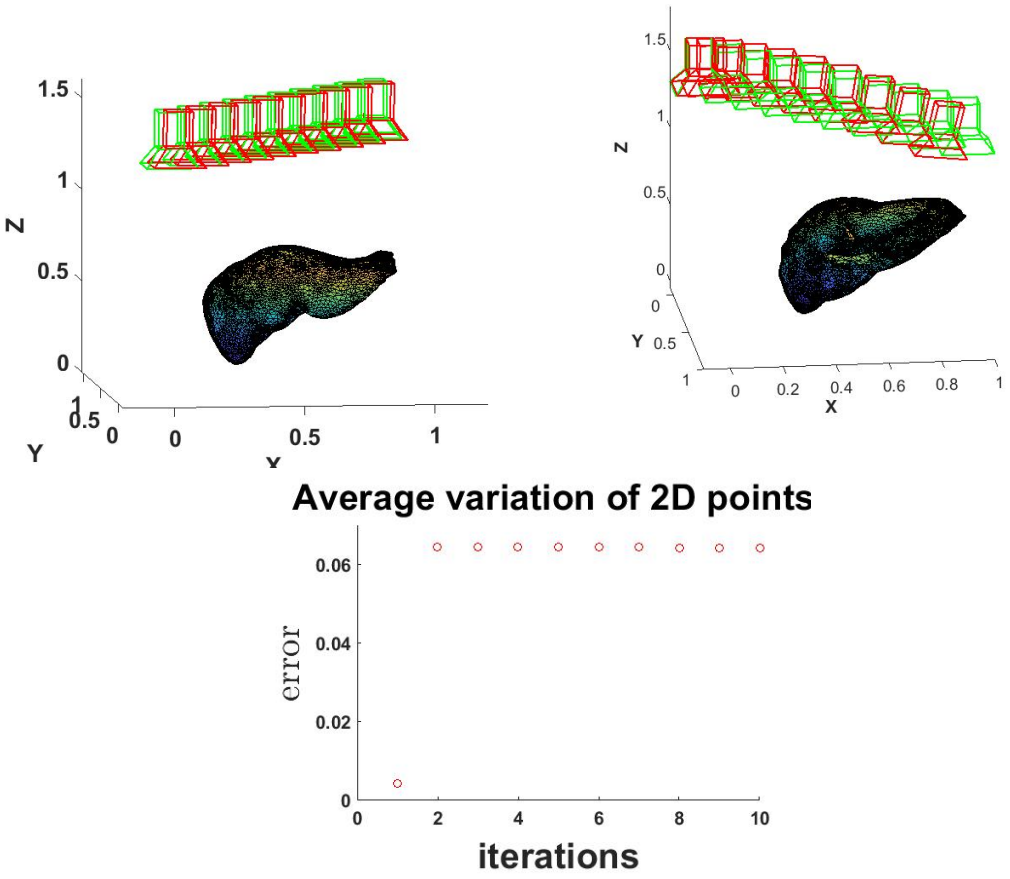

Figure 5.8: The liver undergoes linear elastic deformation while the camera follows a reference straight trajectory. The results of test (1) are presented. On the left, the reference trajectory is shown, while on the right, a side view of the corrected trajectory is displayed. The green cameras represent the reference trajectory, and the red cameras show the corrected trajectory. At the bottom, the variation in pixel points during the camera pose correction is illustrated, with the green cameras indicating the reference trajectory.

5.8.4 Results on nonlinear deformation and linear trajectory

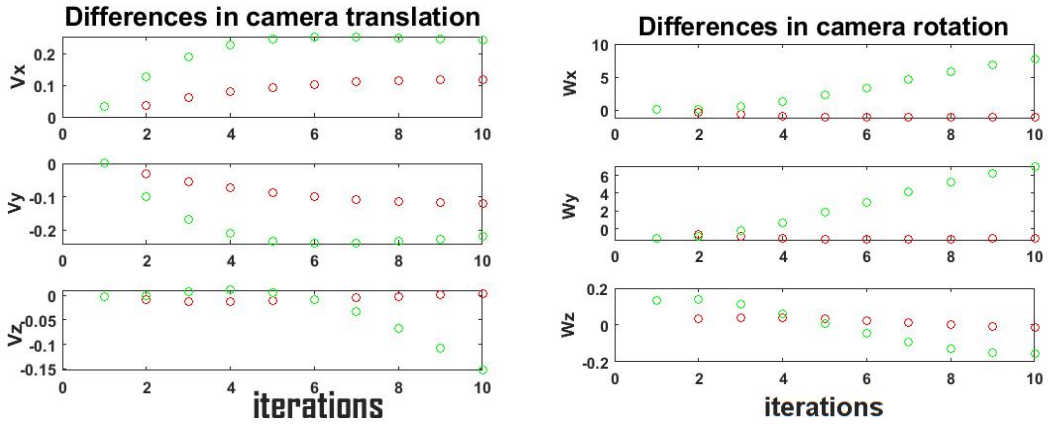

Figure 5.9: The performance analysis of camera speed during the tracking of a deformed object with linear deformation is shown. The left figure illustrates the camera speed in translation, while the right figure shows the camera speed in rotation. Red dots represent the camera speed according to the strategy used in test (1), while green dots represent the camera speed based on the strategy of test (2). ω_y and ω_z are similar for both tests.

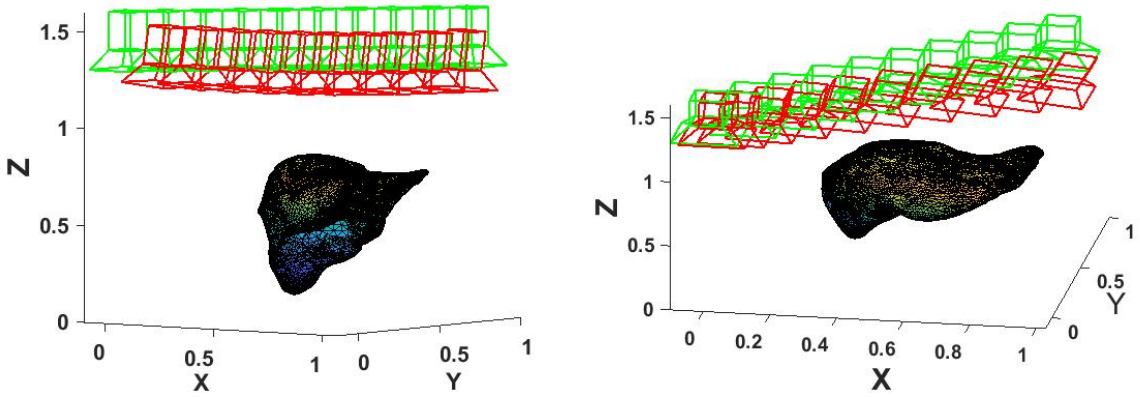

Figure 5.10: The liver undergoes nonlinear elastic deformation while the camera follows a reference straight trajectory. The results of test (2) are presented. The left figure shows a side view, while the right figure shows another side view. Green cameras represent the reference trajectory, and red cameras represent the corrected trajectory.

In this experiment, we applied the same configurations for test (1) and test (2) as previously described. This time, we examine a nonlinear deformation with a maximum displacement of 22 mm in the left liver lobe, as illustrated in Figure 5.7. The camera trajectory follows a straight horizontal line, shown in green, with the corrected trajectory depicted in red, relative to the reference view. Figure 5.10-left presents the corrected camera trajectory in red when all weights are set to 1. Figure 5.10-right illustrates the corrected camera trajectory in red when the most deformed regions have a weight of 5, while the least deformed or non-deformed regions are assigned a weight of 1. The

curvature of the corrected trajectory is more noticeable in test (2) (right figure). The nonlinear deformation of the triangles in the image is kept as close as possible to their areas in the reference image. The weight functions enable us to prioritize the most significantly deformed triangles from the liver and adjust the correction accordingly. As shown in Figure 5.10 on the right, the camera path is more significantly affected, aligning with the nonlinear deformation of the triangles.

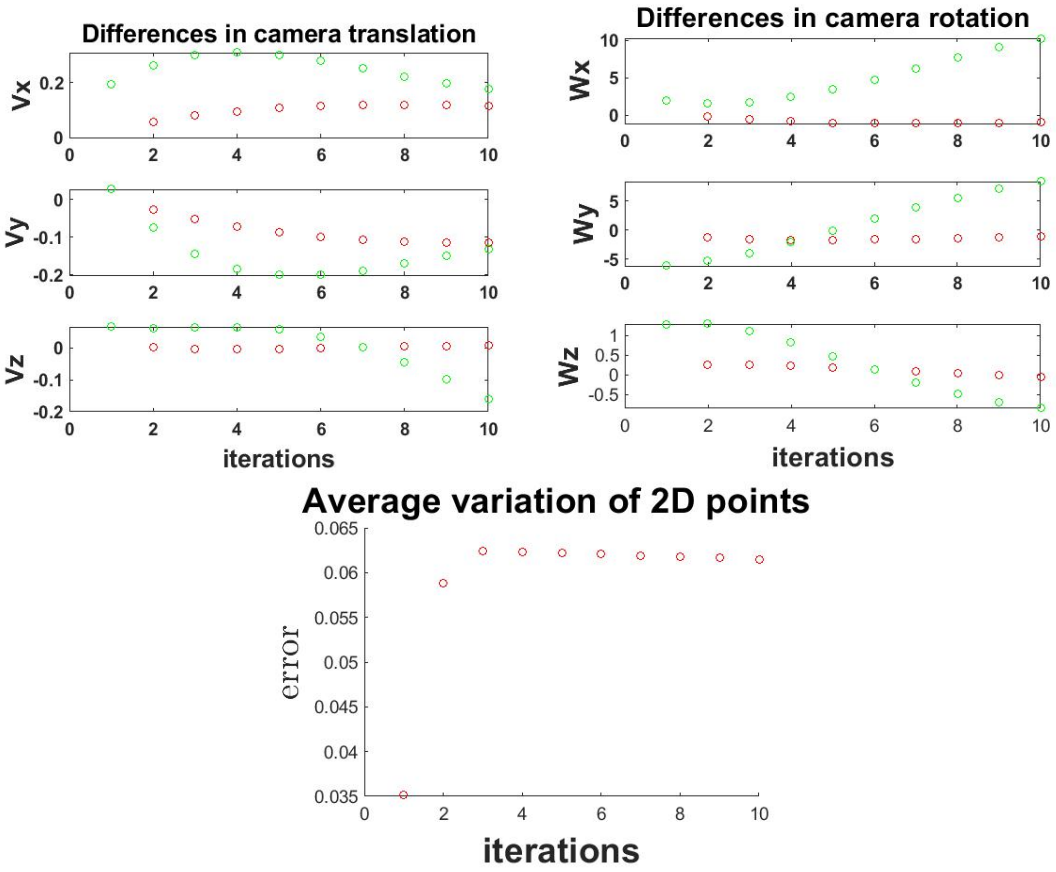

Figure 5.11: An analysis of the camera speed performance when tracking a deformed object with nonlinear deformation is shown. The left figure illustrates the camera speed in translation, while the right figure shows the camera speed in rotation. Red dots represent the camera speed according to the strategy used in test (1), and green dots represent the camera speed following the strategy of test (2).

It can be observed that the convergence speed is relatively fast for this approach (after 10 iterations). A similar configuration was also applied to the liver triangular mesh, and the resulting control converged as well, as shown in the figure 5.11. Note the control of the camera during the tracking trajectory, the correction of the translation and rotation of the camera, and the variation of the pixel points. We observe a difference speed between translation, rotation and pixel points on both figures 5.11.

5.8.5 Nonlinear Deformation and Trajectory Tracking Results

A test was conducted on a semi-circular trajectory plane for a nonlinear trajectory, using the following parameters: $\theta = (0 : 0.01 : 2 * \pi)$, $R = 1$, $c_x = R * cos(\theta)$, $c_y = R * cos(\theta)$ which are used in the equation eq.(5.36) as is showed in the figure 5.12 in the left. Our command was tested in two phases. In the first phase, we applied a constant weight of 1 to all triangles. The camera successfully followed the trajectory, as shown on the right side of Figure 5.12. In the second phase, we repeated the test with the same command and trajectory but modified the triangle weights, giving greater emphasis to the deformed triangles by assigning weights ranging from 1 to 5, as depicted in the figure 5.13. The result obtained shows us that the second test is better than the first one. In the first test, the camera followed the trajectory but the first test, the deformation was not adequately captured. In contrast, the second test successfully tracked the trajectory, capturing the deformation more accurately. Additionally, the results for camera speed, both in translation and rotation, showed improvements over the first test, as illustrated in Figures 5.17. We also noted a difference in pixel variation: the first test exhibited a variation between 0.06 and 0.1, while the second test displayed a variation ranging from 0 to 0.5.

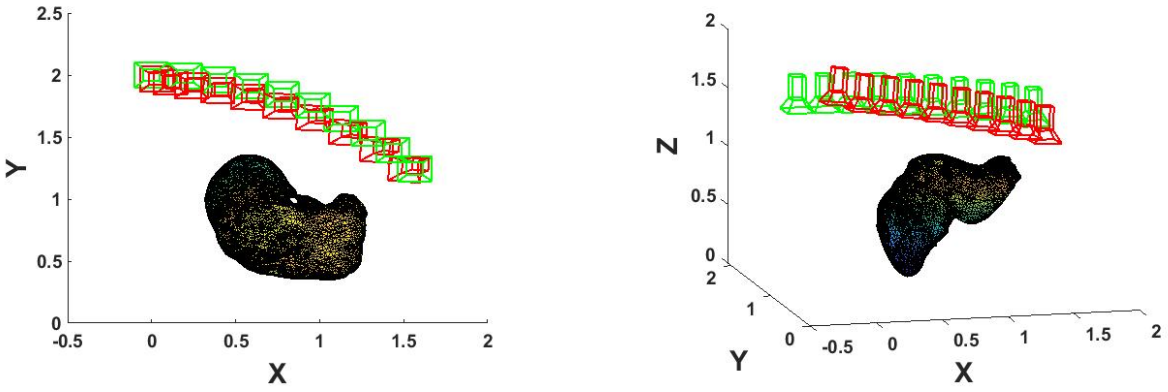

Figure 5.12: The liver experiences nonlinear elastic deformation, while the camera follows a reference curvilinear trajectory. The results from test (1) are presented, with the left image showing the top view and the right image displaying the side view. The green cameras indicate the reference trajectory, while the red cameras represent the corrected trajectory.

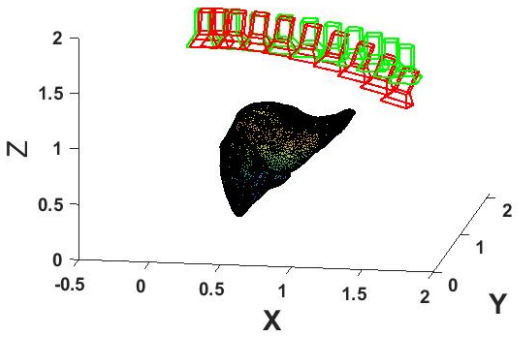

Figure 5.13: The liver experiences nonlinear elastic deformation, and the camera follows a reference curvilinear trajectory. The results from test (2) are shown, with the green cameras representing the reference trajectory and the red cameras indicating the corrected trajectory.

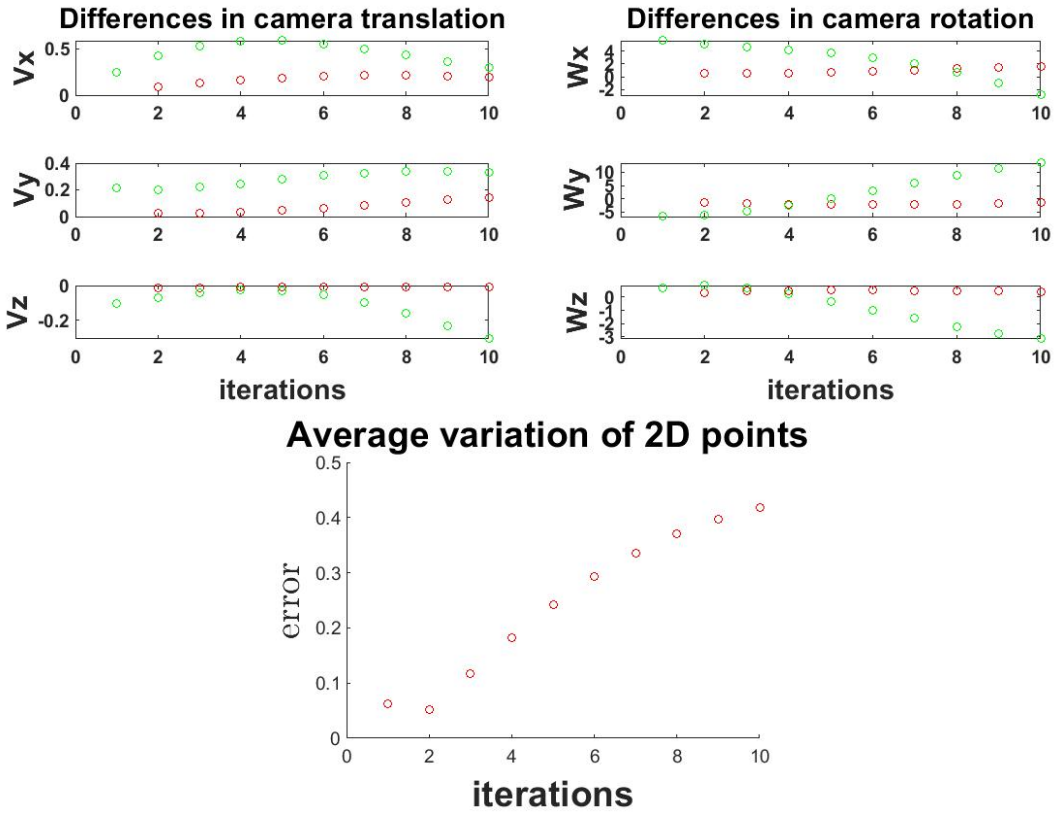

Figure 5.14: The performance of camera speed while tracking a deformed object with nonlinear deformation along a curvilinear trajectory is analyzed. The left figure displays the camera speed in translation, while the right figure shows the camera speed in rotation. Red dots indicate the camera speed following the strategy from test (1), and green dots represent the camera speed based on the strategy from test (2).

5.8.6 Robustness to Image Noise: Experimental Results

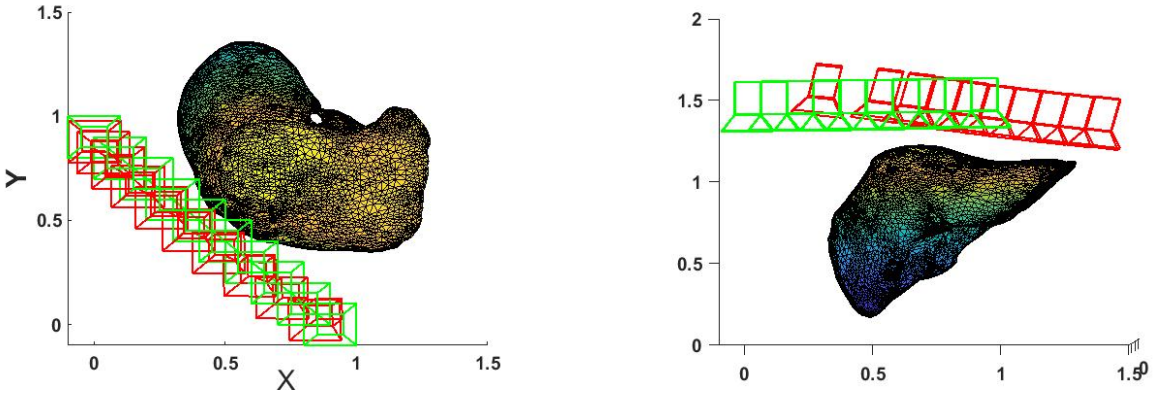

Figure 5.15: The liver experiences nonlinear elastic deformation while the camera follows a reference curvilinear trajectory. The results from test (1), which include noisy data, are shown. The left figure presents the top view, and the right figure shows the side view. The green cameras indicate the reference trajectory, while the red cameras represent the corrected trajectory.

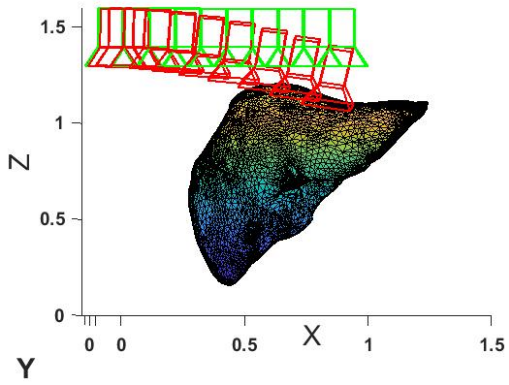

Figure 5.16: The liver undergoes nonlinear elastic deformation as the camera follows a reference curvilinear trajectory. The results from test (1), which involve noisy data, are displayed. The left figure shows the top view, and the right figure illustrates the side view. The reference trajectory is represented by green cameras, while the corrected trajectory is shown by red cameras.

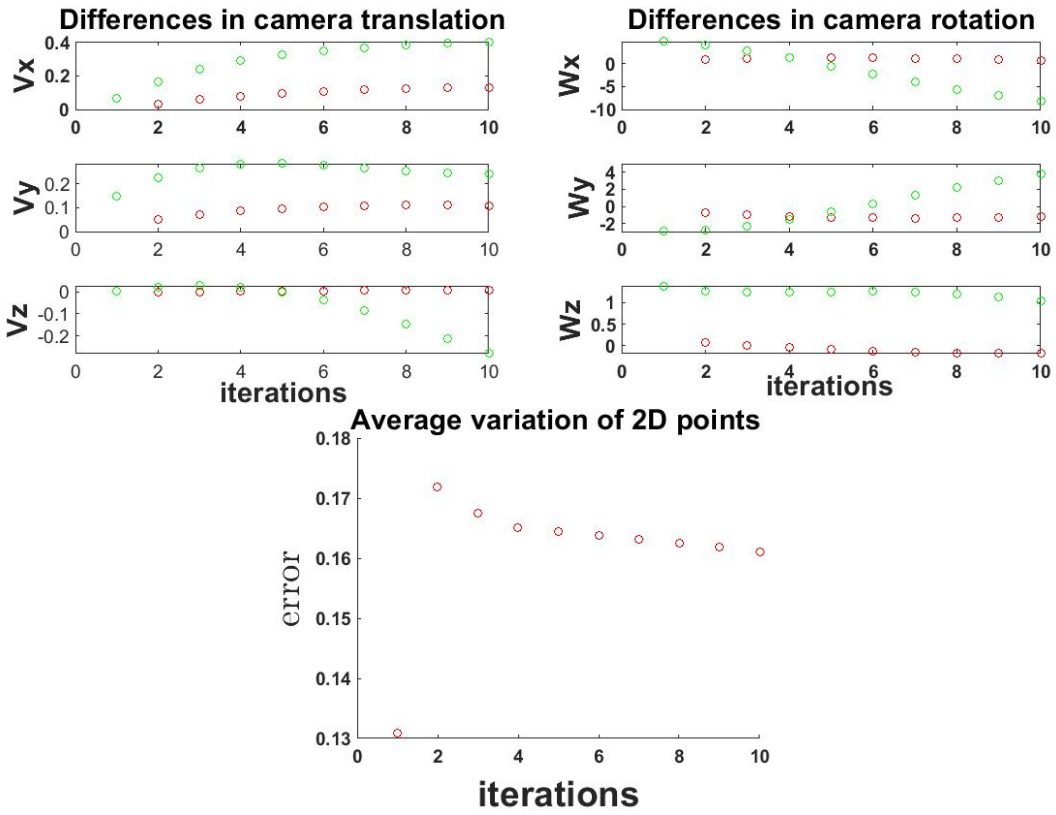

Figure 5.17: This analysis evaluates the camera speed performance while tracking a deformed object with nonlinear deformation and noisy data. The left figure depicts the camera speed in translation, while the right figure shows the camera speed in rotation. Red dots represent the camera speed based on the strategy from test (1), and green dots represent the camera speed according to the strategy from test (2).

Our method focuses on enhancing the robustness of camera motion control. To this end, we attempted to resynthesize the image measurements by introducing Gaussian random noise into the equation eq.5.14. This noise is defined as $\sigma = randn(1, 12226)$. It is added to the coordinates of the image im_i. We conducted a test on nonlinear deformation, comparing the results of test 1 and test 2. We observed that the camera followed the trajectory but was not very accurate for the tracking of the deformation in the case where the weight equals 1 as it is shown in figure 5.15. And for the case of the part that carries an important weight the camera has well followed the trajectory and the deformation as it is indicated in figure 5.16. So we can say that the camera has not been influenced by the perturbations for the trajectory tracking and it is weak influenced for the deformation tracking. Figure 5.17 The differences in speed between the camera's translation, rotation, and pixel point variation are shown. Green represents test 1, while red represents test 2. We observed that test 1 is slower in both translation and rotation compared to test 2, while the pixel point variation remains the same for both tests.

5.8.7 Interpretation and Comments

The significance of the weights is clearly demonstrated in Figures 5.5 and 5.6. In Figure 5.6-left, we observe how the reference trajectory bends significantly as it approaches the most deformed area, a behavior not seen when all weights are equal. In the right figures, we can see that the deformed region in Figure 5.6-right aligns more closely with the reference view than in Figure 5.5-right. The importance of the optimal visual control framework and its robustness is evident in the results presented in Figures 5.8 through 5.17. These figures highlight the reliability of the proposed method across different deformation types (linear/nonlinear), trajectory types (geometrically linear/nonlinear), and noise levels in 2D detection. It is clear that, in all scenarios tested, the proposed approach enables the computation of a precise rigid camera correction. By assigning higher weights to the most relevant triangles in the deformable area, the reference trajectory is adjusted to better align with these regions, regardless of the deformation, trajectory type, or noise present.

5.9 Conclusion

This section introduces a nonparametric method for visually compensating nonrigid soft tissue deformation while maintaining rigid camera motion. The compensation process is carried out as camera follows a reference trajectory, employing continuous weight functions that prioritize the most relevant deformed regions while stabilizing the view in real-time to match the reference. Our compensation strategy is integrated into an optimal control framework. Through experimentation, we demonstrated that our approach is robust to both linear and nonlinear deformations, various types of geometric paths, and noise in 2D detection. Additionally, we emphasize that our method does not depend on any specific information regarding the type of soft tissue, the nature of the deformation, or the trajectory.

Conclusion

The work presented in this dissertation concerns the modeling and definition of viewing strategies for deformable objects. The aim was to develop a method for modeling the interaction between a camera and a large deformation object as well as a strategy for planning to see these objects.

In the first chapter, it describes visual assistance methods for stiff objects by connecting them to non-rigid objects. It addresses technical issues related to the selection of visual characteristics. Then it discusses visual aiding techniques related to deformable objects. It divides these methods into two groups: those that use parameters and mechanical models, and those that use parameters and geometrically-based models only. This classification enables him to recognize the contribution he can make to the state of the art of visual service on deformable objects. It concludes this chapter by discussing applications. visual assistance with deformable objects during surgical procedures assisted by computers and/or robots falls within the medical field. These areas of use allowed him to justify the value of his contribution in light of its potential applicability in a contemporary surgical environment. In the second chapter, there are visual characteristics to look for as the foundation for the subsequent development of the visual assistance algorithm. We demonstrate the calculation of the interaction matrix for both points and lines. It builds the interaction matrix related to these characteristics for a bi-dimensional polygonal primitive geometrie. He presents experiments on stiff and flexible polygons using traditional visual assimilation techniques. This study allows him to draw the first conclusions on the resilience of Conclusion about the durability of using points or lines as visual characteristics of the deformable objects on which the servicing would be performed.

Examine the selection of visual characteristics separate from the target object's geometric form in the third chapter. This characteristic is represented by calculated order 1 and order 2 visual moments based on object outlines. It formulates the mathematical computation of the interaction matrix corresponding to the first and second-order moments for a basic two-dimensional polygonal geometry. In this chapter, he decides to use a simple geometric figure of the type 2D triangle. This decision is driven by the fact that a triangle may be used as a basic elementary geometrie to mail any object. On a rigid triangle and a deformable triangle, a set of visual perception tests were conducted. Through these testing, it was shown that visual characteristics based on moments of order 1 and 2 are less reliable than visual characteristics based on points. This conclusion is supported

by the observation that the visual assimilation using the first characteristic diverges more readily than the visual assimilation using the second characteristic. In the fourth chapter, one makes use of the insights gained from chapters two and three to provide the groundwork for the visual assimilation of a deformable object, which is a major contribution of the author's doctoral thesis. It establishes a global object interaction matrix. The object is said to be made up of bi-dimensional triangles that are arranged in a pattern that covers its surface. Multiple interaction submatrices make up the overall interaction matrix. Each triangle's unique sub-matrix has its own characteristics. Thus, the global matrix may be defined as a set of piecewise Jacobian image-camera constants. Each triangle's interaction with the camera is represented by a real positive number that indicates the triangle's relative relevance to the rest of the triangles. This contribution enables the correction to be focused on the most significant portion of the deformed object during the visual assessment. This general approach makes it possible to address the issue of visual servoing on deformable objects without taking into account the object's geometry or deformation mechanics. It has been suggested to calculate this ponderation as the exact and normalized difference between the target triangle's surface and the deformed triangle's surface. Experiments on deformable triangles serve as the chapter's climax. The numerous tests and experiments carried out in the following chapter are motivated by the first encouraging findings regarding the implementation of this approach. We give the second portion of his contribution in the last chapter. explains in detail how to use the visual referencing style that was developed in the previous chapter of Track the path of a broken object. These explanations are supported by the functional formalization. The task with the criteria is to have the best possible visual of the deformed object at all times along the course of the journey. The triangular weight idea introduced in the preceding chapter allows for the definition and development of the triangular grid-based movement planning concept

Test the robustness of the approach for each combination of trajectory and kind of deformation. Proposal to include a German noise due to the detection of visual characteristics each experiment. They are critically discussed and analyzed in order to highlight their merits. The method suggested and the reasoning of the decisions made to prevent undesirable deviations correction imposed on camera movements

We have presented a novel camera control method designed to execute a series of visual servoing tasks based on the principle of active perception. This method leverages optimal visual control of models in servoing deformable objects, using constraints that guide the camera to adjust and correct errors. The method has been applied to assist multiple robots, even in the presence of noise, to follow predefined trajectories. Through the simulation results provided, we have demonstrated the effectiveness of our approach and emphasized the significance of visible servo movement during the deformation of the control parameters' configuration. Future study should investigate various options and paths that this thesis's work might go. The following are possible future work additions:

Identification of many target objects in a single recorded image use of computer vision for target object detection was implemented with a limit on the number of items that could be identified in a collected image. Future research should look on identifying several target items inside a single picture. Taking into account changing environments: In path planning, we studied static settings whose contents stay constant during the travel. Future route planning should take into account changing environments. The development of autonomous exploration tactics should give solutions to the difficulties of long-term environmental change.

Appendix A

Interaction matrix

A.1 Calculation of the Visual Servoing Jacobian

Considering a point defined by its coordinates in three-dimensional space $(X_A, Y_A, Z_A)^\top$ when expressed in the camera coordinate frame, the 2D perspective projection of a 3D point onto the image plane is given by

$$x_A = X_A/Z_A, \tag{A.1}$$
$$y_A = Y_A/Z_A. \tag{A.2}$$

Where $im_A = (x_A, y_A)^\top$ this represents the image coordinates of the 3D point, excluding the effects of focal length and principal point, as we assume a calibrated camera with known intrinsic parameters. The time derivative of the above projection with respect to a reference 3D vertex coordinate is expressed as follows: (X_0, Y_0, Z_0) is given as

$$\dot{x_A} = \dot{X}_A/Z_A - \dot{Z}_A x_A/Z_A, \tag{A.3}$$
$$\dot{y_A} = \dot{y_A}/Z_A - \dot{Z}_A y_A/Z_A. \tag{A.4}$$

Let us denote the translational velocity of the camera as follows $v_A = (v_x, v_y, v_z)^\top$ and its rotation velocity as $\omega_A = (\omega_x, \omega_y, \omega_z)^\top$ Thus, the complete camera velocity vector can be expressed as $\dot{c_A} = (v_A^\top, \omega_A^\top)^\top$. The relationship between the velocity of a 3D point and the camera's motion can be expressed using the following equation, as established in the classical work by [17]:

$$\dot{X} = -v_x - \omega_y Z_A + \omega_z Y_A, \tag{A.5}$$
$$\dot{Y} = -v_y - \omega_z X_A + \omega_x Z_A, \tag{A.6}$$
$$\dot{Z} = -v_z - \omega_x Y_A + \omega_y X_A. \tag{A.7}$$

By substituting the 3D velocity expression from the previously introduced equations, we obtain the following formulation: A.5-A.7 into the expressions of 2D velocity A.3-A.4

$$\dot{x_A} = -v_x/Z_A + x_A v_z/Z_A + +x_A y\omega_x - (1 + x_A^2)\omega_y + y\omega_z, \tag{A.8}$$

$$\dot{y_A} = -v_y/Z_A + y_A v_z/Z_A + (1 + y_A^2)\omega_x - x_A y_A \omega_y - x_A \omega_z. \tag{A.9}$$

By rearranging the terms and applying the interaction matrix notation introduced in Equation 5.16, the system can be reformulated as follows:

$$(\dot{x_A}, \dot{y_A})^\top = L(x_A, y_A, Z_A)\dot{c_A}. \tag{A.10}$$

Assuming a sufficiently high sampling period T(e.g., at least 30 frames per second), the above continuous-time formulation can be discretized as follows:

$$(x_A - x_0, y_A - y_0)^\top / T = L_A(x_A, y_A, Z_A)(c_A - c_0)/T \tag{A.11}$$

Where c_A is represented by the translation position of the camera center and the Rodrigues rotation vector.

$$c_A = (t_x, t_y, t_z, \theta_x, \theta_y, \theta_z)^\top \tag{A.12}$$

Where (t_x, t_y, t_z) are the 3D position coordinates and $\theta = \sqrt{\theta_x^2 + \theta_y^2 + \theta_z^2}$ is the angle of rotation and $(\theta_x/\theta, \theta_y/\theta, \theta_z/\theta)$ is the unit axis of rotation [65]. $c_0 = (t_x^0, t_y^0, t_z^0, \theta_x^0, \theta_y^0, \theta_z^0)^\top$ is a reference camera configuration close to c. Simplifying equation A.11 by the sampling period on both sides gives

$$(x - x_0, y - y_0)^\top = L(x, y, Z)(c - c_0) \tag{A.13}$$

This mapping is referred to as the interaction matrix, which relates small variations in camera velocity to small variations in the 2D projected points. Consider the mapping Π, defined in equation 5.13, which assumes the projection of a 3D triangle as a composite geometric primitive consisting of 3 vertices. The first-order Taylor expansion of equation 5.14, centered around a reference camera configuration c_0 and close to a given camera configuration c, is provided in equation 5.15. The term $\frac{\partial \Pi}{\partial c}$ represents the interaction matrix, which relates changes in the camera configuration to variations in the 2D projection of the three vertices composing the triangle. Therefore, the analytical expression for $\frac{\partial \Pi}{\partial c}$ is obtained by vertically concatenating the matrix given in equation A.10. Hence, the matrix $\frac{\partial \Pi}{\partial c}$ is 6 columns.It contains 6 rows, as each vertex of the triangle contributes 2 equations, and since the triangle has three vertices, this results in a total of 6 equations.

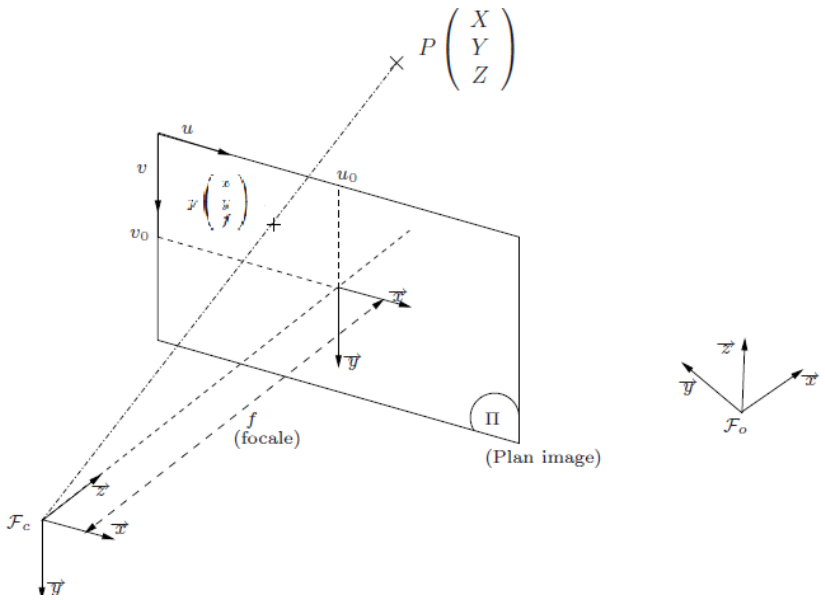

Figure A.1: Visual servo motion control diagram of a robotic system.

A.2 Interaction matrix of line

$$Ax + By + C = 0 \tag{A.14}$$

This represents the equation of the plane to which the line belongs. We will denote the equation using the following parameters: (ρ, Θ).

$$im(\mathbf{x}, p) = xcos\Theta + ysin\Theta - \rho \tag{A.15}$$

2D primitive: $im(\mathbf{x}, p)$.
Minimal parameterization $p = (\rho, \Theta)$

The temporal derivative of A.15

$$\dot{im}(\mathbf{x}, p) = 0 \tag{A.16}$$

We drift in relation (ρ, Θ)

$$\frac{\partial im}{\partial \mathbf{x}}(\mathbf{x}, p) + \frac{\partial im}{\partial p}(\mathbf{x}, p) = 0 \tag{A.17}$$

$$-x\dot{\Theta}sin\Theta + y\dot{\Theta}cos\Theta + \dot{\rho} + \dot{x}cos\theta + \dot{y}sin\Theta \quad = \quad 0 \tag{A.18}$$

with

$$\dot{\rho} + (x\sin\Theta - ycos\Theta) + \dot{x}\cos\Theta + \dot{y}sin\Theta = 0 \tag{A.19}$$

And

$$\dot{\rho} + (x \sin\Theta - ycos\Theta) = \dot{x} \cos\Theta + \dot{y}sin\Theta \tag{A.20}$$

With

$$\dot{x} = -v_x/Z + xv_z/Z + +xy\omega_x - (1+x^2)\omega_y + y\omega_z, \tag{A.21}$$

$$\dot{y} = -v_y/Z + yv_z/Z + (1+y^2)\omega_x - xy\omega_y - x\omega_z \tag{A.22}$$

And

$$x = \frac{\rho}{\cos\Theta} - y \tan\Theta \tag{A.23}$$

We replace $\dot{x}$ and $\dot{y}$ from the equation A.22 in the equation A.20 and x from the equation A.23 in the equation A.20

$$-\frac{\dot{\Theta}}{cos\Theta}y + \dot{\rho} + \rho tan\Theta\dot{\Theta} = yK_1\dot{c} + K_2\dot{c} \tag{A.24}$$

with

$$\begin{pmatrix} K_1 \\ k_2 \end{pmatrix} = \begin{bmatrix} -\lambda_1 cos\Theta & \lambda_2 sin\Theta & \lambda_1\rho & -\rho & -\rho tan\Theta & -\dfrac{1}{cos\Theta} \\ -\lambda_2 cos\Theta & \lambda_2 sin\Theta & \lambda_2\rho & -sin\Theta & \dfrac{cos\Theta + \rho^2}{cos\Theta} & \rho tan\Theta \end{bmatrix} \tag{A.25}$$

And

$$\lambda_1 = Atan\Theta + B \tag{A.26}$$

$$\lambda_2 = \frac{A\rho}{cos\Theta} + C \tag{A.27}$$

$$\tag{A.28}$$

Because we are use equation A.29 and equation A.23 to represent λ_1 and λ_2.

$$Ax + By + C = 0 \tag{A.29}$$

$$\lambda_\rho = -A\cos\Theta + B\rho\sin\Theta + C \tag{A.30}$$

$$\lambda_\Theta = Bcos\Theta - Asin\Theta \tag{A.31}$$

$$L_s = (L_\rho^T, L_\Theta^T)^T \tag{A.32}$$

$$\tag{A.33}$$

A.3 Step to non rigid

$$\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} = \begin{bmatrix}\frac{1}{Z(t)} & 0 & \frac{-x(t)}{Z(t)}\\ 0 & \frac{1}{Z(t)} & \frac{-y(t)}{Z(t)}\end{bmatrix} * v \tag{A.34}$$

v It's speed $v = [\dot{X}, \dot{Y}, \dot{Z}]'$

Π Jacobian of projection

$$\Pi = \begin{bmatrix}\frac{1}{Z(t)} & 0 & \frac{-x(t)}{Z(t)}\\ 0 & \frac{1}{Z(t)} & \frac{-y(t)}{Z(t)}\end{bmatrix}$$

$$X^b(t) = P(t) + N(t) \tag{A.35}$$

$N(t)$:is the deformation in lagrange rotation consider only planar deformation $P'(t), P(t), U^b(t)$. coordinates inplanre framre geometric representation of non-rigide deformation

$$P'(t) = P(t) + N(t) \tag{A.36}$$

$$P'(t) = T(t)P(t) \tag{A.37}$$

$$T(t) = \begin{bmatrix}S_x(t) & \tau_x(t) & \delta_x(t)\\ \tau_y(t) & S_y(t) & \delta_y(t)\\ \delta_z(t) & \tau_z(t) & S_z(t)\end{bmatrix}$$

$$X(t) = RP'(t) + P(t) \tag{A.38}$$

$$\dot{X}(t) = \dot{R}P'(t) + \dot{P}(t) + RP'\dot{(}t) \tag{A.39}$$

$$\dot{X}(t) = \dot{R}(t)R^{-1}(t)[RP'(t) + P(t)] + \dot{P}(t) - R^{-1}(t)P(t) + R\dot{T}P(t) \tag{A.40}$$

$$\dot{X}(t) = \omega_c x X(t) + V_c + R\dot{T}P(t) \tag{A.41}$$

The interaction matrix for a point in the image space, where the point has coordinates $x = (x, y)$ and a depth Zin the camera reference frame, is given by:

$$\begin{pmatrix}\dot{x}\\ \dot{y}\end{pmatrix} = L(t)v + \Pi(t)R(t)\dot{T}(t)P(t) \tag{A.42}$$

$$L(t) = \begin{bmatrix}\frac{1}{Z(t)} & 0 & \frac{-y(t)}{Z(t)} & -x(t)y(t) & 1+x(t)^2 & -y(t)\\ 0 & \frac{1}{Z(t)} & \frac{-y(t)}{Z(t)} & -(1+y(t)^2) & x(t)y(t) & x(t)\end{bmatrix}$$

v : it's speed

$$v = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]'$$

$$P(t) = \begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \end{pmatrix} \quad R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix}$$

$$R(t)\dot{T}(t)P(t) = R(t) \begin{bmatrix} \dot{S}_x(t)X_0 + \dot{\tau}_x(t)Y_0 + \dot{\delta}_x(t)Z_0 \\ \dot{\tau}_y(t)X_0 + \dot{S}_y(t)Y_0 + \dot{\delta}_y(t)Z_0 \\ \dot{\delta}_z(t)X_0 + \dot{\tau}_z(t)Y_0 + \dot{S}_z(t)Z_0 \end{bmatrix} \tag{A.43}$$

$$R(t)\dot{T}(t)P(t) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{A.44}$$

A.4 Visibility-driven triangle weight computation

There is no universally optimal or exact method for computing weight functions, as noted in [56]. Nonetheless, several strategies can be employed to construct continuous weight functions. The most straightforward approach is to assign constant weights, as was done in the protocol of test (1) in our experiment. An alternative method involves adapting the weights based on the magnitude of deformation, thereby prioritizing the most deformed regions. In this context, let us consider a single undeformed 3D triangle observed by a camera moving along a reference trajectory $\gamma(s)$. The projection of the triangle onto the image plane of the reference camera changes gradually as the camera progresses along its trajectory $\gamma(s)$ since it is a static object seen by a continuously moving camera (see figure 5.3). The position of the triangle's 2D projection on the image plane can be expressed as follows

$$im_0(s) = \left(x_1^0(s), x_2^0(s), x_3^0(s), y_1^0(s), y_2^0(s), y_3^0(s)\right)^\top. \tag{A.45}$$

The area of the projected triangle along the trajectory $\gamma(s)$ is also a continuous function and can be computed at each point s as follows:

$$a_0(s) = \frac{1}{2}\det \begin{pmatrix} x_1^0(s) & y_1^0(s) & 1 \\ x_2^0(s) & y_2^0(s) & 1 \\ x_3^0(s) & y_3^0(s) & 1 \end{pmatrix}. \tag{A.46}$$

Where det() let $\det(\cdot)$ denote the determinant of square matrices. If $a(s)$ represents the area of the triangle projected onto the image plane of the camera at runtime, then the corresponding weight function can be computed using the following expression

$$w(s) = a_0(s)\frac{\text{abs}(a_0(s) - a(s))}{(1 + a_0(s))}. \tag{A.47}$$

Where abs() the symbol $|\cdot|$ denotes the absolute value of real numbers. The weight function $w(s)$ defined above is continuous, as it is composed of continuous functions $a(s)$ and $a_0(s)$, and remains strictly positive for all values of s for $0 \leq s \leq 1$. When the triangle is not visible in the reference view, its area $a_0(s)$ becomes zero, resulting in a corresponding weight of zero. Moreover, the greater the deformation of the triangle, the larger the value of the weight function $\text{abs}(a_0(s) - a(s))$ this weighting mechanism allows greater emphasis to be placed on the deformed regions, thereby enhancing the sensitivity of the control strategy to areas undergoing significant deformation

Appendix B

Scientific publications, workshop participations and scientific popularization activities

The following is a list of publications written over the last few years, which are either published or under review.

Publications

* Yasser Derrar, Farah Saidi, Abed Malti: Visual Servoing for Deformable Objects with Pre-Planned Trajectory-Guided Geometric Primitives ». Automation and Remote Control,Volume 86, Issue: 2,2025, Pages 165-189

International communications

* Y.DERRAR , A. MALTI, Visual servoing on trianglar primitive using the moment approach, 4ème conférence internationale sur le signal, l'image, la vision et leurs applications, SIVA 2018, Guelma, Algérie, 26- 27 november 2018

* Y.DERRAR , A. MALTI, A Visual Servoing Strategy on Multiple Triangles using the Image Moment Approach, 6th International Conference on Control Engineering and Information Technology, CEIT 2018, Istamboul, Turkey, 25-27 October 2018

* Y.DERRAR , A. MALTI Comparaison d'approche d'asservissement visuel des lignes droites, 6eme Conférence Internationale en Automatique Traitement, ATS-2018, Tanger, Maroc, 28- 30 Mars 2018.

* Y.DERRAR , A. MALTI A rigid visual servoing approach for elastic objectifs using weighted primitives , 2nd international conference on advanced intelligent systems

Bibliography

[1] 3d steering of a flexible needle by visual servoing | SpringerLink.

[2] Détection et suivi des aiguilles à base de gabor dans des volumes de données d'échographie tridimensionnelle - portail de la recherche de l'université technologique d'eindhoven.

[3] Nouvelles contributions à l'application des moments en asservissement visuel - TEL - thèses en ligne.

[4] An Overview of Visual Servoing for Robot Manipulators - Technical Articles.

[5] Performance tests for visual servo control systems, with application to partitioned approaches to visual servo control - nicholas r. gans, seth a. hutchinson, peter i. corke, 2003.

[6] Real-time parametric curved needle segmentation in 3d ultrasound images - IEEE conference publication.

[7] Ultrasound-guided robot for flexible needle steering - IEEE journals & magazine.

[8] D. J. Agravante, G. Claudio, F. Spindler, and F. Chaumette. Visual servoing in an optimization framework for the whole-body control of humanoid robots. 2(2):608–615. Conference Name: IEEE Robotics and Automation Letters.

[9] Miguel Aranda, Juan Antonio Corrales Ramon, Youcef Mezouar, Adrien Bartoli, and Erol Özgür. Monocular Visual Shape Tracking and Servoing for Isometrically Deforming Objects. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7542–7549, October 2020. ISSN: 2153-0866.

[10] S. F. Atashzar, I. Khalaji, M. Shahbazi, A. Talasaz, R. V. Patel, and M. D. Naish. Robot-assisted lung motion compensation during needle insertion. In *2013 IEEE International Conference on Robotics and Automation*, pages 1682–1687.

[11] M.C. Bernardes, B.V. Adorno, P. Poignet, N. Zemiti, and G.A. Borges. Adaptive path planning for steerable needles using duty-cycling. In *IEEE International Conference on Intelligent Robots and Systems*, pages 2545–2550, 2011. IEEE/RSJ

International Conference on Intelligent Robots and Systems ; Conference date: 25-09-2011 Through 30-09-2011.

[12] Xiaomeng Cao, Jian Lan, and X. Rong Li. Extension-deformation approach to extended object tracking. In *2016 19th International Conference on Information Fusion (FUSION)*, pages 1185–1192, 2016.

[13] David Casillas-Perez, Daniel Pizarro, David Fuentes-Jimenez, Manuel Mazo, and Adrien Bartoli. Equiareal Shape-from-Template. *Journal of Mathematical Imaging and Vision*, 61(5):607–626, June 2019.

[14] R. J. Chang and W. J. Chien. Eye-In-Hand/Eye-To-Hand Cooperated Peg-In-Hole Microassembly Operations Using Coordinate-Based Augmented Reality, June 2022.

[15] P. Chatelain, A. Krupa, and N. Navab. 3d ultrasound-guided robotic steering of a flexible needle via visual servoing. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2250–2255.

[16] F. Chaumette. Image moments: a general and useful set of features for visual servoing. 20(4):713–723.

[17] F. Chaumette and S. Hutchinson. Visual servo control. i. basic approaches. 13(4):82–90. Conference Name: IEEE Robotics Automation Magazine.

[18] François Chaumette. *Visual Servoing*, pages 869–874. Springer US, Boston, MA, 2014.

[19] François Chaumette. Avancées récentes en asservissement visuel. page 103, 2003.

[20] Qiang Chen, Quan-Sen Sun, Pheng Ann Heng, and De-Shen Xia. Two-stage object tracking method based on kernel and active contour. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(4):605–609, 2010.

[21] Zhong Chen, Shuai Li, Ning Zhang, Yuya Hao, and Xianmin Zhang. Eye-to-hand robotic visual tracking based on template matching on fpgas. *IEEE Access*, 7:88870–88880, 2019.

[22] J. Chevrie, N. Shahriari, M. Babel, A. Krupa, and S. Misra. Flexible needle steering in moving biological tissue with motion compensation using ultrasound and force feedback. 3(3):2338–2345.

[23] J. Chevrie, N. Shahriari, M. Babel, A. Krupa, and S. Misra. Flexible needle steering in moving biological tissue with motion compensation using ultrasound and force feedback. 3(3):2338–2345.

[24] Cheng Chi and Dmitry Berenson. Occlusion-robust deformable object tracking without physics simulation. pages 6443–6450.

[25] A.I. Comport, E. Marchand, M. Pressigout, and F. Chaumette. Real-time markerless tracking for augmented reality: the virtual visual servoing framework. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):615–628, July 2006. Conference Name: IEEE Transactions on Visualization and Computer Graphics.

[26] P. Corke and S. Hutchinson. A new partitioned approach to image-based visual servo control.

[27] Yasser Derrar and Abed Malti. A visual servoing strategy on multiple triangles using the image moment approach,. In *6th International Conference on Control Engineering and Information Technology,*, pages 1–7.

[28] S. P. DiMaio and S. E. Salcudean. Needle steering and motion planning in soft tissues. 52(6):965–974.

[29] Bernard Espiau, François Chaumette, and Patrick Rives. A new approach to visual servoing in robotics. In Christian Laugier, editor, *Geometric Reasoning for Perception and Action*, Lecture Notes in Computer Science, pages 106–136. Springer.

[30] Xiaoyu Feng, Wei Mei, and Dashuai Hu. A Review of Visual Tracking with Deep Learning. pages 231–234. Atlantis Press, November 2016. ISSN: 1951-6851.

[31] R. Ginhoux, J. Gangloff, M. de Mathelin, L. Soler, M. M. A. Sanchez, and J. Marescaux. Active filtering of physiological motion in robotized surgery using predictive control. 21(1):67–79.

[32] Abdul Hafez Abdul Hafez and Enric Cervera. Particle-filter-based Pose Estimation from Controlled Motion with Application to Visual Servoing. *International Journal of Advanced Robotic Systems*, 11(10):177, October 2014. Publisher: SAGE Publications.

[33] Nazim Haouchine, Jeremie Dequidt, Igor Peterlik, Erwan Kerrien, Marie-Odile Berger, and Stephane Cotin. Image-guided simulation of heterogeneous tissue deformation for augmented reality during hepatic surgery. In *2013 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 199–208.

[34] Zhe Hu, Tao Han, Peigen Sun, Jia Pan, and Dinesh Manocha. 3-d deformable object manipulation using deep neural networks. *IEEE Robotics and Automation Letters*, 4(4):4255–4261, 2019.

[35] Zhe Hu, Peigen Sun, and Jia Pan. Three-dimensional deformable object manipulation using fast online gaussian process regression. *IEEE Robotics and Automation Letters*, 3(2):979–986, 2018.

[36] F. Janabi-Sharifi and W. J. Wilson. Automatic selection of image features for visual servoing. 13(6):890–903. Conference Name: IEEE Transactions on Robotics and Automation.

[37] Biao Jia, Zhe Hu, Jia Pan, and Dinesh Manocha. Manipulating highly deformable materials using a visual feedback dictionary. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 239–246, 2018.

[38] Hanbyul Joo, Tomas Simon, and Yaser Sheikh. Total capture: A 3d deformation model for tracking faces, hands, and bodies. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[39] Kenta Kajihara, Shouren Huang, Niklas Bergström, Yuji Yamakawa, and Masatoshi Ishikawa. Tracking of trajectory with dynamic deformation based on dynamic compensation concept. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1979–1984, 2017.

[40] C.W. Kennedy, Tie Hu, and J.P. Desai. Combining haptic and visual servoing for cardiothoracic surgery. In *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, volume 2, pages 2106–2111 vol.2, May 2002.

[41] Yeoun Jae Kim, Jong Hyun Seo, Hong Rae Kim, and Kwang Gi Kim. Impedance and admittance control for respiratory-motion compensation during robotic needle insertion - a preliminary test. 13(4).

[42] A Krupa. Asservissement visuel par imagerie médicale. page 15.

[43] A. Krupa. Asservissement visuel par imagerie médicale. In *Journées Nationales de la Recherche en Robotique, JNRR'09*, Neuvy-sur-Barangeon, France, France, 2009.

[44] A. Krupa, J. Gangloff, C. Doignon, M.F. de Mathelin, G. Morel, J. Leroy, L. Soler, and J. Marescaux. Autonomous 3-D positioning of surgical instruments in robotized laparoscopic surgery using visual servoing. *IEEE Transactions on Robotics and Automation*, 19(5):842–853, October 2003. Conference Name: IEEE Transactions on Robotics and Automation.

[45] Alexandre Krupa. Contributions à l'asservissement visuel échographique.

[46] Alexandre Krupa. 3d steering of a flexible needle by visual servoing. In Polina Golland, Nobuhiko Hata, Christian Barillot, Joachim Hornegger, and Robert Howe,

editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2014*, pages 480–487, Cham, 2014. Springer International Publishing.

[47] Romain Lagneau. Shape control of deformable objects by adaptive visual servoing.

[48] Romain Lagneau, Alexandre Krupa, and Maud Marchal. Active deformation through visual servoing of soft objects. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8978–8984, 2020.

[49] Romain Lagneau, Alexandre Krupa, and Maud Marchal. Active Deformation through Visual Servoing of Soft Objects. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8978–8984, May 2020. ISSN: 2577-087X.

[50] Fouad Makiyeh, Maud Marchal, François Chaumette, and Alexandre Krupa. Indirect Positioning of a 3D Point on a Soft Object Using RGB-D Visual Servoing and a Mass-Spring Model. In *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 235–242, December 2022.

[51] E. Malis, F. Chaumette, and S. Boudet. 2 1/2 d visual servoing. 15(2):238–250.

[52] Abed Malti, Adrien Bartoli, and Toby Collins. A pixel-based approach to template-based monocular 3d reconstruction of deformable surfaces. pages 1650–1657.

[53] Abed Malti, Adrien Bartoli, and Richard Hartley. A linear least-squares solution to elastic shape-from-template. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[54] Abed Malti, Richard Hartley, Adrien Bartoli, and Jae-Hak Kim. Monocular template-based 3d reconstruction of extensible surfaces with local linear elasticity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[55] Abed Malti and Cedric Herzet. Elastic shape-from-template with spatially sparse deforming forces. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[56] Abed Malti, Michel Taïx, and Florent Lamiraux. A general framework for planning landmark-based motions for mobile robots. *Advanced Robotics*, 25(11-12):1427–1450, 2011.

[57] Abed C. Malti, Michel Taïx, and Florent Lamiraux. A general framework for planning landmark-based motions for mobile robots. 25(11):1427–1450.

[58] Éric Marchand and François Chaumette. Virtual Visual Servoing: a framework for real-time augmented reality. *Computer Graphics Forum*, 21(3):289–297, 2002. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-8659.t01-1-00588.

[59] Seyed Mojtaba Marvasti-Zadeh, Li Cheng, Hossein Ghanei-Yakhdan, and Shohreh Kasaei. Deep Learning for Visual Tracking: A Comprehensive Survey. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–26, 2021.

[60] Kim Mathiassen, Kyrre Glette, and Ole Jakob Elle. Visual servoing of a medical ultrasound probe for needle insertion. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3426–3433.

[61] R. Mebarki, A. Krupa, and F. Chaumette. 2-d ultrasound probe complete guidance by visual servoing using image moments. 26(2):296–306.

[62] M. Moll and L.E. Kavraki. Path planning for deformable linear objects. 22(4):625–636.

[63] Cecília Molnár, Tamás D. Nagy, Renáta Nagyné Elek, and Tamás Haidegger. Visual servoing-based camera control for the da vinci surgical system. In *2020 IEEE 18th International Symposium on Intelligent Systems and Informatics (SISY)*, pages 107–112, 2020.

[64] Marco Mura, Yasmeen Abu-Kheil, Gastone Ciuti, Marco Visentini-Scarzanella, Arianna Menciassi, Paolo Dario, Jorge Dias, and Lakmal Seneviratne. Vision-based haptic feedback for capsule endoscopy navigation: a proof of concept. 1-4(11):35–45.

[65] Richard M. Murray, Zexiang Li, S. Shankar Sastry, and S. Shankara Sastry. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, March 1994.

[66] Florent Nageotte, Christophe Doignon, Michel de Mathelin, Philippe Zanne, and Luc Soler. Circular needle and needle-holder localization for computer-aided suturing in laparoscopic surgery. In *Medical Imaging 2005: Visualization, Image-Guided Procedures, and Display*, volume 5744, pages 87–98. SPIE, April 2005.

[67] David Navarro-Alarcon, Yunhui Liu, José Guadalupe Romero, and Peng Li. Visually servoed deformation control by robot manipulators. In *2013 IEEE International Conference on Robotics and Automation*, pages 5259–5264, May 2013. ISSN: 1050-4729.

[68] David Navarro-Alarcón, Yun-Hui Liu, José Guadalupe Romero, and Peng Li. Model-Free Visually Servoed Deformation Control of Elastic Objects by Robot Manipulators. *IEEE Transactions on Robotics*, 29(6):1457–1468, December 2013. Conference Name: IEEE Transactions on Robotics.

[69] L. Ott, Ph. Zanne, Fl. Nageotte, M. de Mathelin, and J. Gangloff. Physiological motion rejection in flexible endoscopy using visual servoing. In *2008 IEEE International Conference on Robotics and Automation*, pages 2928–2933, May 2008. ISSN: 1050-4729.

[70] Antoine Petit, Vincenzo Lippiello, Giuseppe Andrea Fontanelli, and Bruno Siciliano. Tracking elastic deformable objects with an RGB-d sensor for a pizza chef robot. 88:187–201.

[71] Ayelet Ramaty and Roy Luria. Visual Working Memory Cannot Trade Quantity for Quality. *Frontiers in Psychology*, 9, 2018.

[72] Ixchel G. Ramirez-Alpizar, Kensuke Harada, and Eiichi Yoshida. Motion planning for dual-arm assembly of ring-shaped elastic objects. In *2014 IEEE-RAS International Conference on Humanoid Robots*, pages 594–600, 2014.

[73] Xiaolin Ren, Hongwen Li, and Yuanchun Li. Image-based visual servoing control of robot manipulators using hybrid algorithm with feature constraints. *IEEE Access*, 8:223495–223508, 2020.

[74] Lucas Royer. Real-time tracking of deformable targets in 3d ultrasound sequences.

[75] Lucas Royer, Alexandre Krupa, Guillaume Dardenne, Anthony Le Bras, Eric Marchand, and Maud Marchal. Real-time target tracking of soft tissues in 3d ultrasound images based on robust visual information and mechanical simulation. 35:582–598.

[76] Lucas Royer, Maud Marchal, Anthony Le Bras, Guillaume Dardenne, and Alexandre Krupa. Real-time tracking of deformable target in 3d ultrasound images. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2430–2435. ISSN: 1050-4729.

[77] Ihab S. Mohamed. Mppi-vs: Sampling-based model predictive control strategy for constrained image-based and position-based visual servoing, 04 2021.

[78] Farah Saidi and Abed Malti. Fast and accurate nonlinear hyper-elastic deformation with a posteriori numerical verification of the convergence of solution: Application to the simulation of liver deformation. 37(5):e3444. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/cnm.3444.

[79] Hiroshi Seki, Mitsutaka Imamura, and Hiroshi Nagase. Evaluating Precise Quantity of Decommissioning Waste by Cutting Virtual 3D Models of Large Equipment. *Nuclear Science*, 5(3):36, October 2020. Number: 3 Publisher: Science Publishing Group.

[80] Agniva Sengupta, Alexandre Krupa, and Eric Marchand. Visual Tracking of Deforming Objects Using Physics-based Models. In *ICRA 2021 - IEEE International Conference on Robotics and Automation*, 2021.

[81] Mohammadreza Shetab-Bushehri, Miguel Aranda, Youcef Mezouar, and Erol Özgür. As-Rigid-as-Possible Shape Servoing. *IEEE Robotics and Automation Letters*,

7(2):3898–3905, April 2022. Conference Name: IEEE Robotics and Automation Letters.

[82] Valentina Staneva and Laurent Younes. Modeling and estimation of shape deformation for topology-preserving object tracking. *SIAM Journal on Imaging Sciences*, 7(1):427–455, 2014.

[83] O. Tahri and F. Chaumette. Complex objects pose estimation based on image moment invariants. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 436–441. ISSN: 1050-4729.

[84] O. Tahri and F. Chaumette. Image moments: generic descriptors for decoupled image-based visual servo. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, volume 2, pages 1185–1190 Vol.2. ISSN: 1050-4729.

[85] Marie-Aude Vitrani and Guillaume Morel. Hand-eye self-calibration of an ultrasound image-based robotic system. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1179–1185. ISSN: 2153-0866.

[86] Tan Wang, Weijun Wang, and Feng Wei. An overview of control strategy and trajectory planning of visual servoing. In Minrui Fei, Kang Li, Zhile Yang, Qun Niu, and Xin Li, editors, *Recent Featured Applications of Artificial Intelligence Methods. LSMS 2020 and ICSEE 2020 Workshops*, pages 358–370, Singapore, 2020. Springer Singapore.

[87] Guo-Qing Wei, K. Arbter, and G. Hirzinger. Real-time visual servoing for laparoscopic surgery. Controlling robot motion with color image segmentation. *IEEE Engineering in Medicine and Biology Magazine*, 16(1):40–45, January 1997. Conference Name: IEEE Engineering in Medicine and Biology Magazine.

[88] Fan Xu, Yuyou Zhang, Jing Sun, and Hesheng Wang. Adaptive Visual Servoing Shape Control of a Soft Robot Manipulator Using Bézier Curve Features. *IEEE/ASME Transactions on Mechatronics*, pages 1–11, 2022. Conference Name: IEEE/ASME Transactions on Mechatronics.

[89] Xingfeng Xu, Rongnian Tang, Lun Gong, Baojun Chen, and Siyang Zuo. Two Dimensional Position-Based Visual Servoing for Soft Tissue Endomicroscopy. *IEEE Robotics and Automation Letters*, 6(3):5728–5735, July 2021. Conference Name: IEEE Robotics and Automation Letters.

[90] D. Yasser and M. Abed. Visual servoing on a trianglar primitive using the moment approach. In *2018 International Conference on Signal, Image, Vision and their Applications (SIVA)*, pages 1–6.

[91] Derrar Yasser and Malti Abed. Comparison of visual servoing approaches on straight lines. page 8.

[92] Jihong Zhu. *Vision-based robotic manipulation of deformable linear objects*. PhD thesis, Université Montpellier, 2020.