



## MONOCULAR 3D RECONSTRUCTION OF SAIL FLYING SHAPE USING PASSIVE MARKERS

Luiz Maurílio da Silva Maciel

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientadores: Ricardo Guerra Marroquim  
Marcelo Bernardes Vieira

Rio de Janeiro  
Abril de 2018

MONOCULAR 3D RECONSTRUCTION OF SAIL FLYING SHAPE USING  
PASSIVE MARKERS

Luiz Maurílio da Silva Maciel

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ  
COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE)  
DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS  
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR  
EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

---

Prof. Ricardo Guerra Marroquim, D.Sc.

---

Prof. Marcelo Bernardes Vieira, D.Sc.

---

Prof. Claudio Esperança, Ph.D.

---

Prof. Daniel Ratton Figueiredo, Ph.D.

---

Prof. Ricardo Fabbri, Ph.D.

---

Prof. Leandro Augusto Frata Fernandes, D.Sc.

RIO DE JANEIRO, RJ – BRASIL

ABRIL DE 2018



Maciel, Luiz Maurílio da Silva

Monocular 3D reconstruction of sail flying shape using passive markers/Luiz Maurílio da Silva Maciel. – Rio de Janeiro: UFRJ/COPPE, 2018.

XIII, 65 p.: il.; 29, 7cm.

Orientadores: Ricardo Guerra Marroquim

Marcelo Bernardes Vieira

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2018.

Referências Bibliográficas: p. 61 – 65.

1. 3D reconstruction. 2. sail. 3. monocular.  
4. passive markers. I. Marroquim, Ricardo Guerra  
*et al.* II. Universidade Federal do Rio de Janeiro, COPPE,  
Programa de Engenharia de Sistemas e Computação. III.  
Título.

*A Deus, à minha esposa e aos  
meus pais.*

# Agradecimentos

Primeiramente gostaria de agradecer a Deus por ter me dado a vida e ter me permitido chegar a esse momento. Quero agradecer à minha esposa Rosa pelo imenso apoio desde o início da minha caminhada neste doutorado. Por ter suportado a distância em tantos momentos, por sempre ter tido tanta paciência e dedicação, por não medir esforços para me ajudar, colaborado na elaboração de figuras, participando de todo o processo de colagem dos marcadores na vela e, principalmente, pelo companheirismo e amor em cada instante. Agradeço aos meus pais Maurílio e Maria Aparecida pelo apoio e incentivo ao longo de toda a minha vida.

Agradeço aos meus orientadores Ricardo e Marcelo pela paciência, apoio e valiosa orientação para o desenvolvimento desse projeto. Gostaria de agradecer ao aluno de iniciação científica Kevyn pela ajuda na realização dos experimentos. Agradeço também ao Instituto de Educação, Ciência e Tecnologia de Minas Gerais, *campus* Ouro Branco, pelo apoio, permitindo que eu me dedique aos trabalhos do doutorado. Agradeço a todos dos laboratórios de pesquisa LCG/UFRJ e GCG/UFJF pelo apoio e suporte dado ao longo da pesquisa. Gostaria de agradecer aos profissionais da área de engenharia naval, Prof. Alexandre e ao Jorge, que nos apoiaram e nos ajudaram no desenvolvimento desse projeto com o seu conhecimento e experiência e nos permitiram realizar os experimentos reais.

Enfim, gostaria de agradecer aos meus amigos, aos funcionários da COPPE/UFRJ e a todos que de alguma forma contribuíram para o desenvolvimento deste trabalho.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

## RECONSTRUÇÃO MONOCULAR 3D DA FORMA DE VELAS USANDO MARCADORES PASSIVOS

Luiz Maurílio da Silva Maciel

Abril/2018

Orientadores: Ricardo Guerra Marroquim  
Marcelo Bernardes Vieira

Programa: Engenharia de Sistemas e Computação

Este trabalho apresenta um método para recuperar a forma 3D de uma vela usando marcadores passivos. No domínio da navegação e engenharia naval, obter a forma da vela pode ser extremamente valioso para confirmar ou refutar o resultado de simulações, e contribuir no projeto de novas velas otimizadas. O sistema de aquisição proposto é muito simples e de baixo custo, sendo necessário somente fixar uma série de marcadores na superfície da vela e registrar, de um barco auxiliar, a forma da vela durante condições reais de velejamento utilizando uma única câmera. A forma média da vela é reconstruída durante um intervalo de tempo de poucos segundos no qual o velejador mantém a vela o mais estável possível, ou seja, a vela está inflada e sua forma praticamente não muda. Essa média é melhorada por um algoritmo de *Bundle Adjustment* (BA). A reconstrução é esparsa, uma vez que poucos pontos na superfície da vela são suficientes para que os engenheiros navais estimem a sua forma. O método proposto foi testado no cenário real de navegação e apresentou resultados promissores. Quantitativamente, verifica-se a precisão com relação a área dos marcadores reconstruídos e os pontos reprojatados. Qualitativamente, apresenta-se o *feedback* dos especialistas do domínio que avaliaram os resultados e confirmaram a utilidade e qualidade da forma reconstruída.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

## MONOCULAR 3D RECONSTRUCTION OF SAIL FLYING SHAPE USING PASSIVE MARKERS

Luiz Maurílio da Silva Maciel

April/2018

Advisors: Ricardo Guerra Marroquim

Marcelo Bernardes Vieira

Department: Systems Engineering and Computer Science

We present a method to recover the 3D flying shape of a sail using passive markers. In the navigation and naval architecture domain, retrieving the sail shape may be of immense value to confirm or contest simulation results, and to aid the design of new optimal sails. Our acquisition setup is very simple and low cost, as it is only necessary to fix a series of printable markers on the sail and register the flying shape in real sailing conditions from a side vessel with a single camera. We reconstruct the average sail shape during a interval of a few seconds, where the sailor maintains the sail as stable as possible, i.e, the sail is inflated and its shape practically does not change over time. The average is further improved by a Bundle Adjustment (BA) algorithm. Our reconstruction is sparse since only a few points on sail surface are enough for the naval architects to estimate its shape. We tested our method in a real sailing scenario and present promising results. Quantitatively, we show the precision in regards to the reconstructed markers area and the reprojected points. Qualitatively, we present feedback from domain experts who evaluated our results, and confirmed the usefulness and quality of the reconstructed shape.

# Contents

<b>List of Figures</b>	<b>x</b>
<b>List of Tables</b>	<b>xiii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Boat terminology . . . . .	2
1.2 Problem statement . . . . .	2
1.3 Objectives . . . . .	3
1.4 Work Organization . . . . .	3
<b>2 Literature review</b>	<b>5</b>
2.1 Deformable surface reconstruction . . . . .	5
2.1.1 Discussion . . . . .	14
2.2 Sail shape reconstruction . . . . .	15
<b>3 Proposed method</b>	<b>21</b>
3.1 Markers fixation . . . . .	22
3.2 Capture . . . . .	24
3.3 Detection . . . . .	25
3.4 Registration . . . . .	27
3.4.1 Filtering markers with RANSAC . . . . .	28
3.5 Reconstruction . . . . .	29
3.5.1 Bundle adjustment . . . . .	30
<b>4 Experiments</b>	<b>32</b>
4.1 Marker detection tests . . . . .	32
4.2 Rigid object tests . . . . .	38
4.3 Sail reconstruction tests . . . . .	39
4.3.1 Sail video dataset . . . . .	39
4.3.2 General parameters evaluation . . . . .	42
4.3.3 Reconstruction results . . . . .	44
4.3.4 Qualitative feedback from domain expert . . . . .	54

<b>5 Conclusion</b>	<b>59</b>
<b>Bibliography</b>	<b>61</b>

# List of Figures

1.1	Boat diagram indicating the terminology of some parts. . . . .	3
2.1	JOJIC and HUANG [1] cloth reconstruction. . . . .	5
2.2	PRITCHARD and HEIDRICH [2] cloth reconstruction example. . . . .	6
2.3	SALZMANN <i>et al.</i> [3] surface reconstruction method. . . . .	7
2.4	HAYASHI <i>et al.</i> [4] surface reconstruction using markers and depth camera. . . . .	8
2.5	PILET <i>et al.</i> [5] surface reconstruction method. (a) Examples of non penalized deformations. . . . .	9
2.6	HILSMANN and EISERT [6] surface tracking example. . . . .	9
2.7	Inextensible object deformation [7]. . . . .	10
2.8	Functions for modeling template-based reconstruction [8]. . . . .	10
2.9	Result of NGO <i>et al.</i> [9] reconstruction. . . . .	11
2.10	Template acquisition step of YU <i>et al.</i> [10] method. . . . .	11
2.11	LIU and CHEN [11] surface reconstruction method. . . . .	13
2.12	Setup of CLAUSS and HEISEN [12]. . . . .	16
2.13	Setup of LE PELLEY and MODRAL [13]. . . . .	17
2.14	(a) Wind Tunnel setup of GRAF and MÜLLER [14]. (b) Full scale setup of MAUSOLF <i>et al.</i> [15]. . . . .	18
2.15	TOF technology and device used by FOSSATI <i>et al.</i> [16]. . . . .	18
2.16	Setup of DEPARDAY <i>et al.</i> [17]. . . . .	19
2.17	Setup of FERREIRA <i>et al.</i> [18]. . . . .	20
3.1	Proposed method steps. . . . .	21
3.2	Marked sail. . . . .	23
3.3	Markers fixation process. . . . .	23
3.4	Markers graph. . . . .	24
3.5	Example of false positive markers. . . . .	26
3.6	Camera and sail dynamics during different time frames. . . . .	27
4.1	Planar pattern built for accuracy tests of the ARToolKit. . . . .	33
4.2	Axes defined for pattern rotation. . . . .	33



4.3	Detection test situations. . . . .	34
4.4	Comparison between mean distance errors achieved by ARToolKit and ArUco libraries. . . . .	35
4.5	Comparison between mean projected error achieved by ARToolKit and ArUco libraries. . . . .	35
4.6	Comparison between mean normal error achieved by ARToolKit and ArUco libraries. . . . .	36
4.7	Comparison between mean tangent error achieved by ARToolKit and ArUco libraries. . . . .	36
4.8	Examples of markers with different number of internal bits. . . . .	37
4.9	Examples of markers with different number of border bits. . . . .	37
4.10	Rigid objects used for preliminary tests. . . . .	38
4.11	Deformable plastic used for preliminary tests. . . . .	39
4.12	Dataset video frames examples. . . . .	40
4.13	Classification of the reflection. . . . .	40
4.14	Sail shape variation due wind changing for clip “near_4k_08.mp4”. . . . .	42
4.15	Examples of “Too far” and “Bad angle” frames. . . . .	43
4.16	Visualization of the reconstruction of the clip “near_4k_17.mp4” centered in the frame 457 from two views. . . . .	44
4.17	Area error in function of total of selected frames. . . . .	46
4.18	Reprojected error in function of total of selected frames. . . . .	46
4.19	Reconstruction ratio in function of total of selected frames. . . . .	46
4.20	Area error in function of skip size. The max error scale is indicated by the right vertical axis. . . . .	47
4.21	Reprojected error in function of skip size. . . . .	47
4.22	Reconstruction ratio in function of skip size. . . . .	48
4.23	Histogram of the markers area distribution. . . . .	48
4.24	Reconstruction ratio by frame for clip “near_4k_17.mp4”. . . . .	49
4.25	Reconstructed points reprojected on the central frame 457 for clip “near_4k_17”. The red rectangle highlighted shows a region where the natural illumination obfuscate some markers. . . . .	50
4.26	Sail markers rigid motion in relation to the hull markers. The reconstructions central frames are separated by 0.5 seconds (frames 457 and 472). The movement is consistent with the expected behavior, that is, larger motion at the top of the sail. . . . .	51
4.27	Visualization of the reconstruction of the clip “near_4k_08.mp4” centered in the frame 219 from two views. . . . .	52
4.28	Reconstructed points reprojected on the central frame 219 for clip “near_4k_08”. . . . .	53

4.29	Area error criterion comparison between “near_4k_17.mp4” and “near_4k_08.mp4” clips. . . . .	53
4.30	Reprojected error criterion comparison between “near_4k_17.mp4” and “near_4k_08.mp4” clips. . . . .	54
4.31	Visualization of the reconstruction of the clip “far_4k_14.mp4” centered in the frame 246 from two views. . . . .	55
4.32	Reconstructed points reprojected on the central frame 246 for clip “far_4k_14”. . . . .	56
4.33	Area error statistics comparison between “near_4k_17.mp4” and “far_4k_14.mp4” clips. . . . .	56
4.34	Reprojected error statistics comparison between “near_4k_17.mp4” and “far_4k_14.mp4” clips. . . . .	57
4.35	Reconstructed points reprojected on the central frame 205 for clip “far_4k_14”. . . . .	57
4.36	Visualization of the reconstruction of the clip “near_4k_07.mp4” centered in the frame 205 from two views. . . . .	58
4.37	Profile of the sail sections. The lower curve in red presented some distortion, but we were unable to precise its source. . . . .	58

# List of Tables

2.1	Features of the deformable surface reconstruction works. . . . .	15
2.2	Features of the sail reconstruction works. . . . .	20
4.1	Sail dataset video features. . . . .	41

# Chapter 1

## Introduction

Reconstructing 3D surfaces of real objects is a challenging problem that has attracted the attention of many researchers in the past years, and has several applications, such as in: medicine, entertainment, cultural heritage, virtual clothing, and engineering, to name a few. In this work, we focus our attention on the naval architecture application domain, more specifically, on recovering the shape of sail over time during navigation. Designing and manufacturing sailboats require expertise and precision. For racing boats it is an even more critical aspect, as slight differences on the sail shape may give the winning edge to a team.

There are several tools to help designers to project new sails, for example, there are software packages that consider specific conditions such as the pressure on the sail, wind direction and velocity, and sail cloth [19, 20]. It is also possible to simulate the sail behavior numerically using Computational Fluid Dynamics (CFD). Moreover, a controllable test environment can be created in a wind tunnel.

Unfortunately, none of these strategies are able to accurately estimate or predict the sail shape in real sailing conditions, that is, the flying shape. Computer designed sails usually do not exactly match the flying shape. This fact is confirmed by recent publications in this domain, as will be made explicit when the related literature will be discussed in Chapter 2. Likewise, tests performed in the controlled environment of a wind tunnel may not correspond precisely to the true behavior of a real scenario.

During navigation, sailors rely mostly on their own visual feedback and expertise to evaluate the sail performance and propose adjustments. Therefore, retrieving the sail shape in a navigation environment may be paramount to confirm or contest simulation results, and to aid in designing new high performance sails.

Nevertheless, this is a challenging task, as there are many external factors that influence the sail shape, such as wind and the loads applied by the mast and lines. Moreover, some of these factors have high variability in time, and, thus, instantaneous configurations are too noisy for individual analysis. Hence, our goal is to retrieve the average sail shape during a time interval in seconds, in which the sailor

maintains the sail as stable as possible. In other words, we are interested to estimate an mean sail shape during a period in which it is inflated and its shape practically does not change.

It is important to mention that our reconstruction is sparse. Only a few points on the sail surface are enough for the naval architect to estimate the sail shape. In our application, it is much more important to recover a precise average position of just some interest points on the sail surface, than achieving a densely sampled surface without guaranteeing the accuracy for all points.

Briefly, in this work we propose a non-invasive methodology for capturing the sail shape using a single video camera and passive markers. We target a low cost and simple setup to allow for a more general and broad acquisition system. Our method was tested for a Finn class sail but it can be adapted for other sails and boats by changing our capture setup.

Our main contribution is a detailed description of a low cost and simple acquisition system for sail shape reconstruction. We previously verified the precision of the method against a battery of tests using rigid objects. A series of parameters were evaluated in regards to proposed reconstruction metrics for fine tuning the method. Furthermore, the system was tested in real sailing conditions with feedback from domain experts.

## 1.1 Boat terminology

In this section, we introduce the terminology of some parts of the boat that we refer to in this work. These parts are indicated in the boat diagram presented in Figure 1.1:

- the sail edges: leech, luff and foot;
- the spars (poles): mast and boom;
- the boat hull.

Furthermore, the sailboat has several lines, which are the ropes used to perform adjustments in the sail shape and boat direction.

## 1.2 Problem statement

The main problem we want to solve is: given a deformable object, whose behavior is captured over time by a single camera, estimate the mean shape of the object during a time interval of a few seconds. More specifically we want to recover the mean position of a set of predetermined points on the surface.

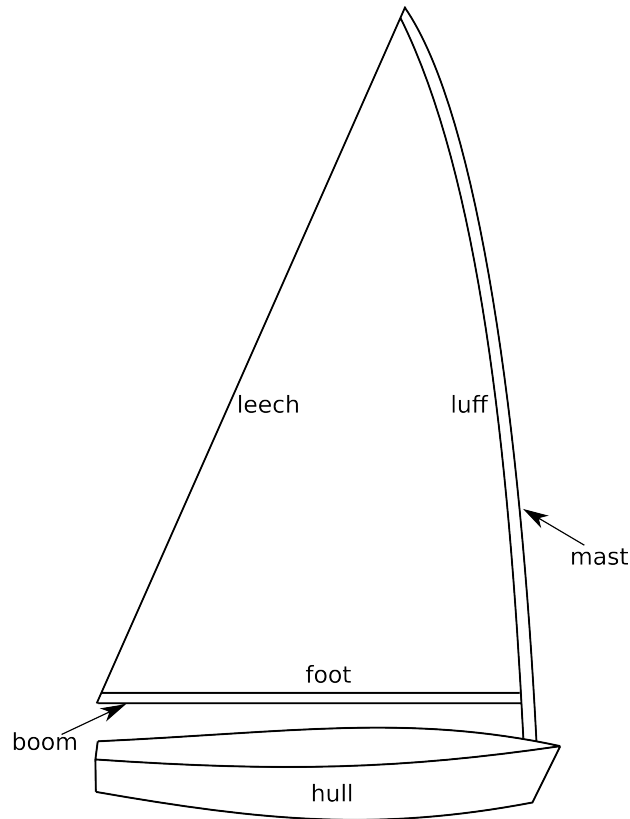


Figure 1.1: Boat diagram indicating the terminology of some parts.

### 1.3 Objectives

The main objective of the present work is to propose a method to retrieve the mean shape of a sail covered by passive markers from a video recorded by a single camera. The achieved reconstruction should be useful for naval architecture purposes. The secondary objectives are:

- to establish the best possible configuration to capture the sail and allow the reconstruction given our problem conditions;
- to filter the extracted data to ensure a good reconstruction;
- to evaluate qualitatively and quantitatively the achieved reconstruction.

### 1.4 Work Organization

The onward part of this work is organized in the following manner:

- **Chapter 2 - Literature review:** works related to the problem of reconstructing deformable surfaces and, more specifically, sails.
- **Chapter 3 - Proposed method:** our proposed method for sail reconstruction.

- **Chapter 4 - Experiments:** tests we performed to understand our problem, propose our method and evaluate qualitatively and quantitatively our proposal. This chapter presents also the domain experts feedback and limitations of our method.
- **Chapter 5 - Conclusion:** the conclusions of the present work and some directions for future works.

# Chapter 2

## Literature review

### 2.1 Deformable surface reconstruction

JOJIC and HUANG [1] presented the first effort to capture cloth motion. They proposed an algorithm that fit the shape of a cloth model with the range data. The cloth can be placed over a supporting object, which is also estimated. The cloth model is calculated through a particle system where each particle interacts with its neighbors through forces, which are determined by the algorithm. Furthermore, the interaction between the cloth particles and the supporting objects is also considered by the particles system. The range data is generated synthetically and scanned using a laser scanner. Figure 2.1 shows a reconstructed cloth.



Figure 2.1: JOJIC and HUANG [1] cloth reconstruction.

CARCERONI and KUTULAKOS [21] introduced a framework for extracting the 3D shape and motion of curves from multiple images sequences simultaneously captured from different viewpoints. They proposed stereo and motion constraints that can be combined into an over-determined system enforcing both constraints. Initial stereo correspondences between curves in different views are established using epipolar geometry. These correspondences are used as input to solve the proposed stereo-motion over-determined system to recover the 3D shape and motion for each curve point independently. They performed experiments on synthetic data with added noise and observed that their proposed method slightly improves the



reconstruction precision when compared to stereo analysis alone. The same authors presented another work [22] that captures the 3D shape, and non-rigid motion of dynamic scenes from multiple views. They use the representation of dynamic surfel (surface element), which is used as a starting point for a set of spatiotemporally-distributed optimization problems to recover scene’s shape, reflectance and motion in a small bounded region of space-time. They solve the problem in two steps: first they achieve the shape and reflectance information and, second, estimate the motion using the previous step results. They performed experiments in several real scenes, such as clothing and skin. The results overcome several limitations of the previous works. However, the method effectiveness was demonstrated only for scenes captured in controlled conditions of illumination and cameras positioning.

PRITCHARD and HEIDRICH [2] proposed to capture cloth motion through stereo correspondence and SIFT [23] features. They use a reference image, which is a flat and undistorted view of the cloth. Correspondences are obtained between this view and each input view. These correspondences are used to estimate an initial parameterization of the geometry. They map the image space (2D) to world space (3D) through a disparity map. Finally, the geometry is completed by interpolating the features in the world space in order to get a dense and regularly sampled surface. They eliminate false correspondences by two constraints: a compression constraint and a stretch constraint. These constraints take into account the degree of extensibility of the cloth, allowing certain compression and stretch. An example of their cloth reconstruction is shown in Figure 2.2.

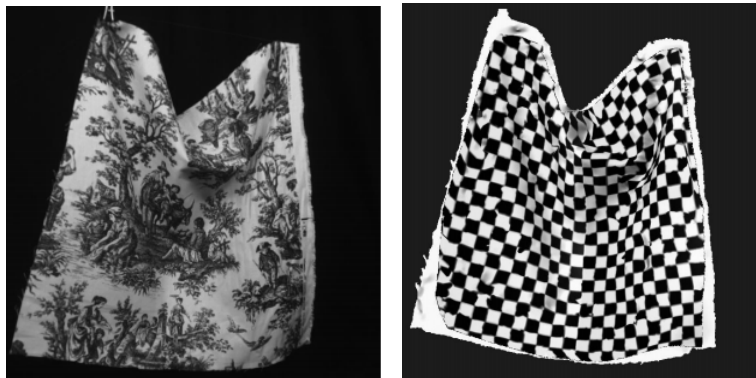


Figure 2.2: PRITCHARD and HEIDRICH [2] cloth reconstruction example.

More recently, SALZMANN *et al.* [3] presented a method for detecting and recovering 3D deformable surfaces from monocular image sequences. The surface is represented by a triangular mesh and the deformations are parameterized in terms of a small subset of the angles between the faces. Based on this parameterization, a set of samples is drawn by randomly choosing the angles between the faces edges. The dimension of this set is reduced by applying Principal Component Analysis (PCA), getting the principal components or modes. Figure 2.3a shows examples of these

modes. Thus, the shape of a mesh can be described as a sum of the undistorted mesh plus a weighted sum of the modes. The shape of the deformed mesh is recovered by an optimization process, where the weights of each mode are calculated in order to minimize objective functions associated to model-to-image point correspondences, image-to-image point correspondences, and contour information. It is important to notice that they introduce a penalty term to force constancy of edge lengths to surfaces that do not stretch. An example of a T-shirt reconstruction with this method is shown in Figure 2.3b. An extension of their work was presented by SALZMANN and FUA [24]. In this extension, the mesh is subdivided into sets of overlapping patches which are combined linearly to model the deformation. The constraints are presented as distance equalities, when the distance between neighbors vertices is kept constant after the deformation, or as distance inequality, when the distance is bounded above by the distance onto the original mesh.

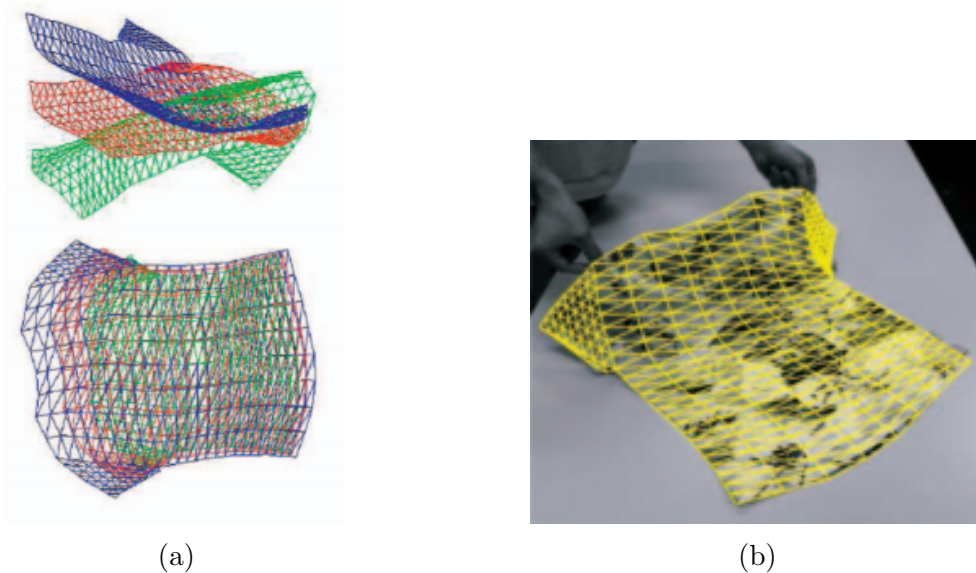


Figure 2.3: SALZMANN *et al.* [3] surface reconstruction method. (a) Examples of modes. (b) T-shirt reconstruction.

HAYASHI *et al.* [4] applied the angle parameterization of SALZMANN *et al.* [3] in their method for overlaying a texture onto the surface of a deformable object. They used a depth camera to retrieve the scene information. In an off-line step they generated deformation models using the approach of SALZMANN *et al.* [3]. The on-line stage consists in estimating the surface and overlaying this surface with a template texture. For this purpose, they sampled points from the cloud provided by the depth camera and computed the weight of each model generated in the off-line phase in order to adjust the sampled points. After finding these weights, the surface can be reconstructed and the texture can be overlaid on the object. It is important to note that they used passive markers to perform the sampling, but their use is

restricted to bound the region of interest. In Figure 2.4, we can see these markers and the reconstructed surface. SHIMIZU *et al.* [25] presented a version that did not use markers but a textured region. The region was detected and tracked by a modified SIFT matching algorithm adapted for deformable objects.

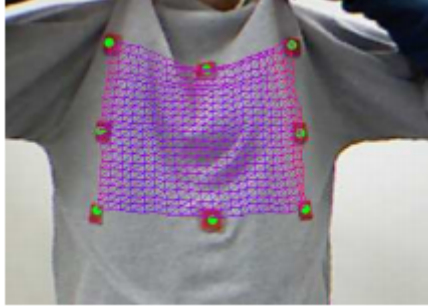


Figure 2.4: HAYASHI *et al.* [4] surface reconstruction using markers and depth camera.

PILET *et al.* [5] proposed a method for detecting deformable surfaces in real time and apply a new texture over the non-rigid object. The input of the method is an image of the object in an undeformed pose and a monocular image sequence of the object. They establish correspondences between the current image and the undeformed pose image. Given the correspondences, they find the transformation mapping the undeformed surface model into the deformed one, minimizing the distances between correspondences and smoothing the deformations. For this propose, they represent the model by a mesh of triangles hexagonally connected and propose an energy which should be minimized to find the correct deformation. The terms of the energy penalize transformations that are not caused by rigid motion. For example, vertices that do not remain roughly equidistant in two collinear connected edges are penalized. Figures 2.5a and 2.5b show examples of non penalized and penalized deformations, respectively. Consequently, erroneous correspondences are removed and the resulting transformation is achieved. Figure 2.5c shows an example of a reconstruction.

In order to create a virtual mirror, HILSMANN and EISERT [6] proposed a method for tracking and retexturing cloth from monocular image sequences. The detection of a shirt is achieved by segmenting the image and finding a rectangular highly textured region. This detection process considers color information. After detecting the shirt, tracking is performed using optical flow constraints combined with mesh-based motion. They represent the surface as a triangulated regular mesh, where each point is represented by its barycentric coordinates. In order to maintain the mesh regularity they add a constraint that the barycentric coordinates of a point are preserved after vertex displacement. Furthermore, they handle the self-occlusion problem by estimating an occlusion map and forcing the mesh to shrink

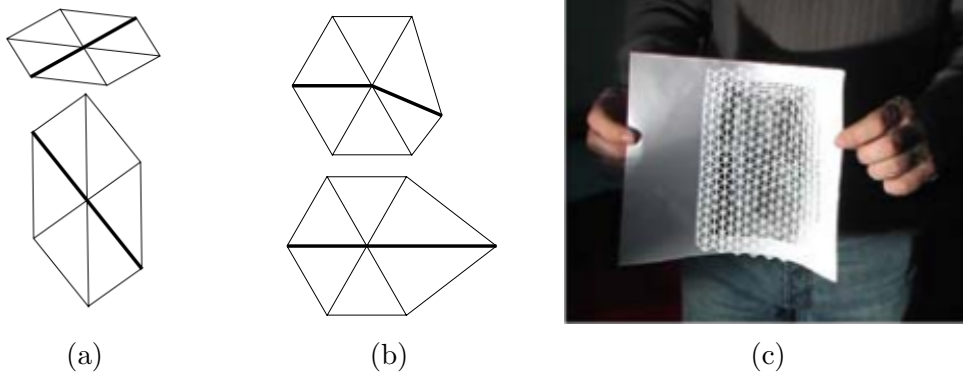


Figure 2.5: PILET *et al.* [5] surface reconstruction method. (a) Examples of non penalized deformations. (b) Examples of penalized deformation. (c) Reconstruction example.

instead of folding at the occluded regions. Finally, they recover the illumination and shading from the original cloth and overlay a virtual texture. An example of surface detection and reconstruction of HILSMANN and EISERT [6] is shown in Figure 2.6.

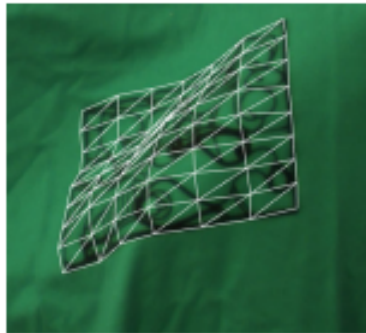


Figure 2.6: HILSMANN and EISERT [6] surface tracking example.

BRUNET *et al.* [7] presented methods to reconstruct inextensible surfaces from monocular image sequences. They find correspondences between a template image and an input image. First, since they know the intrinsic camera parameters beforehand, they propose a convex formulation for maximizing the depth of the reconstructed 3D points subject to the inextensibility constraint, which forces the Euclidian distance between two points to be never greater than the length of the corresponding geodesic distance measured in the template image. Figure 2.7 illustrates an inextensible deformation. They consider noise in both the template and the input images. Second, they propose an approach based on the principle of surface inextensibility. Thus, the function mapping from the planar template into the 3D space must be everywhere a local isometry. They parameterized the surface and formulated a least-square problem including a data, an inextensibility, and a smoothing term to find the reconstructed surface.

Another monocular reconstruction was proposed by PERRIOLLAT *et al.* [26].

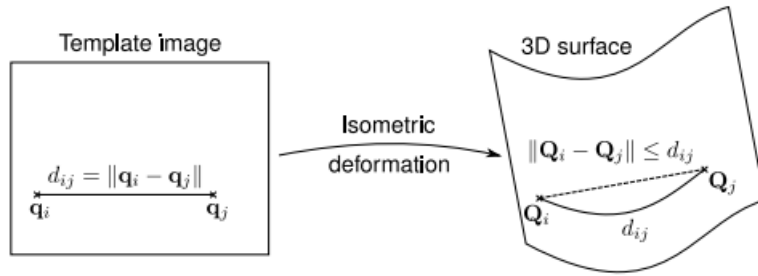


Figure 2.7: Inextensible object deformation [7].

They present an algorithm to reconstruct 3D inextensible deformable surfaces, based on correspondences between a 3D template registered from an image of the object and the current image of the object. They propose to find upper bounds on the surface depth. The upper bound of the depth of a pair of points can be estimated from the projection camera matrix by the inextensibility constraint. This way, the bounds initially estimated are refined. Finally, the surface is recovered in two steps: 3D points are found from the upper bound and these points are interpolated to retrieve the final surface.

BARTOLI *et al.* [8] presented analytical solutions for the problem of monocular 3D deformable surface recovery. The geometry is modeled by five functions, which are shown in Figure 2.8. In order to find the unknown functions the problem is mathematically modeled as a differential problem, whose constraints are different for developable, isometric or conformal surface. These differential formulations allow to find the functions and reconstruct the surface. They proved the solution for each surface and note that isometric (developable or non developable) surfaces can be reconstructed unambiguously and conformal surfaces can be reconstructed up to a global scale and a few discrete ambiguities. They presented interesting analytical solutions for the monocular reconstruction problem.

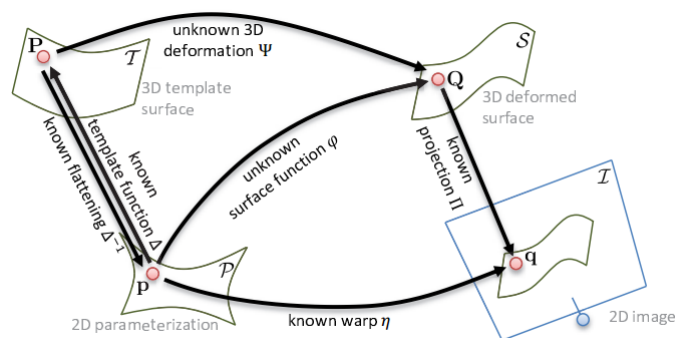


Figure 2.8: Functions for modeling template-based reconstruction [8].

Recently, NGO *et al.* [9] introduced a new method for reconstructing a 3D deformable surface from monocular images. The linear formulation of the problem

of monocular reconstruction based on correspondences between a template image and the current image is ill-posed. They used the Laplacian formalism to turn this linear formulation better-posed. The correspondences are established by generic features. The mesh is parameterized by some control points. The configuration of the mesh is estimated by adding a constraint that penalizes non rigid transformations. They also used inextensibility constraints that bound the distance between neighbor vertices. Figure 2.9 shows a result of their reconstruction.

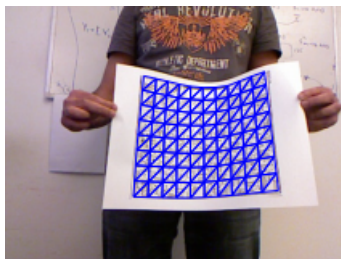


Figure 2.9: Result of NGO *et al.* [9] reconstruction.

The method proposed by YU *et al.* [10] reconstructs densely a 3D deformable surface from a single camera. Although, they used a stereo method in an initial step to compute a template of the object. They acquire the images by keeping the object fixed and moving the camera, or keeping the camera static and the object is rigidly moved. This step is shown in Figure 2.10. Once the template is computed, the tracking of the surface is performed for each frame by minimizing an energy. This energy is composed by four terms: a photometric error, which minimizes the difference of appearance between the current image and the template shape; a regularization term, which aims to smooth deformations in the spatial sense; an as rigid as possible term, which penalizes non-rigid transformations of the shape; and a temporal term, which smooths the deformation over time. Their approach is robust against occlusions.

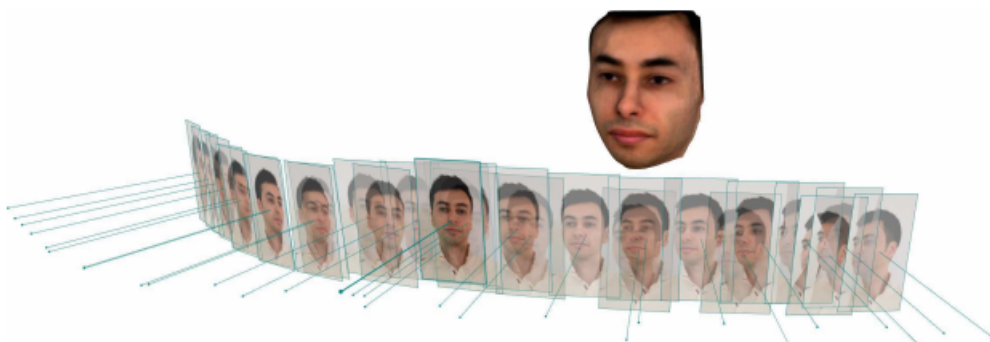


Figure 2.10: Template acquisition step of YU *et al.* [10] method.

A well quoted multi-view 3D reconstruction method was proposed by FURUKAWA and PONCE [27]. From images of different views, they detect generic

features and match some candidate correspondences. A small rectangular oriented patch is constructed for each pair of features. The initial patches are optimized and expanded in order to cover a dense region of the object. Finally, the filtering is performed to eliminate erroneous patches. The authors show that a polygonal mesh model can be built from the resulting set of patches. They estimate the orientation from the views correspondences.

In their work, FABRI and KIMIA [28] proposed a framework for multi-view 3D reconstruction based on curve correspondences. They started from a set of images from different views and an approximation of the calibration of each camera. After extracting the edges of the images, correspondences are established among curves in different views. These correspondences are used to construct a partial 3D curve sketch which is the input for the next stage. In this stage, the calibration is refined minimizing a curve-based reprojection error. Once the refined camera is obtained, the 3D curve sketch can be complemented by curve fragments that were not included in the first sketch. This curve sketch approach was improved by USUMEZBAS *et al.* [29], where a 3D drawing from the curves is generated. They identify and integrate information across related edges. The curves are joined based on topological connectivity resulting in a 3D graph, i.e, the 3D drawing. Recently, USUMEZBAS *et al.* [30] proposed a method to reconstruct surfaces from pairs of curves of the 3D drawing using lofting algorithms. The pairs are selected based on length, proximity and curvature, and the hypothesis viability is verified by a occlusion consistency.

Another multi-view reconstruction method was presented by LIU and CHEN [11]. They proposed a method to estimate simultaneously the 3D structure and the motion of moving surfaces. They started by establishing point correspondences across time and views using Scale-Invariant Feature Transform (SIFT) features. These correspondences are used to estimate an initial 3D triangulation. The surface is locally modeled as a 9-parameter spatio-temporal plane. They perform a region growing mechanism and optimize the local plane parameters based on the assumption that if two points are close, their depths and normals are also close to each other. This optimization is performed by minimizing an energy that considers the homography between views and the rigid transformation of the points over time. Figure 2.11 shows an example of surface and flow computed by their method. LIU and CHEN [31] presented a follow up of this work. The main difference between the previous work [11] is that they minimize the energy first considering the homography between views and use the result to minimize the energy considering the rigid transformation over time. This new approach results in a cleaner motion field than LIU and CHEN [11].

BLANZ *et al.* [32] introduced an algorithm to fit a Morphable Model to scans



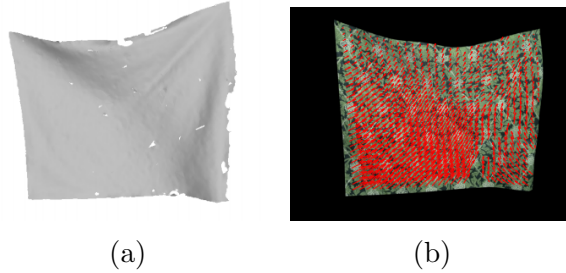


Figure 2.11: LIU and CHEN [11] surface reconstruction method. (a) Example of reconstructed surface. (b) Example of computed motion field.

of faces, which simultaneously optimizes shape, texture, pose and illumination. Initially, from a training set of relatively high quality laser scans, they build a Morphable Model by performing a PCA. In the sequence, given an input of a scan, containing the texture and cartesian coordinates of each image point, they perform an optimization to find the shape, the texture and the rigid pose transformation, minimizing the color image differences and the surface distances to the Morphable Model. Furthermore they optimize the camera parameters and the lighting, minimizing the effect of overexposure and inhomogeneous shading of the image input.

ZOLLHÖFER *et al.* [33] addressed the problem of reconstructing a non-rigidly deformable object in real time using a RGB-D camera. They built their own RGB-infrared sensor, which provides more flexibility than consumer ones. They modeled the object as a hierarchical triangle mesh. From the images and depths provided by the camera, they optimized an energy function composed by four terms: a point term, which evaluates the distance of each vertex to the closest data point; a plane term, which measures the conformity of the vertex to its plane; a color term, which evaluates the difference between the intensities of the projected vertex on the image at current and previous timestep; and a regularizer term, which penalizes non-rigid transformations, forcing local deformations of the surface to be as close as possible to isometries. The minimization of the sum of these energy terms, appropriately implemented in GPU, allows the real time reconstruction of the objects.

NEWCOMBE *et al.* [34] proposed a method for reconstructing scenes containing non-rigidly deforming objects, using a RGB-D camera. The scene motion is represented by a volumetric warp-field, where each point has a rotation and a translation associated. The field is estimated for each frame by minimizing an energy composed by a data term, which takes into account the plane point distance and a regularizer term, which smooths the surface over time. The regularizer term takes into account the temporal information penalizing non-rigid transformations over time. An efficient implementation ensures a dense reconstruction of the scene.

BRONSTEIN *et al.* [35] presented a review of non-Euclidian geometric problems and the solutions for these problems using deep learning techniques. They intro-



duced the non-Euclidian geometry of the manifolds and graphs structures and generalized Convolutional Neural Network(CNN) type constructions to these domains. The learning methods that can be applied to graphs and manifolds are mathematically described showing the solutions in the frequency, spatial and spatio-frequency domains. In terms of manifold structures, the problem of adjust and matching of meshes is appointed as a possible application for the geometric deep learning field.

### 2.1.1 Discussion

JOJIC and HUANG [1] presented the first effort to capture cloth motion using a particle system. Thereafter, several other works were proposed to reconstruct 3D deformable surfaces. As with our approach, some of them opted for a monocular reconstruction [3, 5–10, 24, 26], others proposed multi-view approaches [2, 11, 21, 22, 28, 31], while some made use of RGB-D devices [4, 25, 32–34]. The use of multiple cameras and RGB-D devices may indeed improve the acquisition performance and precision, but renders the system more complex and costly, going against our goal to keep the system as simple and low-cost as possible.

Most of the proposed works perform the reconstruction based on generic features, such as SIFT, extracted from the images [2, 3, 5, 7–11, 24–26, 31], requiring that the object presents a highly textured surface with distinguishable elements. In our work, we opted for passive markers, which are more easily and accurately detectable, and mostly important, most sails have practically uniform textures and do not allow for a straightforward extraction of generic features. Furthermore, passive markers allow identifying and labeling specific points on the sail surface, an important feature for our application. HAYASHI *et al.* [4] do use colored markers, but only to bound the region of interest on the object, and the surface is actually reconstructed based on a RGB-D device information.

Some works explore the inextensibility property of certain objects, by adding constraints of equality and/or inequality between points on the object surface [2, 3, 7–9, 24, 26]. A more specific study of the elasticity of the sail would be necessary to evaluate the application of this kind of restriction to our problem. Nevertheless, high-performance sails are highly customizable and do not have an unique elasticity behavior [36, 37], hence, such specific evaluation goes far beyond the scope of the current work.

Other reconstruction approaches apply a temporal smooth constraint [10, 11, 21, 22, 31–34]. This constraint assumes the object deforms as minimum as possible over time and is often accompanied by an as rigid as possible constraint [5, 9–11, 31–34], which penalizes non-rigid transformations. For our sail reconstruction, we assume the sail shape is constant over time, presenting only a rigid transformation

between frames. Note, however, that we do not know how the shape deformed concerning its resting state, and does we cannot discard its extensibility effect. These rigid transformations are used to perform a registration between different frames (Section 3.4). There are works that employ machine learning methods, such as PCA [3, 4, 24, 25, 32] and deep learning [35], to retrieve the surface deformation. Since we do not have enough data on sails configurations to apply learning strategies, such approaches are not possible for our problem at this moment.

Table 2.1 summarizes the choices made in the presented works and compare to our proposed method, which is presented in Chapter 3.

	Monocular	Multi-view	RGB-D	Generic features	Passive markers	Color information	Temporal smooth	Physical model	Distance constraint	Reference image	As rigid as possible	Angle parameterization	Depth maximization	Curves	Bundle adjustment	Machine learning
JOJIC and HUANG [1]								•								
CARCERONI and KUTULAKOS [21]		•					•			•				•		
CARCERONI and KUTULAKOS [22]		•				•	•			•						
PRITCHARD and HEIDRICH [2]		•		•					•	•						
SALZMANN <i>et al.</i> [3]	•			•					•	•		•				•
SALZMANN and FUA [24]	•			•					•	•		•				•
HAYASHI <i>et al.</i> [4]			•		•	•						•				•
SHIMIZU <i>et al.</i> [25]			•	•								•				•
PILET <i>et al.</i> [5]	•			•						•	•					
HILSMANN and EISERT [6]	•					•										
BRUNET <i>et al.</i> [7]	•			•					•	•			•			
PERRIOLLAT <i>et al.</i> [26]	•			•					•	•			•			
BARTOLI <i>et al.</i> [8]	•			•					•	•						
NGO <i>et al.</i> [9]	•			•					•	•	•					
YU <i>et al.</i> [10]	•			•		•	•				•					
FURUKAWA and PONCE [27]		•		•						•						
FABBRI and KIMIA [28]		•							•					•		
USUMEZBAS <i>et al.</i> [29]		•							•					•		
USUMEZBAS <i>et al.</i> [30]		•							•					•		
LIU and CHEN [11]		•		•			•			•	•					
LIU and CHEN [31]		•		•			•			•	•					
BLANZ <i>et al.</i> [32]			•			•	•				•					•
ZOLLHÖFER <i>et al.</i> [33]			•			•	•		•		•					
NEWCOMBE <i>et al.</i> [34]			•				•		•		•					
BRONSTEIN <i>et al.</i> [35]							•									•
Our method	•				•					•	•				•	

Table 2.1: Features of the deformable surface reconstruction works.

## 2.2 Sail shape reconstruction

Besides generic surface reconstruction methods, some specific approaches for sails were introduced in recent years. CLAUSS and HEISEN [12] proposed a method to capture the flying shape of the sails of a sailing yacht DYNA. For this purpose, they applied a set of black square markers on the sail in discrete positions forming a grid. They captured the sail during sailing using six cameras placed on the boat (Figure

2.12). The markers are identified in the images by image processing. The localization of the markers in the 3D space is determined by photogrammetry routines. They used a physical model based on the distance between markers neighbors to correct erroneous or missing markers. After refining the marker positions using the physical model, the surface is constructed using the *Multisurf* modeller software. The resulting surface was applied in Computational Fluid Dynamics (CFD).

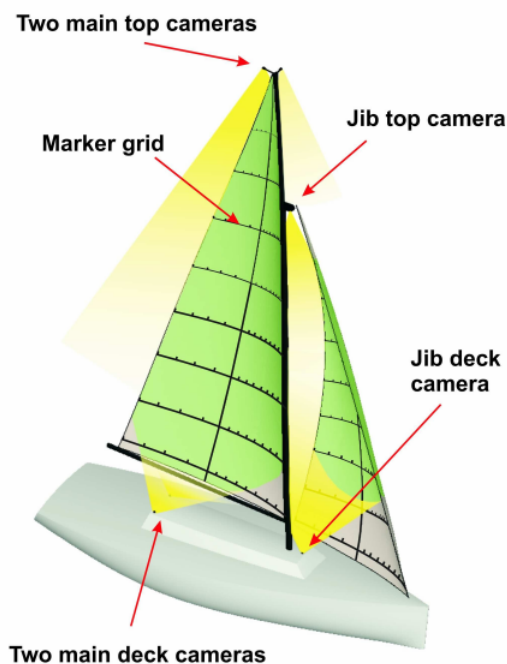


Figure 2.12: Setup of CLAUSS and HEISEN [12]. We can note the marker grid and the internal cameras positioning.

The Visual Sail Position And Rig Shape (VSPARS) software, popular among sail designers, was presented by LE PELLEY and MODRAL [13]. They determine the 3D localization of colored stripes on the sails and colored points on the rig using three cameras fixed on the boat's deck (Figure 2.13). The targets are extracted and their positions in the global coordinate system are estimated based on the hypothesis that the stripes are in the horizontal plane when flying. This is, nonetheless, a strong hypothesis, which is not true for several apparent wind angles, according to some recent works FOSSATI *et al.* [16], DEPARDAY *et al.* [17]. In order to validate the method, they performed tests on wind tunnel using a solid fiberglass and soft sails. They also performed experiments with full scale boats.

GRAF and MÜLLER [14] proposed a method to acquire the flying shape of sails in wind tunnel. The sail is covered by coded passive markers and four cameras are arranged outside the boat (Figure 2.14a). After preprocessing the images, they recover the markers' 3D positions using the *Photo Modeler Pro* photogrammetry software. Finally, the software *Rhinoceros 3D* is used to create NURBS curves from

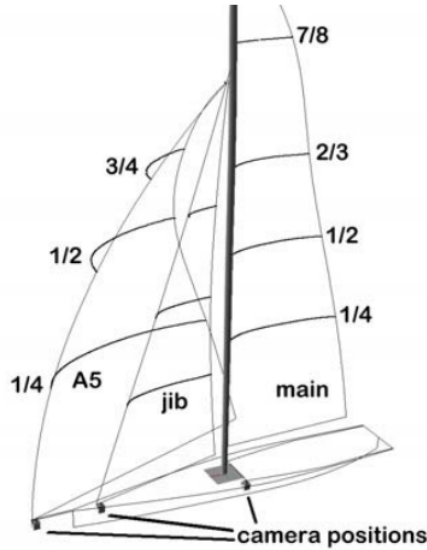


Figure 2.13: Setup of LE PELLE and MODRAL [13]. Observe the cameras positioning on the boat deck and the sail stripes.

the points, which are interpolated in NURBS surfaces. They performed accuracy tests using an object of known shape presenting an average error of approximately 1 mm, and maximum error of 10 mm. Furthermore, they compare the reconstructed shape with the design shape and note meaningful differences, since the flying shape is significantly more asymmetric in comparison to the design shape. MAUSOLF *et al.* [15] extended this work by recovering the flying shape of sails at full scale in real conditions. In order to capture the images, they placed cameras on four tenders around the target boat, moving in approximately the same speed. Figure 2.14b shows the arrangement of the tenders around the boat. They compare the reconstructed shape in a wind tunnel and in full scale and observe a considerable difference, which they attribute to the human factor of sail trimming. More recently, the method of GRAF and MÜLLER [14] was used by RENZSCH and GRAF [38] to estimate the flying shape in a wind tunnel and show the sail movement on consecutive photo sets for two different sails. For both sails the movement occurs mainly at the luff, but the paper does not give further detail on the reconstruction evaluation. They also calculated the forces over the sail and the results were used as a benchmark for Fluid-structure interaction (FSI) simulation.

FOSSATI *et al.* [16] introduced another method to measure flying shapes in wind tunnel at full scale. The method is based on active capture. They use Time Of Flight (TOF) technology, which projects on the object a light in the visible or near infrared range and observe the time of the reflected light (Figure 2.15a). They built a device, which rotates around an axis, brushing the whole sail area (Figure 2.15b). This device retrieves a point cloud, which is used to recover the sail corners, edges and sections. They compute the instantaneous flying shape and

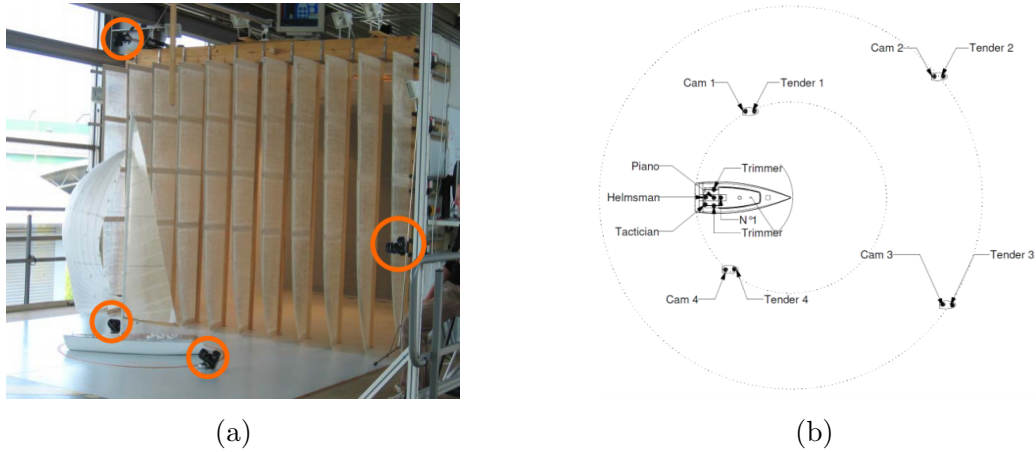


Figure 2.14: (a) Wind Tunnel setup of GRAF and MÜLLER [14]. (b) Full scale setup of MAUSOLF *et al.* [15]. In both scenarios we note the external positioning of the cameras.

point out the possibility of averaging multiple shapes. Precision and accuracy were verified by preliminary tests using known reference objects. The reconstructed sail shape was evaluated comparing the measurements retrieved against those provided by the design sail tool, achieving significant differences. As in MAUSOLF *et al.* [15], these differences were associate to the trim adjustment. Unfortunately, the authors do not explicit any quantitative results in their paper.

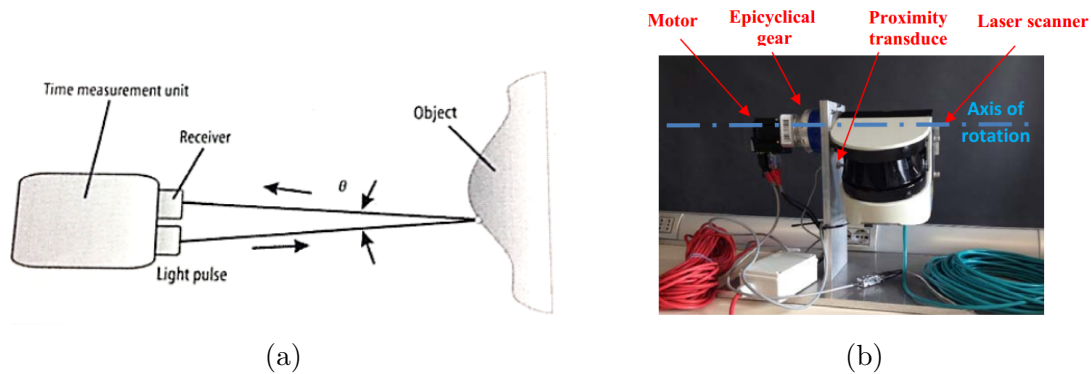


Figure 2.15: TOF technology and device used by FOSSATI *et al.* [16]. (a) Illustration of TOF technology for capture an object. (b) TOF device highlighting the laser scanner and the axis of rotation.

DEPARDAY *et al.* [17] introduced a method to retrieve the shape of sails in full scale while, simultaneously, measuring the aerodynamic load on the corners with navigation and wind data. To recover the sail shape, they fixed blue square markers on the sail forming six equidistant rows. The sail is captured by six cameras located on the boat and synchronized by a laser. The images are delivered to the *Photo Mod-eler* software, which recovers the 3D positions of the markers using photogrammetry algorithms. The calculated points are lofted in splines which in turn are lofted in

a Non-Uniform Rational B-Spline (NURBS) surface. They estimate instantaneous sail shape during a stable period and choose one timestamp as representative of the average shape. The validation of the reconstruction is performed by comparing the retrieved shape and the designed shape. They also observed strong differences in the sail shape, indicating the design is not representative of the real sailing conditions.

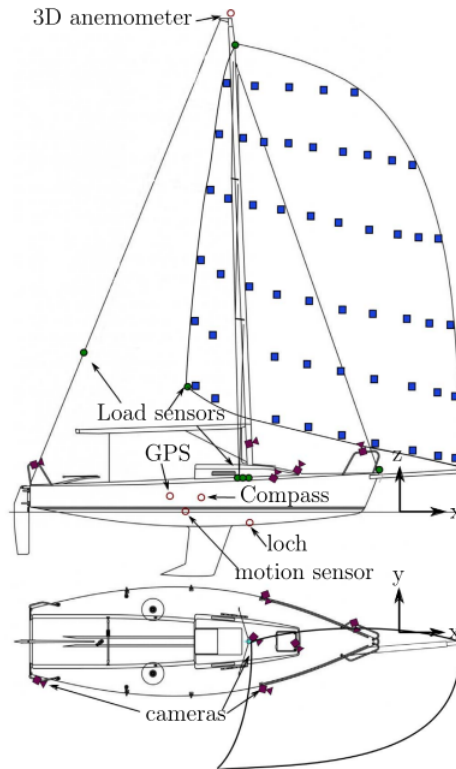


Figure 2.16: Setup of DEPARDAY *et al.* [17]. We can see the internal cameras positioning, the markers grid and the positioning of other aerodynamic load measurer devices.

Recently, FERREIRA *et al.* [18] proposed a method to detect the sail flying shape based on fiber optic strain gauge sensors. They insert such sensors into a set of horizontal sections of the sail and connect them to an optical interrogation unit located in the boat. This unit acquires multiplexed data, which is processed to estimate the curvature of the sections. The estimated curvature may be sent to mobile devices and seen by the sailor in real time. Figure 2.17 illustrates their proposal. They validate their method in lab conditions using a rigid model, but are still studying the influence of the sensors material on the aerodynamic of real flexible sails.

Table 2.2 summarizes the main features of the presented sail reconstruction methods, as well as a confrontation with our proposal. We again draw the attention to one important point communicated in previous work, that is, the significant divergence between the designed shape and the one retrieved in real scenarios by the related methods, reinforcing the need to appropriately and accurately reconstruct

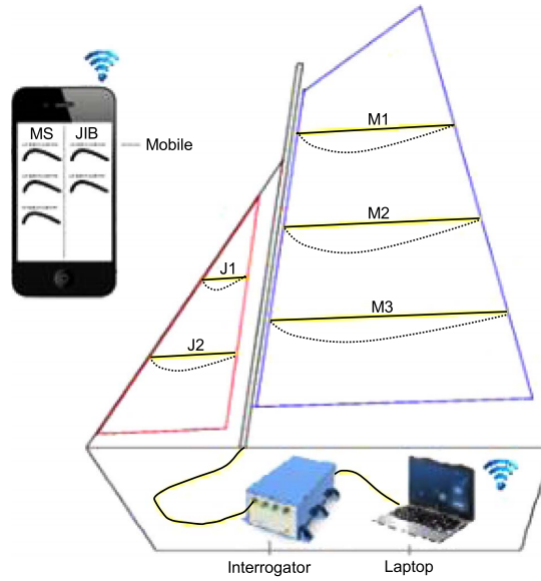


Figure 2.17: Setup of FERREIRA *et al.* [18]. They illustrate the strain sensors and the interrogator device.

the flying shape in such conditions. Another worthy comment is that our method is the only one that works with a single camera, and thus offers a much simpler and generic setup for capturing the sail in a real sailing environment. For boats with large sails or more than one sail, we can separately capture and reconstruct each sail or sail part using more cameras. Moreover, if the wind conditions do not change, we can capture perform each record in different instants using a single camera.

	Monocular	Multi-view	Internal cameras	External cameras	Wind tunnel	Full scale	Black markers	Colored markers	Colored stripes	Coded markers	Photogrammetry	Active capture	Error correction	Strain sensors	Bundle adjustment
CLAUSS and HEISEN [12]		•	•		•	•	•				•		•		
LE PELLEY and MODRAL [13]		•	•		•	•		•	•						
GRAF and MÜLLER [14]		•		•	•					•	•				
MAUSOLF <i>et al.</i> [15]		•		•	•	•				•	•				
RENZSCH and GRAF [38]		•		•	•					•	•				
FOSSATI <i>et al.</i> [16]												•			
DEPARDAY <i>et al.</i> [17]		•	•			•		•			•				
FERREIRA <i>et al.</i> [18]														•	
Our method	•			•		•				•	•		•		•

Table 2.2: Features of the sail reconstruction works.

# Chapter 3

## Proposed method

In this Chapter, we will describe our proposed method for solving the problem of sail shape estimation. As depicted in Figure 3.1, it is composed of five steps, which will be described in the next sections:

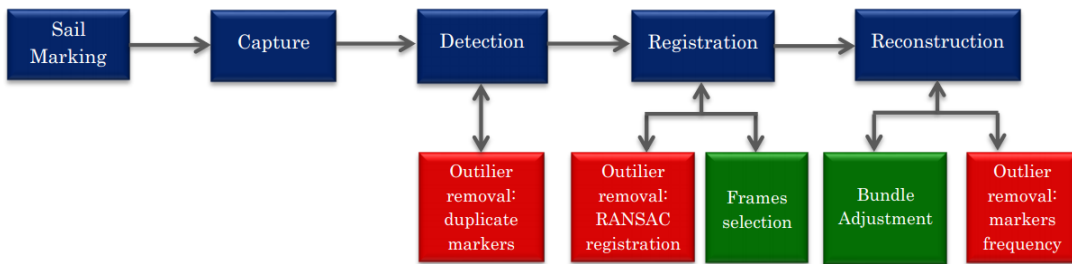


Figure 3.1: Proposed method steps. Blue: main steps. Red: outlier removal internal steps. Green: selection and optimization internal steps.

1. **Markers fixation** (Section 3.1): markers are chosen, printed and fixed on the sail. This disposition should ensure that the markers will be detected and the sail can be properly captured.
2. **Capture** (Section 3.2): the marked sail is captured during a real sailing situation. The capture needs to ensure that the markers can be detected from the images, trying to avoid as best as possible adversary conditions such as strong reflections. Moreover, it is necessary to record from a position that captures the whole sail, since we use a single camera.
3. **Detection** (Section 3.3): markers are extracted from the captured images. Each marker is labeled in order to integrate the temporal information in the next steps based on the correspondences. Duplicate markers elimination is performed by a simple verification of topological consistency. Besides the marker label, the detection step provide its 2D points on the image and the corresponding 3D points in the camera coordinate system.



4. **Registration** (Section 3.4): since each image is captured under a different coordinate system, it is necessary to perform a global registration. In this step, we also select the frames in a given time frame that will be used to estimate the mean sail shape. Furthermore, our registration performs a filtering step to remove outliers.
5. **Reconstruction** (Section 3.5): an average shape of the sail over the previously selected frames is achieved by integrating the registered data. Before the average shape estimation the least frequent markers are removed and are not used to calculate the mean. Furthermore, the average is improved by a Bundle Adjustment (BA) algorithm [39].

One important observation regarding our method is that we reconstruct an average sail shape during a time interval, since instantaneous configuration recovered from single frames are very noisy. During the recording interval the sailor keeps the boat and sail as stable as possible, and we assume that any noise resulting from external forces can be treated as a normal distribution with zero mean, and consequently may be averaged out.

### 3.1 Markers fixation

The first step of our method is to place markers on the sail. We opted for augmented reality markers printed on waterproof adhesives, fixed on one side of the sail surface. As described in Section 4.1, we compared the robustness of detection of two libraries: ARToolKit [40] and ArUco [41]. ArUco presented best results for our tests, and we decided to use markers from this library. For naval architecture purposes, it is important to retrieve horizontal sections along the sail, since they convey well its general shape. In fact, for simulation and design purposes the sail surface is mainly defined by horizontal curves [42]. Thus, markers were placed forming horizontal lines (Figure 3.2) in strategic positions pointed out by the naval architects. We also made a vertical line of markers on the sail, which is important to get an orthogonal orientation of the sail and verify the coherence among the horizontal stripes. Moreover, it is interesting to have a rigid reference for the sail’s markers in order to properly capture the sail behavior over time. For this purpose, some markers were fixed on the hull.

The markers fixation should be performed carefully to avoid losing them during sailing. The markers should tolerate some amount of water, wind and sail deformations. Since the adhesive glue may not be enough to avoid these issues, we fixed scotch tape on the markers border, as presented in Figure 3.3.



Figure 3.2: Marked sail.



Figure 3.3: Markers fixation process.

Once the markers are fixed, their positions on the sail allow to establish an adjacency map. This map defines a graph as a set:

$$G = \{V, E\},$$

where  $V = \{v_i \mid v_i \text{ is the marker with index } i\}$  is the vertices set and  $E = \{e_{ij} \mid e_{ij} \text{ is the edge connecting the vertices } v_i \text{ and } v_j\}$  is the edges set. We established the adjacencies as shown in Figure 3.4: markers on horizontal lines are connected to the markers on right and left; markers on the vertical line are connected to the markers above and below; and markers on the hull are connected to all adjacent markers. This graph is useful to verify the topological coherence and remove duplicate detected markers (Section 3.3). We define the distance between two vertices as the number of edges connecting them. Thus, the smaller the number of edges between two vertices, closer they are. For example, in Figure 3.4 the vertices  $v_j$  and  $v_k$  are the nearest vertices to  $v_i$  because only one edge separates these vertices. In other words,  $v_j$  and  $v_k$  have distance 1 to  $v_i$ . The next nearest vertex is  $v_l$ , which has distance 2 to  $v_i$ .

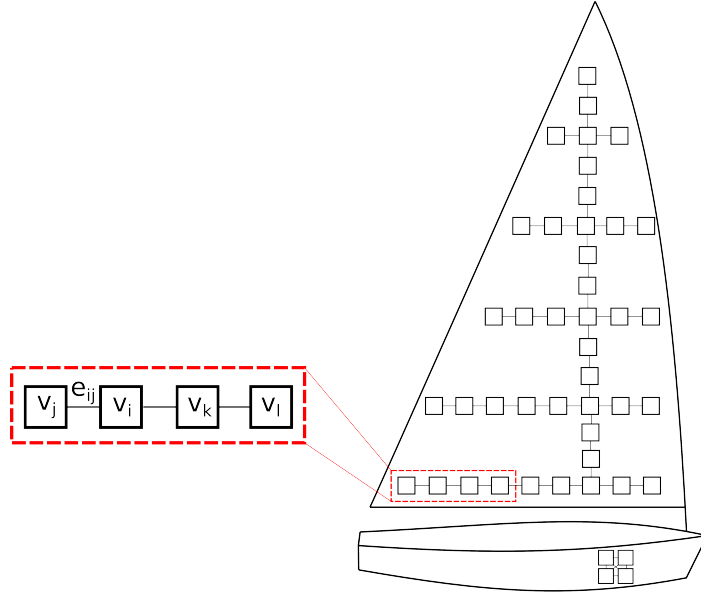


Figure 3.4: Markers graph.

## 3.2 Capture

The next step of our method is to capture a video of the marked sail in real sailing conditions. We use a single camera placed in another boat that follows the target boat from a distance of a few meters. Three to five meters is enough to not affect the sail boat performance, retrieve the markers, and, at the same time, capture the whole sail surface. Alternatively, we could place the cameras inside the target boat. However, this configuration has disadvantages, such as the need for more cameras in order to capture the whole sail surface [12, 13], and the perspective distortion of the images, especially on the sail top [17]. Positioning the cameras in another boat

allows to record the sail at a more perpendicular angle, and is a more generic and simple setup that can be used for a broader range of boats, and does not interfere with the sailing of the tracked boat.

The main challenge of capturing the sail is to keep the camera in a distance that allows a good marker detection while avoiding illumination problems. Since the sailor needs to change the boat direction and the sail position to keep the sail shape, the another boat should also change to track the target boat properly. These changes of positioning makes difficult to keep the ideal capture conditions for a long time.

### 3.3 Detection

After capturing the markers, we move on to identify each one. Given a video frame  $f$ , for a detected marker whose index is  $k$ , its four corners  $\{\mathbf{x}_{k,1}, \mathbf{x}_{k,2}, \mathbf{x}_{k,3}, \mathbf{x}_{k,4}\}$  are extracted in image domain  $\Omega \subset \mathbb{R}^2$ , while its center's transformation (translation  $\vec{\mathbf{t}}_{k,0} \in \mathbb{R}^3$  and rotation  $\mathbf{R}_k \in SO(3)$ ) are recovered in relation to the camera's coordinate system.  $\mathbf{R}_k$  also defines the marker's normal and tangent vectors, while its center position in camera space  $\mathbf{p}_{k,0} \in \mathbb{R}^3$  is directly obtained from  $\vec{\mathbf{t}}_{k,0}$ . Similarly, we can find the corners positions  $\{\mathbf{p}_{k,1}, \mathbf{p}_{k,2}, \mathbf{p}_{k,3}, \mathbf{p}_{k,4}\}$  by a rigid transformation of  $\mathbf{p}_{k,0}$ . Conversely, the image point of the marker's center  $\mathbf{x}_{k,0} \in \Omega$  can be found by projecting  $\mathbf{p}_{k,0}$  onto the image. Thus, for each marker we can define a matrix of 2D points in image coordinates:

$$\mathbf{X}_{k,f} = \begin{bmatrix} \mathbf{x}_{k,0} & \mathbf{x}_{k,1} & \mathbf{x}_{k,2} & \mathbf{x}_{k,3} & \mathbf{x}_{k,4} \end{bmatrix}^T,$$

and a matrix of their respective 3D points in camera coordinates:

$$\mathbf{P}_{k,f} = \begin{bmatrix} \mathbf{p}_{k,0} & \mathbf{p}_{k,1} & \mathbf{p}_{k,2} & \mathbf{p}_{k,3} & \mathbf{p}_{k,4} \end{bmatrix}^T.$$

Therefore, a marker of index  $k$  detected at frame  $f$  can be defined by the pair:

$$M_{k,f} = (\mathbf{X}_{k,f}, \mathbf{P}_{k,f}).$$

Commonly, false positives arise during detection. Artifacts on the image may be confused with a marker, and markers may be mislabeled, as shown in Figure 3.5. In order to simplify our process, we use markers with unique indices  $k$ , i.e., any marker detected more than once clearly indicates a detection error.

We identify and remove the duplicate markers using topological constraints, which are based on the graph defined in Section 3.1. For each index  $k$  and frame  $f$ , we have a set of candidates markers  $C_{k,f} =$

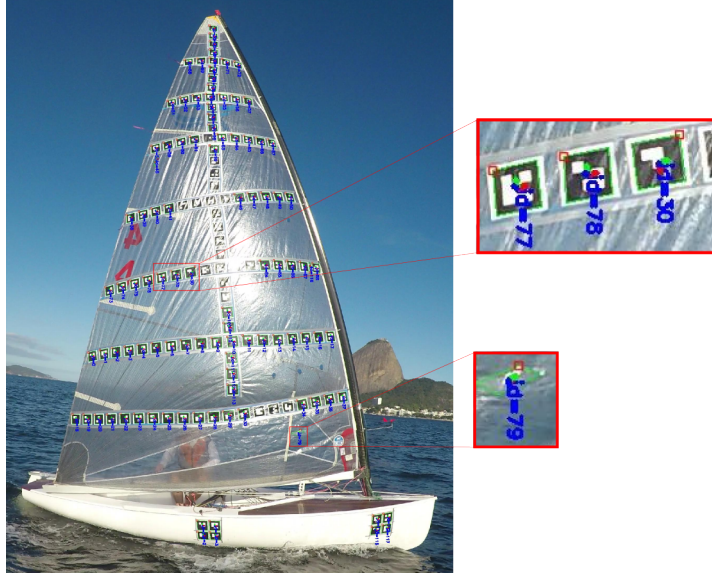


Figure 3.5: Example of false positive markers: marker of index 79 was detected as an artifact on the bottom of the sail and the correct marker 79 was mislabeled with index 30.

$\{M_{k,f}^i \mid M_{k,f}^i \text{ is a candidate for marker } k \text{ at frame } f\}$ . Initially, all markers with only one candidate, that is  $|C_{k,f}| = 1$ , are marked as correct. Given a marker index  $k$ , such that  $|C_{k,f}| > 1$ , for each candidate  $M_{k,f}^i \in C_{k,f}$ , we compute the average distance in pixels (px) between its marker center  $\mathbf{x}_{k,0}^i$  and the three topologically nearest vertices that are already marked as correct. If a marker is an outlier, we expect it to be far from its topological neighbors. For example, in Figure 3.5, marker 30 is far from markers 28, 29, 31 and 32. Thus, the candidate with smallest average distance is selected as the marker with index  $k$ , and all other candidates are discarded. Even though this criterion is not fail proof, it works well because duplicate markers are rare in practice. After this initial selection, we have only a single candidate for each marker. Algorithm 1 shows the pseudocode of our duplicate removal algorithm. We further implemented another topological verification for the non-duplicate candidates to verify that they are really correct. However, we noted that this verification did not improve the reconstruction results. The two additional filtering steps applied during registration (Section 3.4.1) and reconstruction (Section 3.5) are more effective to remove outliers. Thus, we have chosen to handle only the duplicate markers in the detection step.

Thus, for each frame  $f$ , we define the set:  $D_f = \{M_{k,f}\}$  of markers detected and verified at frame  $f$ . Henceforth, when a marker of index  $k$  is discarded at a frame  $f$ ,  $M_{k,f}$  will be removed from  $D_f$ .

**Input** : A set of candidates markers  $C_{k,f} = \{M_{k,f}^i\}$  for each marker index  $k$  detected at frame  $f$

**Output**: A single candidate  $M_{k,f}$  selected for each  $k$  and frame  $f$

```

begin
  for each frame  $f$  do
    for each marker index  $k$  do
      if  $|C_{k,f}| = 1$  then
         $M_{k,f}^0$  is marked as correct and selected as the marker with
        index  $k$ 
      end
    end
    for each marker index  $k$  do
      if  $|C_{k,f}| > 1$  then
        for each candidate  $M_{k,f}^i$  do
          compute the average distance in pixels (px) between its
          marker center  $\mathbf{x}_{k,0}^i$  and the three topologically nearest
          vertices that are already marked as correct
        end
         $M_{k,f}^i$  with smallest average distance is selected as the marker
        with index  $k$ 
      end
    end
  end
end

```

**Algorithm 1:** Duplicate removal algorithm. Markers that were detected more than once are checked against its closest neighbors on the graph to remove the incorrect duplicates.

### 3.4 Registration

The markers on the sail and the camera move independently over time. Their relative position changes constantly during recording, as illustrated by Figure 3.6a. For each video frame  $f$  we initially have a different coordinate system, therefore, we need to define a common reference system for all frames, as illustrated in Figure 3.6b.

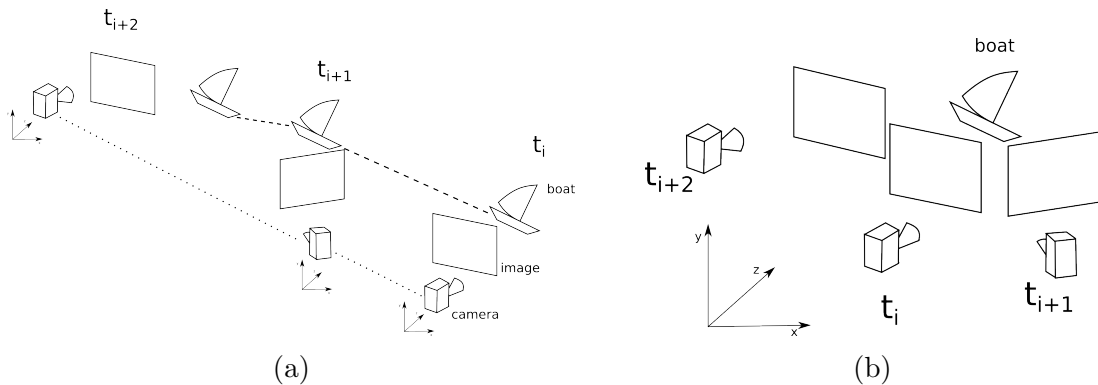


Figure 3.6: Camera and sail dynamics during different time frames. (a) Before registration. (b) After registration with a global reference frame.

To perform the reconstruction, we define a central frame  $r$ , around which we intend to achieve the average sail configuration. Next, we select  $n$  frames before and  $n$  frames after  $r$ . These  $n$  frames do not need to be selected consecutively, as frames with small time differences are very similar and do not add much new information to the reconstruction. In fact, very similar frames may even cause numerical issues for the reconstruction. The spacing between frames depends on recording conditions such as boat velocity and video frame rate, and the criterion to select the frames will be detailed below. For now, without loss of generality, let's define the set that contains the selected  $2n + 1$  frames as:

$$S = \{f \mid \text{frame } f \text{ was selected to compose the reconstruction}\}.$$

For each frame  $f \in S$ , given its verified markers  $M_{k,f} \in D_f$ , we need to find the rigid transformation that optimally aligns all the markers centers  $\mathbf{p}_{k,0} \in \mathbf{P}_{k,f}$  denoted by  $\mathbf{p}_{k,0}^{(f)}$  and  $\mathbf{p}_{k,0} \in \mathbf{P}_{k,r}$  denoted by  $\mathbf{p}_{k,0}^{(r)}$ :

$$(\mathbf{R}_{f,r}, \vec{\mathbf{v}}_{f,r}) = \arg \min_{\mathbf{R} \in SO(3), \vec{\mathbf{v}} \in \mathbb{R}^3} \sum_{k \in D_f \text{ and } k \in D_r} \|(\mathbf{R} \cdot \mathbf{p}_{k,0}^{(f)} + \vec{\mathbf{v}}) - \mathbf{p}_{k,0}^{(r)}\|^2, \quad (3.1)$$

where  $k \in D_f$  and  $k \in D_r$ ,  $\mathbf{R}_{f,r} \in SO(3)$  and  $\vec{\mathbf{v}}_{f,r} \in \mathbb{R}^3$  are the rotation and the translation that align  $f$ 's reference system with  $r$ 's. Equation 3.1 is a least square problem which can be solved by Singular Value Decomposition (SVD) [43]. It must be solved for each  $f \in S$ , resulting in  $|S| - 1 = 2n$  rigid transformations.

### 3.4.1 Filtering markers with RANSAC

Some markers can be erroneously estimated by ArUco at frame  $f \in S$ . These wrong markers are not related to central frame  $r$  by the same transformation as the correct markers. Since the least square solution of Equation 3.1 searches for a solution that best fits all markers, these outliers disturb the solution  $(\mathbf{R}_{f,r}, \vec{\mathbf{v}}_{f,r})$ . It is important to filter these wrong markers to maximize the registration quality. For this purpose, we employ a Random Sample Consensus (RANSAC) scheme to select the best points to perform the registration. Markers that are identified as outliers by RANSAC are removed from  $D_f$ , resulting in a filtered version of  $D_f$ , which is used to solve Equation 3.1 and find  $(\mathbf{R}_{f,r}, \vec{\mathbf{v}}_{f,r})$ .

This RANSAC strategy is also used to select the  $n$  frames before and after frame  $r$ . Starting from frame  $r$ , we skip  $s$  frames backwards to frame  $c_0 = r - s$ . We then apply RANSAC between  $r$  and each frame between  $c_0 - m$  and  $c_0 + m$ . The frame  $f \in [c_0 - m, c_0 + m]$  with the largest number of inliers is selected. Next, we start from frame  $f$  and skip  $s$  frames backward defining a new frame  $c_1 = f - s$  and

repeat the process around the  $c_1$  neighborhood. This search is repeated until we select  $n$  frames before, and, likewise,  $n$  frames after  $r$ . It is important to note that the parameters  $n$  (number of selected frames),  $s$  (skip size) and  $m$  (neighborhood size) need to be carefully chosen, and will be discussed in Section 4.3.3.

Finally, for each  $\mathbf{P}_{k,f}$  such as  $f \in S$  and  $k \in D_f$ , we apply the estimated rigid transformation:

$$\mathbf{P}'_{k,f} = \mathbf{R}_{f,r} \cdot \mathbf{P}_{k,f} + \vec{\mathbf{v}}_{f,r} \quad (3.2)$$

where  $\mathbf{R}_{f,r}$  and  $\vec{\mathbf{v}}_{f,r}$  are the rotation and translation between  $f$  and  $r$ , using  $D_f$  with hindering markers removed. Points from  $\mathbf{P}'_{k,f}$  are in the same reference system as the central frame  $r$ . Notice that the markers image points are not modified by the registration, since we transform only the points in camera space.

One pertinent observation is that any marker detected in a frame  $f \in S$  and not detected in frame  $r$  is not handled by RANSAC, and thus may not be classified as an outlier. These markers do not participate in the computation of  $(\mathbf{R}_{f,r}, \vec{\mathbf{v}}_{f,r})$ , but we opted to register them using Equation 3.2, and evaluated them by the weighted average described in Section 3.5 instead of RANSAC. Hence, we avoid discarding a marker that is not detected in central frame  $r$ , but is correctly detected in other frames  $f \in S$ .

It is important to note that we use the 3D marker information  $\mathbf{P}_{k,f}$  provided by the ArUco. We also implemented a triangulation method to estimate the 3D points  $\mathbf{P}_{k,f}$  from the image points  $\mathbf{X}_{k,f}$  of two images. However, a triangulation algorithm presents very numerical instability, depending on a good parallax between the frames. Thus, the results achieved with this approach were not satisfactory.

## 3.5 Reconstruction

Let:

$$Q_k = \{f \mid M_{k,f} \in D_f \text{ and } f \in S\}$$

be the set of selected frames where marker of index  $k$  was correctly detected. A marker needs to appear in a minimum number of frames so that its position can be correctly optimized by the Bundle Adjustment (BA) algorithm [39]. To avoid optimization problems, if  $|Q_k| < \beta$ ,  $M_{k,f}$  is removed from  $D_f$ , for all  $f \in S$ . The threshold  $\beta$  is our frequency tolerance and its value will be discussed in Chapter 4. Thus, only markers of index  $k$  such that  $|Q_k| \geq \beta$  will be reconstructed. The set of these marker indexes to be reconstructed is then defined as:

$$I = \{k \mid |Q_k| \geq \beta\}.$$



After the frequency tolerance removal, the updated set  $D_f$  can be used to estimate a matrix  $\bar{\mathbf{P}}_k$  of the mean markers positions for markers indexes  $k \in I$ . We initialize  $\bar{\mathbf{P}}_k$  as:

$$\bar{\mathbf{P}}_k^0 = \frac{1}{|Q_k|} \sum_{f \in Q_k} \mathbf{P}'_{k,f},$$

where  $k \in I$ .

After computing this initial mean  $\bar{\mathbf{P}}_k^0$ , we start an iterative algorithm to compute a weighted averaged [43] position for each marker  $k \in I$ . For each iteration  $i$ ,  $\bar{\mathbf{P}}_k^i$  is given by:

$$\bar{\mathbf{P}}_k^i = \left( \sum_{f \in Q_k} \mathbf{W}_{k,f}^i \right)^{-1} \cdot \sum_{f \in Q_k} \mathbf{W}_{k,f}^i \cdot \mathbf{P}'_{k,f},$$

where  $\mathbf{W}_{k,f}^i$  is a weight matrix defined as:

$$\mathbf{W}_{k,f}^i = \begin{bmatrix} w_{k,0}^i & 0 & 0 & 0 & 0 \\ 0 & w_{k,1}^i & 0 & 0 & 0 \\ 0 & 0 & w_{k,2}^i & 0 & 0 \\ 0 & 0 & 0 & w_{k,3}^i & 0 \\ 0 & 0 & 0 & 0 & w_{k,4}^i \end{bmatrix}$$

where  $w_{k,j}^i = e^{-\frac{\|\mathbf{P}'_{k,j} - \bar{\mathbf{P}}_{k,j}^{i-1}\|}{\sigma}}$ , for  $j = 0, 1, 2, 3, 4$ . Thus,  $\mathbf{W}_{k,f}^i$  is a gaussian weight matrix that favors points nearer to the average in the previous iteration. Points far from the average will have a decreasing weight and the process converges after few iterations [43]. At the end of this iterative process, we have a matrix:

$$\bar{\mathbf{P}}_k = \begin{bmatrix} \bar{\mathbf{p}}_{k,0} & \bar{\mathbf{p}}_{k,1} & \bar{\mathbf{p}}_{k,2} & \bar{\mathbf{p}}_{k,3} & \bar{\mathbf{p}}_{k,4} \end{bmatrix}^T$$

of the mean position of the markers points for each  $k \in I$ . This weighted iterative estimation converges to a fair estimative by progressively penalizing points far from the mean. Thus, we can define a set:

$$\bar{P} = \{\bar{\mathbf{p}}_{k,j} \mid \bar{\mathbf{p}}_{k,j} \in \bar{\mathbf{P}}_k, j = 0, 1, 2, 3, 4, k \in I\} \quad (3.3)$$

of the mean marker points positions. This estimative of mean points will be refined by the BA algorithm.

### 3.5.1 Bundle adjustment

We further refine the mean points estimative using Bundle Adjustment (BA) [39]. It optimizes the points reconstructed in world space and the cameras poses by minimiz-

ing the points projection error in image space. BA is important to globally optimize our reconstructed points taking into account all the selected frames. Note that up to this point, we were only computing transformation between pair of frames, but to have a globally consistent set of frames, it is important to optimize the points and cameras simultaneously.

The algorithm needs three inputs: a set  $W = \{\mathbf{w}_i \in \mathbb{R}^3\}$  of points in world space, a set  $C = \{(\mathbf{R}_j, \vec{\mathbf{t}}_j) \mid \mathbf{R}_j \in SO(3) \text{ and } \vec{\mathbf{t}}_j \in \mathbb{R}^3\}$  of camera poses, and a set  $Y = \{\mathbf{y}_{ij} \in \Omega \mid \mathbf{y}_{ij} \text{ is the image of point } \mathbf{w}_i \text{ by camera } j\}$ . For our problem, we define:

$$W = \bar{P}, \quad (3.4)$$

where  $\bar{P}$  is the set of mean points defined in Equation 3.3. In this case, the coordinate system of points in  $\bar{P}$  is the world coordinate system.

For each  $f \in S$ , we need to find an initial estimative for camera pose  $(\mathbf{R}_f, \vec{\mathbf{t}}_f)$  in relation to the world system. This estimative pose can be achieved by finding the rigid transformation that optimally aligns all the markers position centers between frame  $f$  and the world points, similar to the problem of Equation 3.1:

$$(\mathbf{R}_f, \vec{\mathbf{t}}_f) = \arg \min_{\mathbf{R} \in SO(3), \vec{\mathbf{t}} \in \mathbb{R}^3} \sum_{k \in D_f} \|(\mathbf{R} \cdot \bar{\mathbf{p}}_{k,0} + \vec{\mathbf{t}}) - \mathbf{p}_{k,0}\|^2, \quad (3.5)$$

where  $\mathbf{p}_{k,0} \in P_{k,f}$  is the center position of marker  $M_{k,f}$ ,  $k \in I$  before registration and  $\bar{\mathbf{p}}_{k,0}$  is the world position of this marker center. Solving Equation 3.5 for each  $f \in S$  we find our camera poses set  $C$ :

$$C = \{(\mathbf{R}_f, \vec{\mathbf{t}}_f) \mid f \in S\}, \quad (3.6)$$

where  $(\mathbf{R}_f, \vec{\mathbf{t}}_f)$  is the pose of the camera that captured the frame  $f \in S$  in world space.

Our image points set  $Y$  is defined as:

$$Y = \{\mathbf{x}_{k,j} \mid \mathbf{x}_{k,j} \in \mathbf{X}_{k,f}, j = 0, 1, 2, 3, 4, f \in S, k \in I \text{ and } k \in D_f\}. \quad (3.7)$$

Thus, we apply the BA algorithm implemented by g2o library [44] using our sets  $W$ ,  $C$  and  $Y$  defined by Equations 3.4, 3.6 and 3.7 as input. The algorithm returns the optimized sail points  $W^*$  and camera poses  $C^*$ . Although BA may not maintain the real points scale, this issue can be corrected since we know the real size of the markers. We first compute the average marker side length  $\bar{l}$  from the points in  $W^*$ , and then scale each  $\mathbf{w}_i \in W^*$  and  $\vec{\mathbf{t}}_j \in C^*$  by  $l/\bar{l}$ , where  $l$  is the real marker side length. This scaled version of  $W^*$  is our average sail configuration around the central frame  $r$ .

# Chapter 4

## Experiments

As presented in Chapter 3, we extract information from augmented reality markers placed on the sail surface at known initial positions. Augmented reality markers are squares with a black border and an internal pattern. This pattern differentiates the markers from each other. KATO and BILLINGHURST [40] proposed an algorithm to detect these markers and extract their 3D position and orientation in camera coordinate system. This algorithm was implemented in the ARToolKit library [45] and is widely used for augmented reality applications. More recently, GARRIDO-JURADO *et al.* [41] presented a new augmented reality library based on OpenCV named ArUco. They use a detection method similar to ARToolKit and allow creating custom dictionary of markers. We performed tests using the ARToolKit and ArUco libraries in order to evaluate the accuracy of the position and orientation detected, and compare the performance of them.

### 4.1 Marker detection tests

In order to evaluate the accuracy of the libraries detection, we built a planar pattern and analyzed the errors of the detected markers. We produced a pattern composed by 12 markers of 40 mm of side length and spaced by known distances according to Figure 4.1. This pattern was fixed on a planar glass and used to perform the tests.

We evaluated the precision of the position and the orientation (tangents and normal) of the markers provided by ARToolKit and ArUco. The videos were captured by a Logitech webcam C920 with resolution of 720p. We recorded videos of the pattern in eight different situations. These situations vary according to the pattern orientation and the rotation around the axes defined in the Figure 4.2:

- a. the pattern is in a horizontal orientation and almost static;
- b. the pattern is in a horizontal orientation and rotates approximately around the y axis;

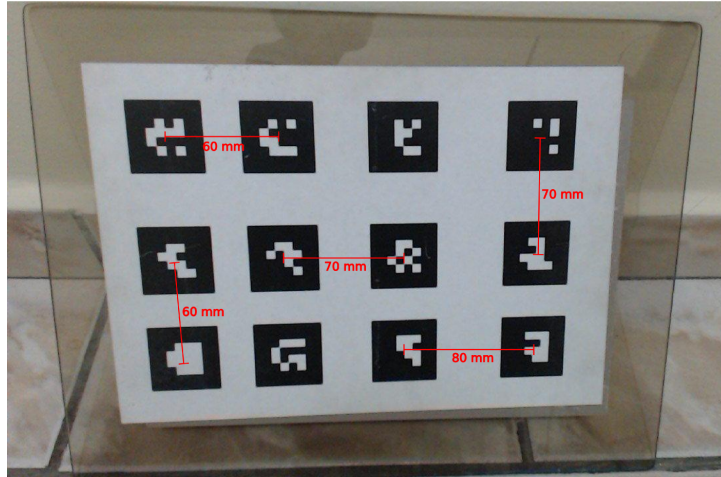


Figure 4.1: Planar pattern built for accuracy tests of the ARToolkit.

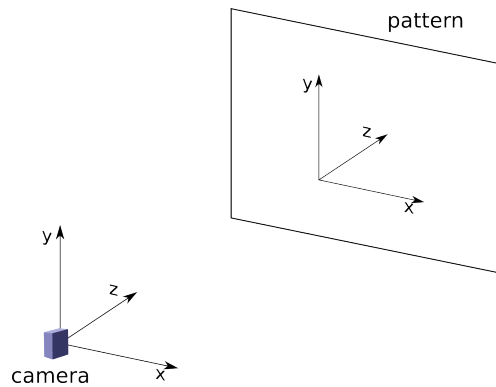


Figure 4.2: Axes defined for pattern rotation.

- c. the pattern is in a horizontal orientation and rotates approximately around the x axis;
- d. the pattern is in a horizontal orientation and rotates randomly approximately around the pattern center;
- e. the pattern is in a vertical orientation and almost static;
- f. the pattern is in a vertical orientation and rotates approximately around the y axis;
- g. the pattern is in a vertical orientation and rotates approximately around the x axis;
- h. the pattern is in a vertical orientation and rotates randomly approximately around the pattern center.

Figure 4.3 shows images of these situations.

In order to measure the accuracy of the position we calculated the Euclidian distance between the positions provided by the library for each pair of adjacent

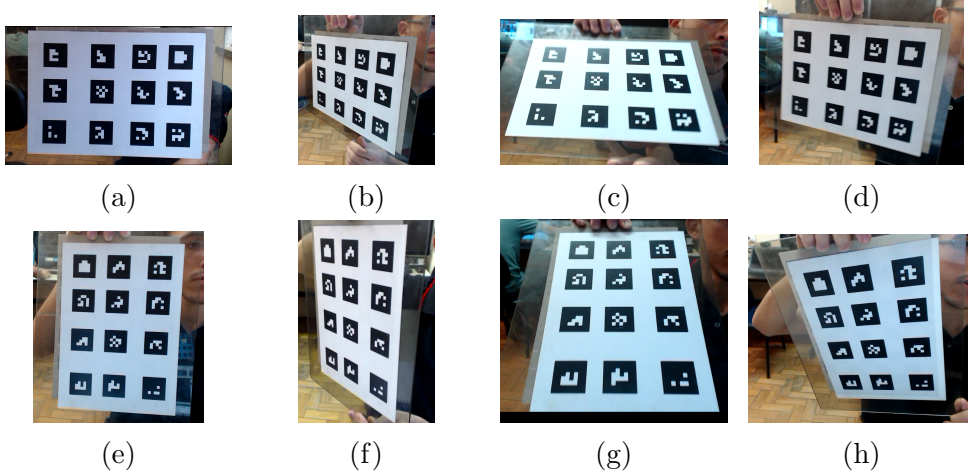


Figure 4.3: Test situations. (a) the pattern is in a horizontal orientation and almost static. (b) the pattern is in a horizontal orientation and rotates approximately around the y axis. (c) the pattern is in a horizontal orientation and rotates approximately around the x axis. (d) the pattern is in a horizontal orientation and rotates randomly approximately around the pattern center. (e) the pattern is in a vertical orientation and almost static. (f) the pattern is in a vertical orientation and rotates approximately around the y axis. (g) the pattern is in a vertical orientation and rotates approximately around the x axis. (h) the pattern is in a vertical orientation and rotates randomly approximately around the pattern center.

markers and compared to the real distances. The distance error  $e_d$  was computed as the absolute value of difference between the calculated distance  $d_{calc}$  and the real distance  $d_{real}$ :

$$e_d = |d_{calc} - d_{real}|.$$

Since this first test was performed with a planar pattern, we expected that the detected markers would be on a same plane. Thus, we performed a plane regression with the markers positions and computed the distances between each marker and the resulting plane. This distance is the plane projected error of each marker.

In addition to the position of the markers, the detection algorithms also provides their orientation, i.e., the normal and tangents of the marker. Since our pattern is planar, all markers should have the same orientation. In order to evaluate the extracted orientation, we computed the normalized mean vector  $\vec{v}_{mean}$  for each orientation vector for every frame:

$$\vec{v}_{mean} = \frac{\sum \vec{v}_i}{\|\sum \vec{v}_i\|},$$

where  $\vec{v}_i$  is the orientation vector for each marker. We calculated this normalized mean vector for the normal and the x-direction tangent. We evaluated the orientation of each marker by calculating the angle between the marker's detected orientation and the normalized mean vector.

For every frame of each video, we computed the distance, plane projected, and orientation errors for the detected markers. To identify a general behavior we compute the average error values for each situation presented in Figure 4.3. Figures 4.4, 4.5, 4.6 and 4.7 show charts comparing the mean errors achieved using the two libraries.

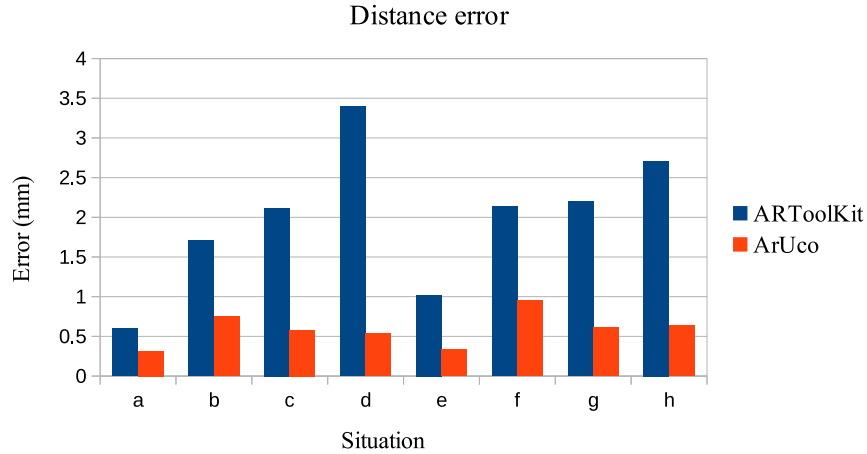


Figure 4.4: Comparison between mean distance errors achieved by ARToolKit and ArUco libraries.

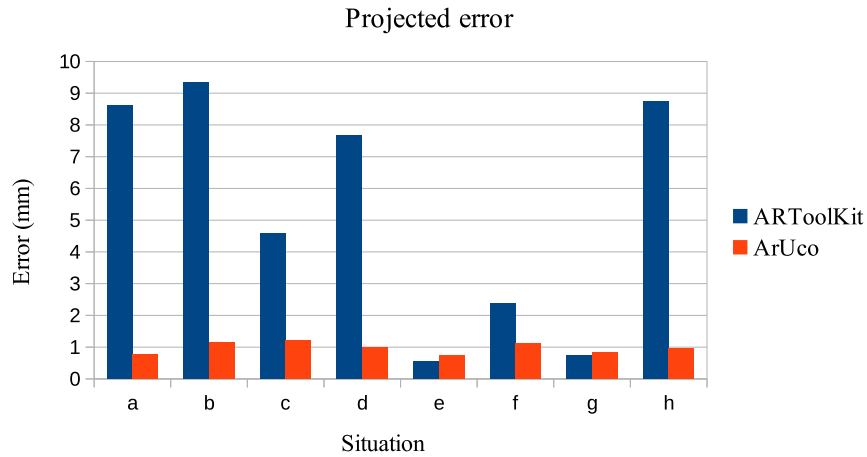


Figure 4.5: Comparison between mean projected error achieved by ARToolKit and ArUco libraries.

The analysis of Figures 4.4, 4.5, 4.6 and 4.7 shows that ArUco presents a performance significantly better than ARToolKit for most computed errors. We observe that the errors are stable for all the situations, including the most challenging, for the ArUco library, while for the ARToolKit the performance is significantly worst. Moreover, we observe that the mean errors are around one millimeter. Although these initial tests were performed in a controlled laboratory scenario, using an ideal

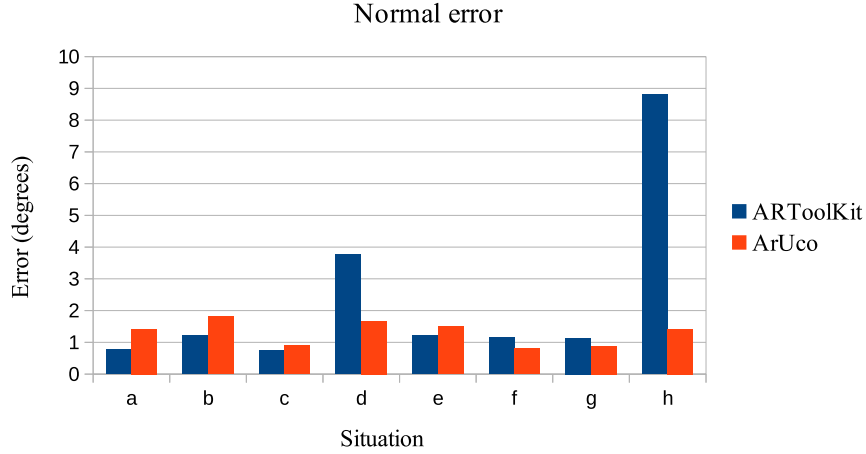


Figure 4.6: Comparison between mean normal error achieved by ARToolKit and ArUco libraries.

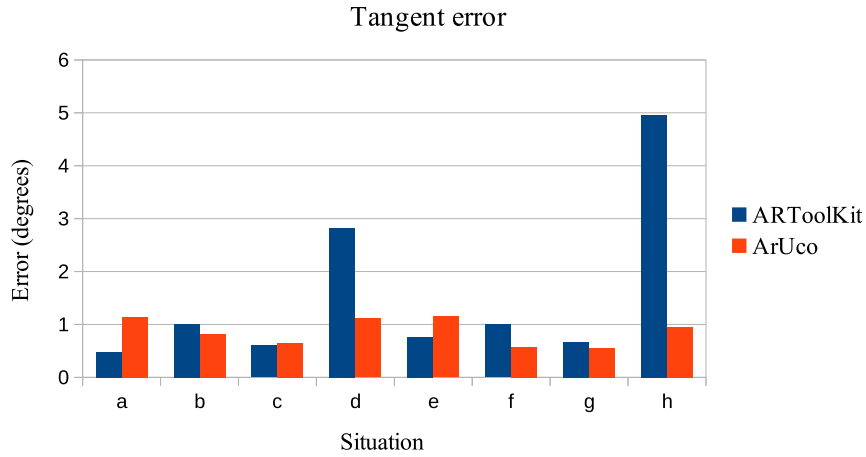


Figure 4.7: Comparison between mean tangent error achieved by ARToolKit and ArUco libraries.

object, and recorded at a short distance, we compensated some hardships concerning our outdoor environment with a camera with better resolution and stability. Thus, based on the presented tests, we opted to use ArUco markers in our proposed method.

The ArUco library allows creating markers dictionary. Besides the number of different dictionary markers, we can define the number of internal and border bits. In order to find the ideal marker parameters for our application, we perform some detection tests varying these parameters. The tests were performed using a Go Pro Hero 5 Black camera, the same used for the real experiments. Figure 4.8 shows markers with 9, 16 and 25 bits of internal pattern. The greater the number of internal bits the more differentiable it is. However, markers with less internal bits

are more easily detectable from greater distances because the squares that represent each bit has greater area. We perform some experiments varying the internal bits and the distance from the camera and confirmed that markers with fewer bits are better for our purpose. Thus, we decided to use markers of 9 internal bits.

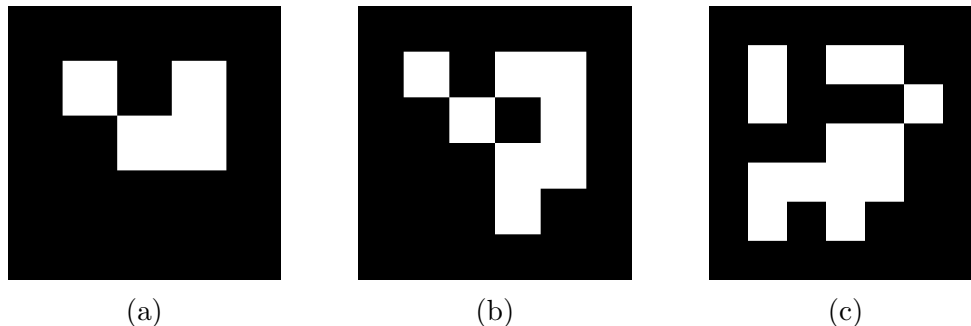


Figure 4.8: Examples of markers with different number of internal bits. (a) 9 bits. (b) 16 bits. (c) 25 bits.

Figure 4.9 presents markers with 1, 2 and 3 border bits. A greater border has a larger black external area, which facilitates the detection of the marker in the image, but makes the area of the internal pattern smaller, hindering the labeling. In particular, for great distances a smaller border is better for detection and labeling, since the smaller the number of bits the larger the area of each square bit. Thus, after some tests, we opted by using markers with 1 bit of border.

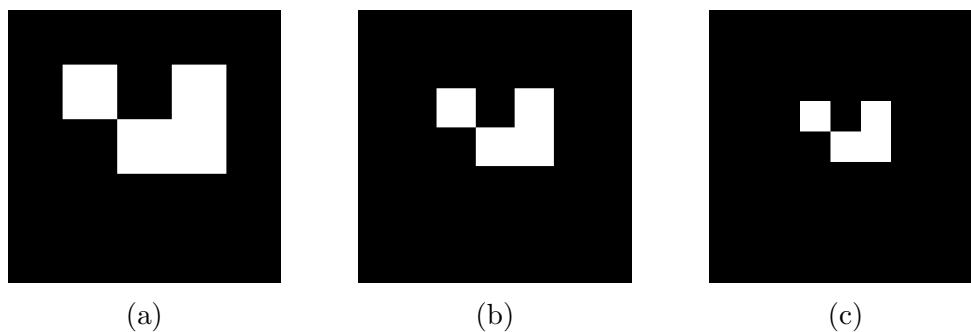


Figure 4.9: Examples of markers with different number of border bits. (a) 1 bit. (b) 2 bits. (c) 3 bits.

Concerning the marker side length, we performed some experiments varying the marker size and the camera distance. We tested markers of side length varying from 6 to 12 cm. We varied the distance from 3 to 7 meters, since we knew that a distance from 3 to 5 meters is a possible range in practice for capturing the sail. We noticed that markers larger than 8 cm of side can be detected from up to 7 meters. Since these tests were performed in a controlled environment, we opted to use 10 cm markers to ensure they would be detected in our real sailing tests.



## 4.2 Rigid object tests

Before our first experiments with a real sail, we performed some tests with rigid objects and ground truths. These experiments allowed to evaluate our proposed method and limit the parameters range. Figure 4.10 present two rigid objects used in these experiments: the walls of a room (Figure 4.10a) and a pattern fixed on a glass (Figure 4.10b).

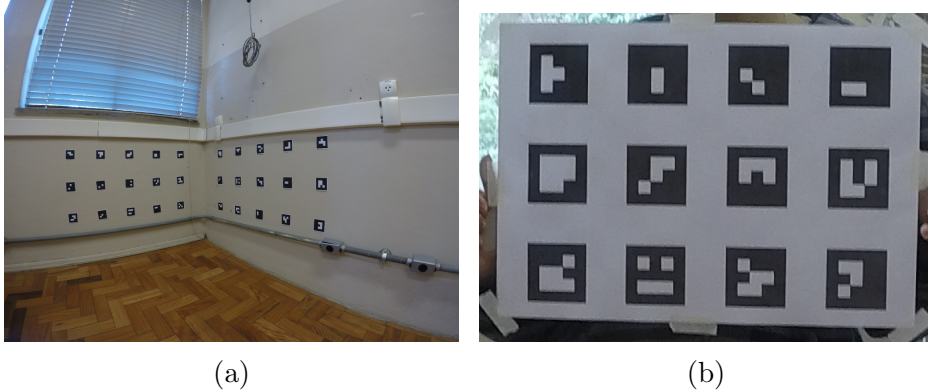


Figure 4.10: Rigid objects used for preliminary tests. (a) Room walls. (b) Pattern fixed on a glass.

For the room (Figure 4.10a) we analyzed the plane projected error described in Section 4.1 for the two walls. We achieved results smaller than 0.5 mm after bundle adjustment. We also computed the angle between the normal of the two planes, which is expected to be  $90^\circ$ . The angle errors were smaller than  $0.5^\circ$  after bundle adjustment.

For the glass (Figure 4.10a), we analyzed the distance between adjacent markers and compared to the real distance. We performed several tests recording the pattern from several distances in range from 3 to 7 meters and, moving the camera, the object or both. For the most challenging case, where both camera and object move, we achieved errors smaller than 0.5 mm for the plane projected error and smaller than 1 mm for the markers distance, after applying the bundle adjustment algorithm.

In our last experiments before real sail tests, we used a plastic deformable surface showed in Figure 4.11. Since the markers were manually fixed and the surface is deformable, we could not establish an accurate ground truth for this object. But we analyzed the reconstructed shape and compare to the expected shape. We observed that the results were consistent with the expected.



Figure 4.11: Deformable plastic used for preliminary tests.

### 4.3 Sail reconstruction tests

In this section we present the experiments performed to evaluate our method. We printed and fixed 122 markers on a Finn Class sail forming 7 horizontal and 1 vertical line. Furthermore, 8 markers were fixed on the boat hull as depicted in Figure 4.12. The sail was captured by a Go Pro Hero 5 Black camera using the following resolutions and frame rates: 4K at 30 FPS, 2.7K at 60 FPS and 12 MP at 2 FPS (time lapse mode). Preliminary experiments showed that 4K resolution at 30 FPS gives the best trade-off between spatial and temporal resolutions. Thus, all results are presented using this configuration.

It is important to note that the GoPro camera presents high lenses distortion. Nevertheless, the camera has fixed and pre-calibrated intrinsic matrix and radial distortion coefficients, hence we can readily rectify the images.

#### 4.3.1 Sail video dataset

The sail videos registered throughout our experimental sessions are available at <http://www.lcg.ufrj.br/sail3D>. In order to evaluate a more controlled environment, we recorded some videos with the sail ashore (Figure 4.12a). This scenario allowed more control over the capture distance and the illumination. We recorded a total of 20 ashore sequences, including all camera resolutions. After this controlled scenario, we captured the sail in a real sailing environment (Figure 4.12b), totalizing 28 sequences. Our original videos were divided in these two categories: ashore and sailing.

The original videos were splitted and classified into two main classes based on the capture distance to the sail: near or far. This division resulted in 39 clips. Each clip presents some particular features, which are presented in Table 4.1. As



(a)



(b)

Figure 4.12: Dataset video frames examples. (a) Ashore video frame. (b) Sailing video frame.

previously mentioned the videos at 4K presented a better performance, only them are discussed.

According to the reflection occurrence the clips can be classified as “Weak reflection” when the reflection obfuscates few markers or “Strong reflection” when many markers are obfuscated by the natural illumination. Figure 4.13 presents examples of these situations. It is important to note that both reflection types can occur in the same clip. For future records, we can soften this issue using filters in the camera.



(a)



(b)

Figure 4.13: Classification of the reflection. (a) Example of weak reflection. (b) Example of strong reflection.

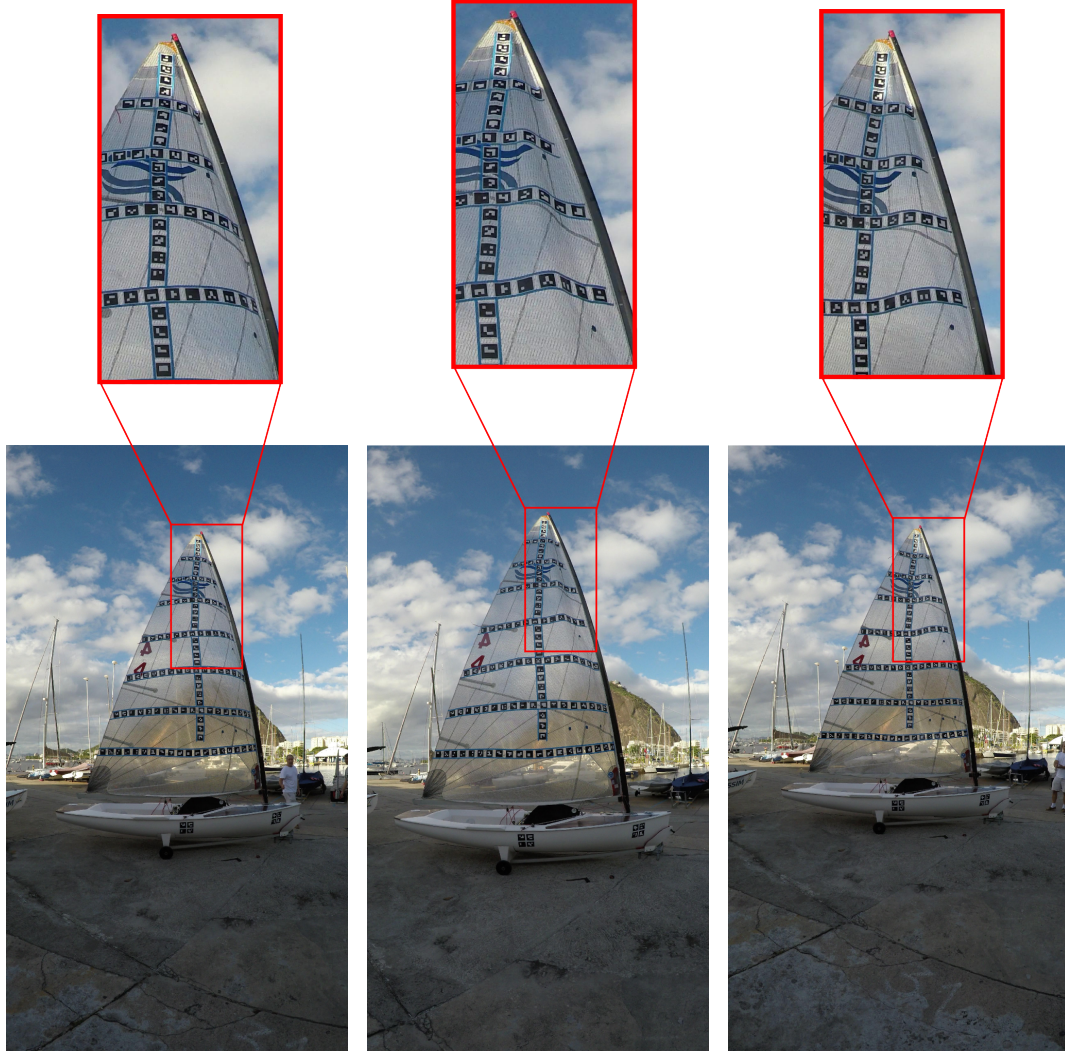
Video	Ashore	Sailing	Weak reflection	Strong reflection	Wind change	Too far	Bad angle	Duration (sec.)
far_4k_01		•						71
far_4k_02		•			•			83
far_4k_03		•	•					27
far_4k_04		•		•			•	18
far_4k_05		•		•				31
far_4k_06		•		•			•	19
far_4k_07		•		•				9
far_4k_08		•		•				74
far_4k_09		•		•				14
far_4k_10		•	•	•		•		85
far_4k_11		•	•			•		16
far_4k_12		•		•		•	•	120
far_4k_13		•		•		•	•	89
far_4k_14		•		•				20
far_4k_15		•		•				20
far_4k_16		•				•		35
far_4k_17		•	•			•		39
far_4k_18		•	•		•	•		35
far_4k_19		•		•		•		48
far_4k_20		•		•			•	42
far_4k_21		•		•			•	13
far_4k_22		•		•			•	51
near_4k_01		•		•			•	18
near_4k_02		•	•					10
near_4k_03		•	•					10
near_4k_04		•		•			•	4
near_4k_05		•		•				8
near_4k_06		•		•				7
near_4k_07		•		•			•	17
near_4k_08	•				•			20
near_4k_09	•							37
near_4k_10	•							30
near_4k_11	•							50
near_4k_12	•				•			40
near_4k_13	•							39
near_4k_14	•				•			25
near_4k_15	•							37
near_4k_16	•				•			25
near_4k_17		•	•	•				59

Table 4.1: Sail dataset video features.

In some clips the wind changes during the capture, modifying the sail shape. The clips where this changing occur were classified as “Wind change”. Figure 4.14 illustrates this fact comparing three frames of the clip “near\_4k\_08.mp4”.

Some clips were recorded from a great distance, which makes markers detection very difficult. These clips are classified as “Too far”. Figure 4.15a presents an example frame of this case. Furthermore, some clips were captured from an almost parallel angle in relation to the sail. Ideally the capture should be performed from an as perpendicular angle as possible to the sail. Thus, the clips captured almost in





(a) Frame 60 (2 sec)

(b) Frame 210 (7 sec)

(c) Frame 330 (11 sec)

Figure 4.14: Sail shape variation due wind changing for clip “near\_4k\_08.mp4”.

parallel were classified as “Bad angle” as showed in Figure 4.15b.

### 4.3.2 General parameters evaluation

We used three metrics to quantitatively evaluate our reconstruction:

- **marker area error**  $e_a = |a_r - a|$ : the absolute difference between the area of the reconstructed marker  $a_r$  and the real area  $a$ ;
- **image reprojected error**  $e_r = \|\Pi(\mathbf{P}) - \mathbf{x}\|$ : the absolute distance in image space between the reconstructed point  $P$  reprojected on the central frame and the respective points detected by ArUco library  $\mathbf{x}$ ;
- **reconstruction ratio**  $\frac{n_r}{N}$ : ratio of reconstructed markers  $n_r$  over the total number of markers on the sail  $N$ .

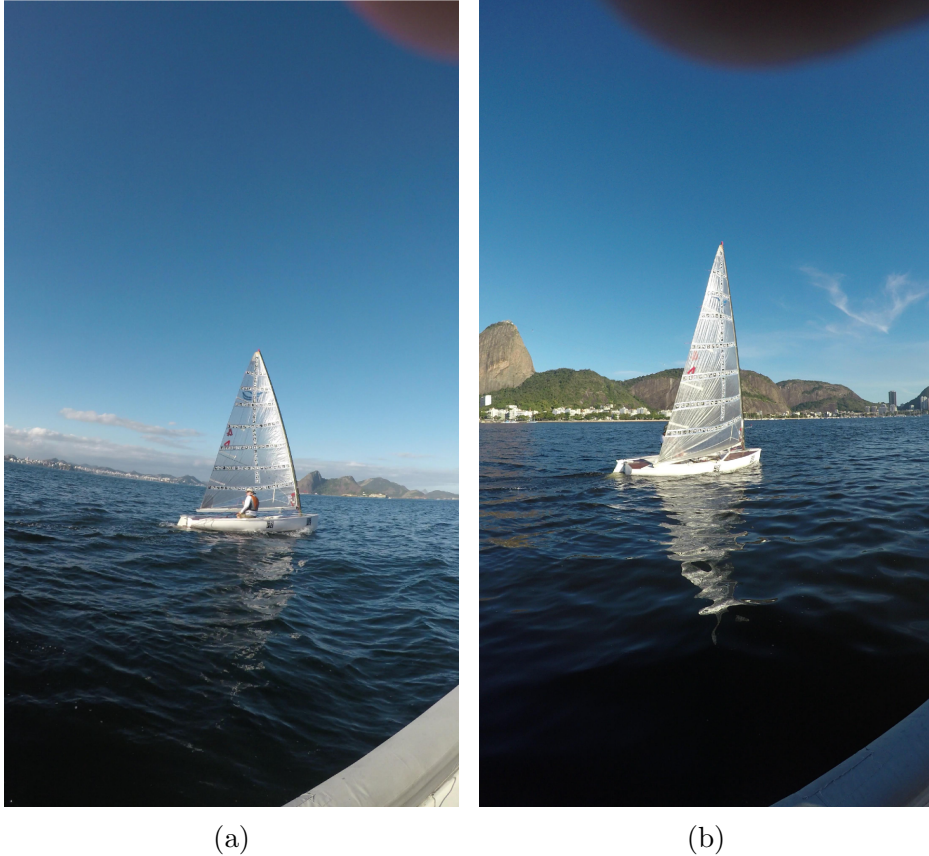


Figure 4.15: Examples of “Too far” and “Bad angle” frames. (a) “Too far” frame. (b) “Bad angle” frame.

For each clip in our dataset, we compute the reconstruction centered in several frames. The area error  $e_a$  was computed for each reconstructed marker and the reprojected error  $e_r$  was computed for each marker point. In order to achieve a general evaluation of the reconstruction and fine tune the parameters, we computed the statistics of the errors: average, standard deviation, median, minimum and maximum.

In Section 3.5, we described our iterative weighted average of the markers points. This average uses a Gaussian weight with parameter  $\sigma$ . We tested some  $\sigma$  values in the interval  $[0.1, 5.0]$  and observed its influence on the metrics. We noted that  $\sigma$  does not influence the reconstruction ratio, and values  $\sigma \geq 0.6$  do not disturb  $e_a$  and  $e_r$ . Thus, we set  $\sigma = 0.6$  for our experiments. We also analyze the threshold  $\beta$  for the frequency filter by varying its value between 10% and 40%. Small values increase the number of reconstructed markers, but also increases  $e_a$  and  $e_r$ . The value  $\beta = 30\%$  presented the best trade-off between reconstruction ratio and errors. We use 30 iterations, which were enough to achieve convergence.

For the RANSAC strategy described in Section 3.4.1, we need to define a threshold for considering a point as an inlier. In our case, this value is the acceptable distance between the registered point and the point in the central frame. We tested

values between 50 and 300 *mm*, and noticed that 100 *mm* presents good results considering  $e_a$  and  $e_r$ . Values below 100 slightly decrease the errors but considerably reduces the number of reconstructed markers. On the other hand, values above 100 increase the reconstruction ratio at the cost of increasing errors.

### 4.3.3 Reconstruction results

The clip “near\_4k\_17.mp4” is the longest sequence recorded in sailing conditions from a reasonable distance. This clip present a good sail stability, parts with weak and strong reflection. Thus, it is considered ideal for reconstruction and the results presented in this section use this clip. It will be used in the next subsections for comparing the results with difficult clips.

Figure 4.16 presents a visualization of the reconstruction centered in the frame 457 from two views. It shows all points for each reconstructed marker.

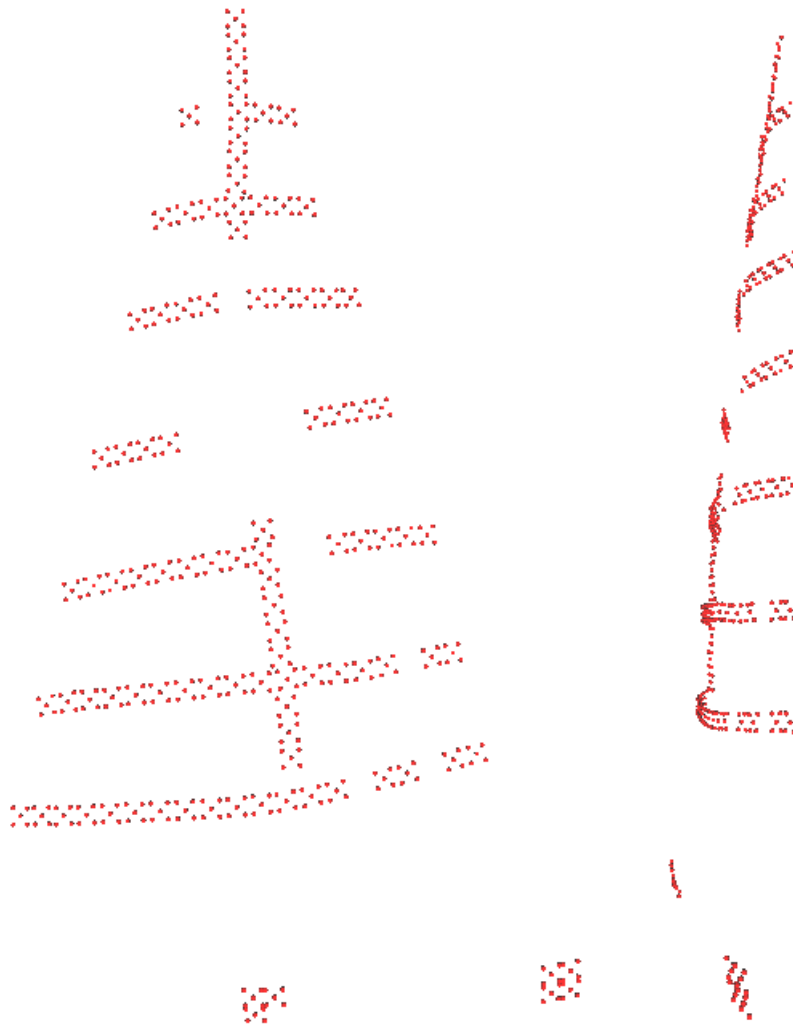


Figure 4.16: Visualization of the reconstruction of the clip “near\_4k\_17.mp4” centered in the frame 457 from two views.

As previously mentioned, we compute the reconstruction centered in several frames and the reconstruction metrics. In order to statistically evaluate the behavior for the entire clip, the metrics statistics presented are computed as follows:

- For a reconstruction performed centered in a frame:
  - Compute the  $e_a$  for each marker, the  $e_r$  for each marker point and the  $\frac{n_r}{N}$  for the reconstruction
  - Compute the average, standard deviation, median, minimum and maximum over all  $e_a$  and  $e_r$  for the reconstruction
- Compute the mean of the statistics over all reconstructions
- Compute the mean of  $\frac{n_r}{N}$  over all reconstructions of the clip

Our frame selection procedure described in Section 3.4.1 depends on three parameters:  $n$  (number of selected frames),  $s$  (skip size) and  $m$  (neighborhood size). We varied  $n$  in the interval  $[5, 50]$ , which results in varying  $|S| = 2n + 1$  in the interval  $[11, 101]$ . Figures 4.17, 4.18 and 4.19 show the metrics. Figure 4.17 presented the mean statistics in function of  $|S|$ . Notice in Figure 4.17 that the error decreases with the increase of selected frames, but after 41 frames the variation is small. The average error was around  $250 \text{ mm}^2$ , which represents 2.5% of the marker area and the maximum around  $1000 \text{ mm}^2$ , which is 10% of the area. For the reprojected error, we observe that the error slightly increase with the frames increase. This increase is expected since we have more frames to be adjusted by the bundle adjustment. Despite this increase, the maximum error does not change, keeping around 2 px. The reconstructed ratio decreases with the total of selected frames, varying from 56.9% for 11 frames to 40.1% for 101 frames, i.e, the more frames used for the reconstruction, the smaller the reconstruction rate. The decrease is more accentuated after  $|S| = 31$ . Thus, we can summarize the analysis of the Figures 4.17, 4.18 and 4.19 as:

- More frames decrease the area error, presenting a stable behavior after  $|S| = 41$ ;
- More frames slightly increase the reprojected error;
- More frames decrease the reconstruction rate, mainly after  $|S| = 31$ .

Based on this analysis we opted to use  $n = 20$ , i.e., selecting  $|S| = 41$  frames for reconstruction. This value ensures a good area error without penalizes too much the reconstruction ratio.



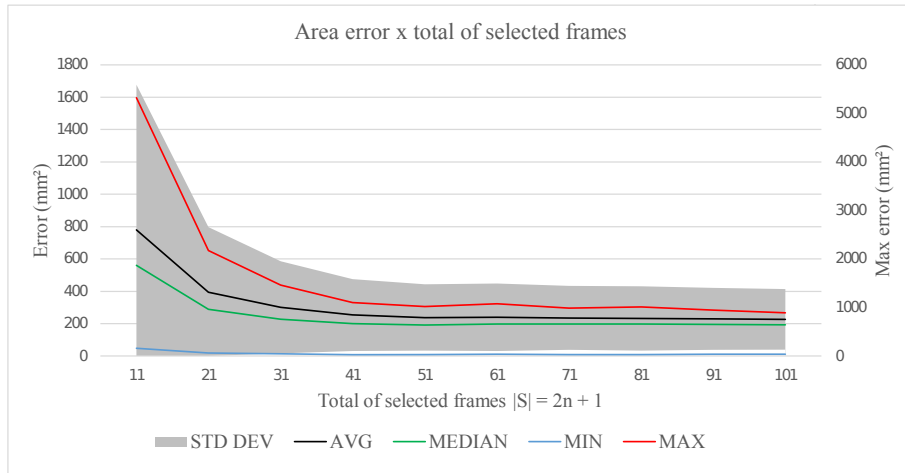


Figure 4.17: Area error in function of total of selected frames. The max error scale is indicated by the right vertical axis.

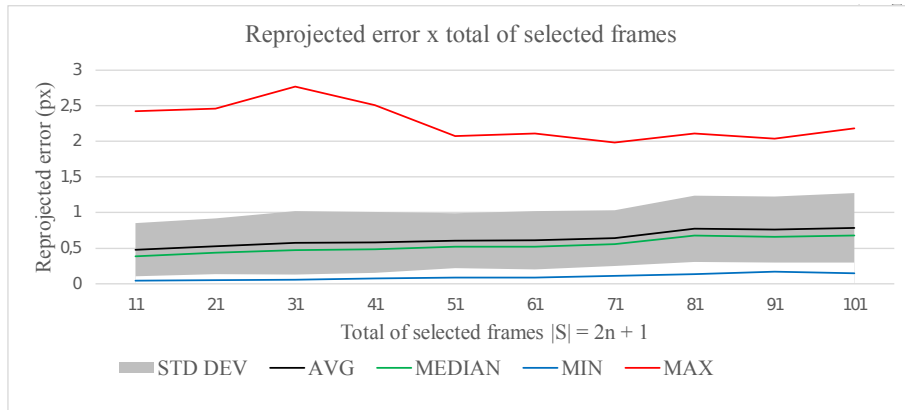


Figure 4.18: Reprojected error in function of total of selected frames.

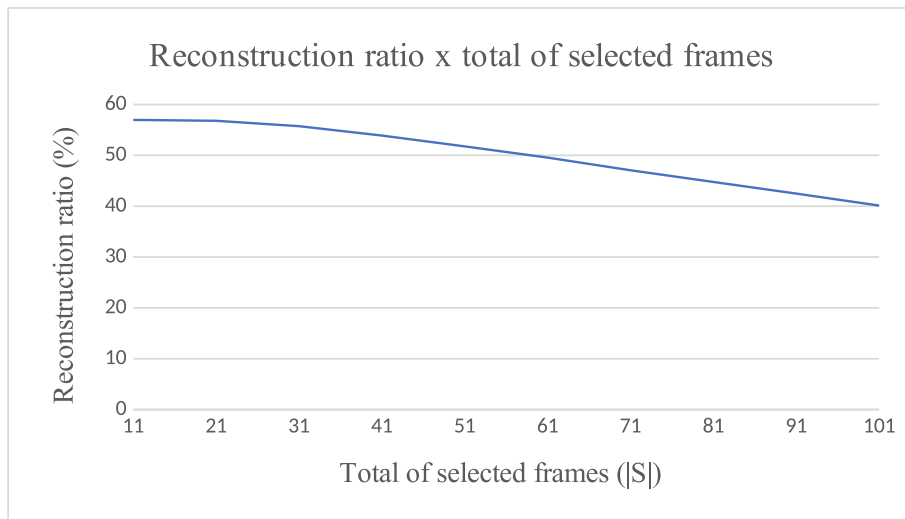


Figure 4.19: Reconstruction ratio in function of total of selected frames.

Figures 4.20, 4.21 and 4.22 present the metrics results in function of the skip size  $s$ . By analyzing Figure 4.20, we note that small values perform poorly. This perfor-

mance is explained by the frames similarity, since the frame rate is high in relation to the movement variation of the scene. On the other hand, if we increase the value of  $s$ , a more long clip is necessary to select the frames, since the interval between two selected frames will be larger, and keeping the sail stable for an extended period is usually not a trivial task. Furthermore, Figure 4.22 shows that the reconstruction ratio decreases with the  $s$  increase. In relation to the reprojected error (Figure 4.21), the behavior was similar to the Figure 4.18, i.e. the error slightly increase with the  $s$  increase. The explanation is also similar. Since the bundle adjustment should adjust frames with more variability between them, it is expected an increase of the mean error to adjust all frames. We observed that  $s = 10$  is a good choice for videos recorded at 30 FPS.

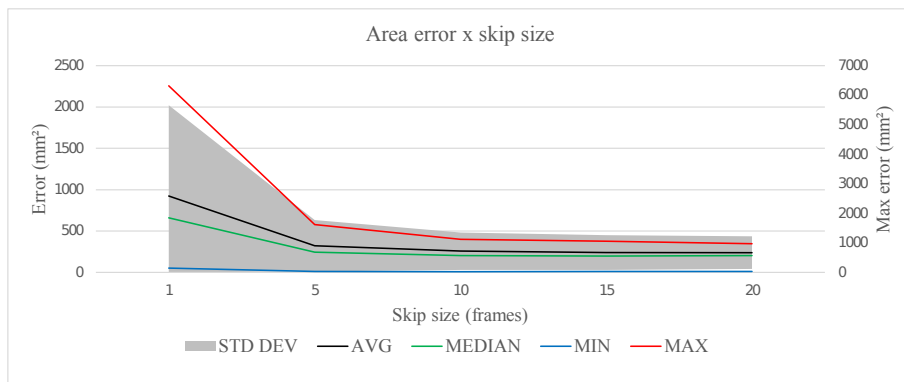


Figure 4.20: Area error in function of skip size. The max error scale is indicated by the right vertical axis.

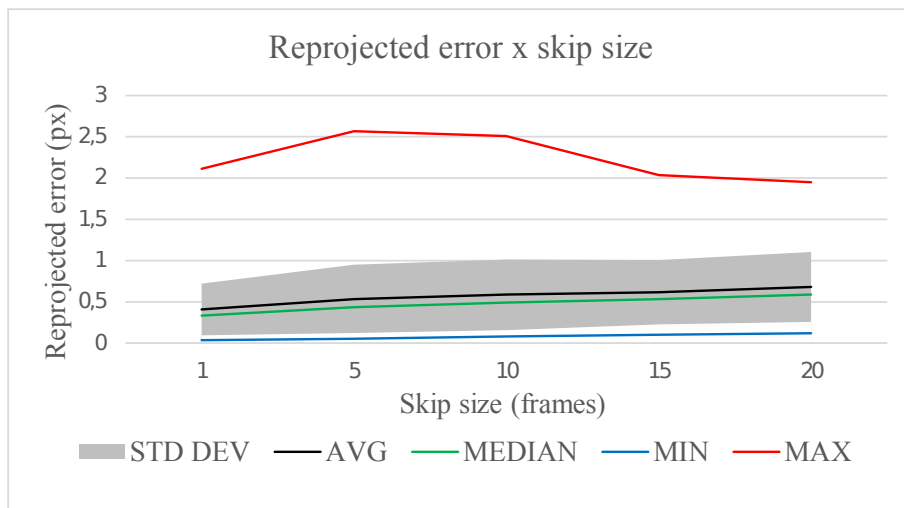


Figure 4.21: Reprojected error in function of skip size.

We also observed that  $m$  has no significant impact on the errors, but increasing  $m$  also increases the reconstruction ratio. This increase occur because we search by frames with the largest number of inliers to register with the central frame. The

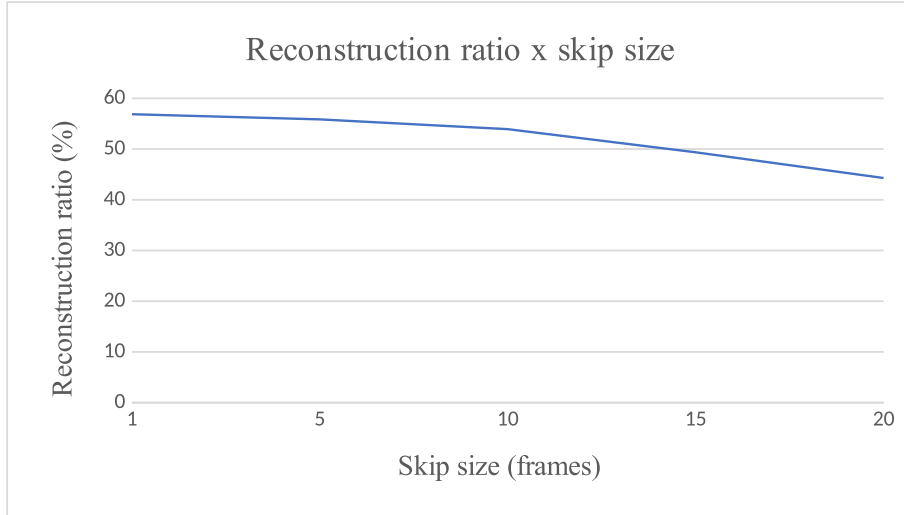


Figure 4.22: Reconstruction ratio in function of skip size.

value of  $m$  should not be greater than  $s$  to avoid overlapping the intervals. We found that  $m = 5$  is a good choice for  $s = 10$ . Considering the values of  $n = 20$ ,  $s = 10$  and  $m = 5$ , we can estimate a minimum video length. In the worst case for these values, all frames are selected with spacing of 15 frames. To select 41 frames ( $n = 20$ ) at least 20 seconds of video at 30 FPS is necessary. However, larger videos allow us to also vary the central frame.

In order to analyze the markers area, Figure 4.23 presents the histogram of the area distribution for the reconstruction using  $n = 20$ ,  $s = 10$  and  $m = 5$ . This histogram considers the area of the markers reconstructed for all reconstructed frames. We observe the most markers presented area near to the expected value.

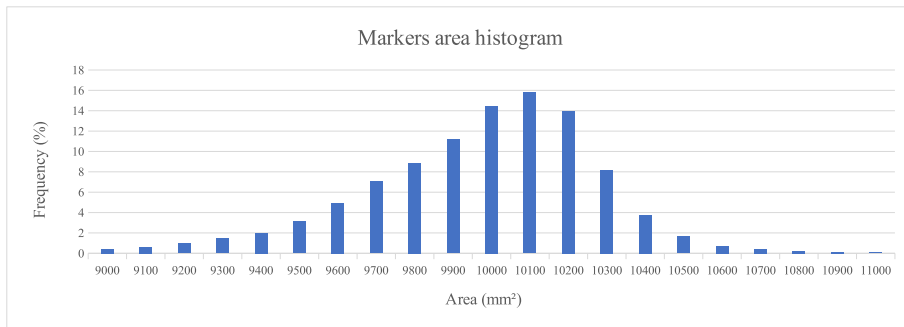


Figure 4.23: Histogram of the markers area distribution.

We notice a considerable difference between the results in the real sailing environment and the controlled tests in laboratory. For our controlled tests we use rigid objects, whose geometry and markers position are accurately known. On the other hand, in our real tests, the only information we have is the marker's length and area. Furthermore the sail surface is deformable, which changes the markers geometry, since they are not exactly plane.

Figures 4.19 and 4.20 present values smaller than 60% for reconstruction ratio. It is important to clarify that the values presented in these figures are the average reconstruction ratio for the all clip frames. Figure 4.24 shows the reconstruction ratio for each frame from 202 to 1502 for the clip using using  $n = 20$ ,  $s = 10$  and  $m = 5$ . We observe that the reconstruction ratio is around 70% before the frame 600, i.e., before 20 seconds of the video. After this frame, the ratio decreases, only rising again near to the clip end. The decreasing can be justified by the increase of the capture distance.

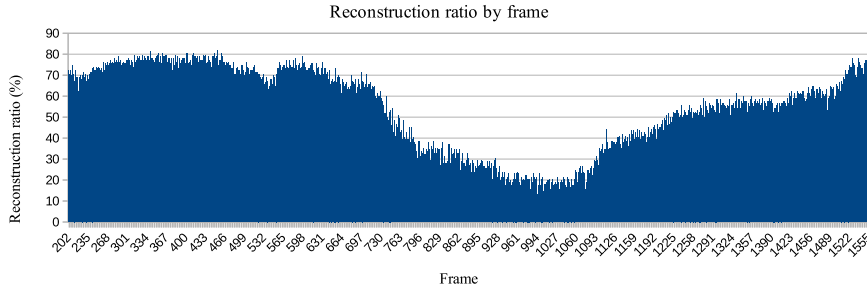


Figure 4.24: Reconstruction ratio by frame for clip “near\_4k.17.mp4”.

Figure 4.25 shows the reprojection of the reconstructed points (Figure 4.16) on the central frame 457. It is possible to note that the points are projected on the expected positions, i.e., at the center of the markers. Observe that this frame presents a weak reflection (highlighted in red in Figure 4.25) and the markers in this region were not reconstructed.

Figure 4.26 presents the rigid motion of the sail markers in relation to the hull markers. This motion is computed by aligning two reconstructions centered in different frames in relation to the hull markers. The distance between the central frames of the reconstructions is 15 frames, i.e., 0.5 second (frames 457 and 472). We note that the motion occurs mainly on the sail top, which is coherent with the sail dynamics and confirmed by domain experts as the expected behavior.

## Wind changing video results

Since our method estimate the mean sail shape during a time interval, it is important the sail shape keeps as stable is possible during this period. However, as presented in Section 4.3.1, the wind changes during some videos, modifying the sail shape (Figure 4.14). In this section we discuss the results of our method for the clip “near\_4k\_08.mp4”, which presents wind changing. The reconstruction was performed using the parameters previously chosen ( $n = 20$ ,  $s = 10$  and  $m = 5$ ).

Figure 4.27 presents a visualization of the reconstruction of the frame 219 from two views. We note that the sail region near to the luff are incorrectly reconstructed.

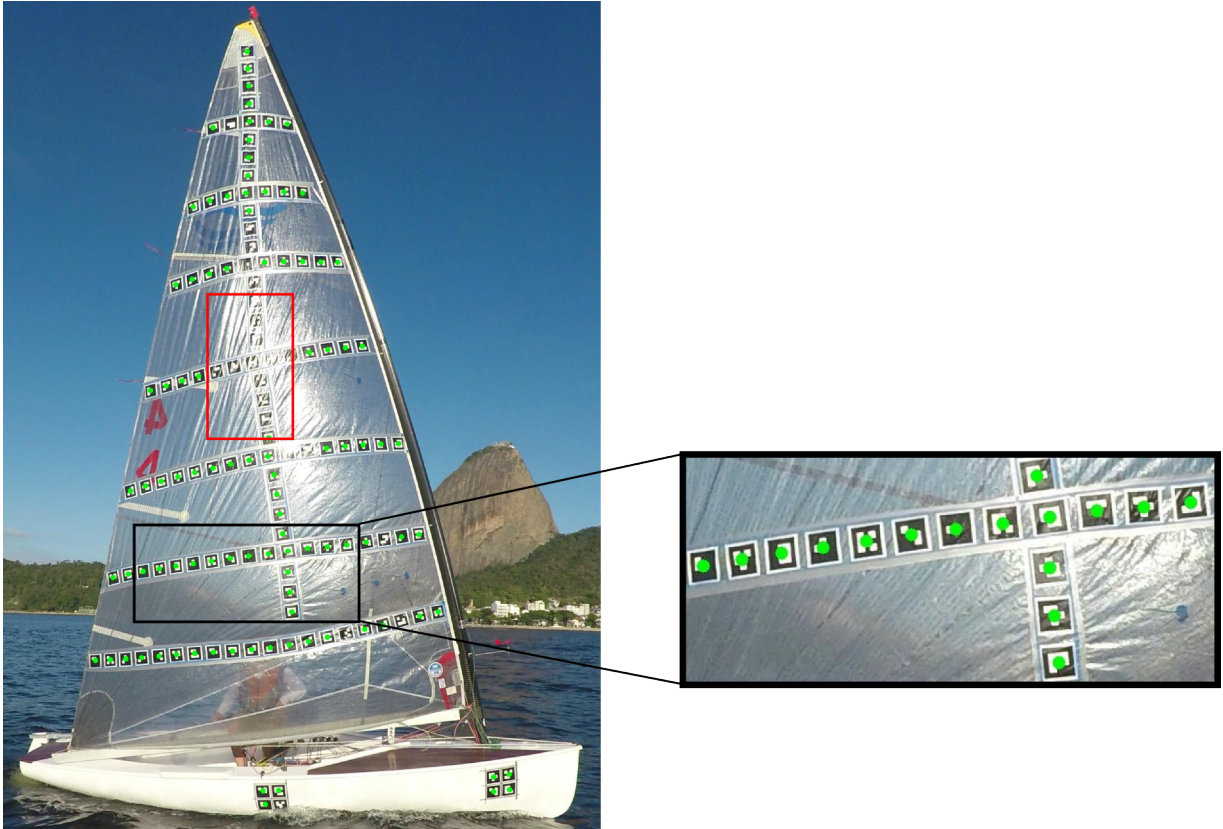


Figure 4.25: Reconstructed points reprojected on the central frame 457 for clip “near\_4k\_17”. The red rectangle highlighted shows a region where the natural illumination obfuscate some markers.

This is just the sail region that deforms with the wind change, as depicted in Figure 4.14.

Figure 4.28 shows the reconstructed markers centers reprojected on the frame 219. We observe that the centers are not reprojected in the expected position in the region where the sail shape changes. However, the remaining markers are correctly reprojected on the markers centers.

Concerning to the reconstruction metrics, Figures 4.29 and 4.30 present the comparison between the clips “near\_4K\_17.mp4” and “near\_4k\_08.mp4” for the area and reprojected errors, respectively. We observe that the errors were greater for all the criterion for the clip “near\_4k\_08.mp4”. This confirms quantitatively that our algorithm does not work properly under wind changing conditions. On the other hand, the clip “near\_4k\_08.mp4” presents a high reconstruction ratio (85%) since the conditions of distance and illumination are favorable. Summarizing, the sail shape stability is essential for the correct working of our method. Even a clip recorded in conditions that allow the markers detection is not correctly reconstructed if the wind changes the sail shape.

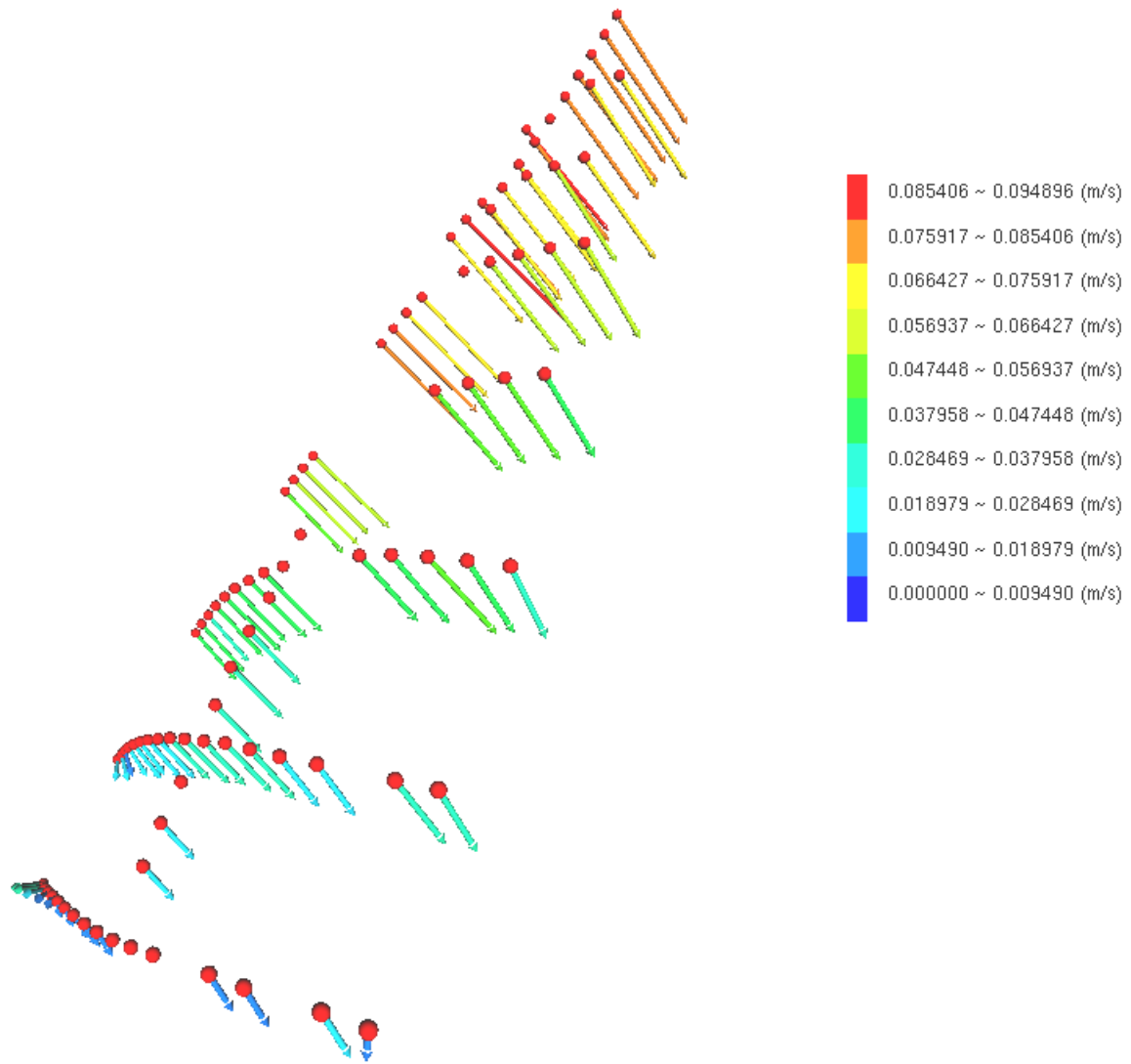


Figure 4.26: Sail markers rigid motion in relation to the hull markers. The reconstructions central frames are separated by 0.5 seconds (frames 457 and 472). The movement is consistent with the expected behavior, that is, larger motion at the top of the sail.

### Strong reflection video results

Our sailing videos were recorded under natural illumination condition, which are not controllable. As described in Table 4.1, several videos presented a strong reflection. To illustrate the effect of this issue in our method, Figure 4.31 presents the visualization of reconstruction of the frame 246 of the clip “far\_4k\_14.mp4” from two views. We note that many markers could not be reconstructed due the reflection.

Figure 4.32 shows the markers centers reprojected on the central frame. Analyzing the image we note that, although many markers were not reconstructed due to the reflection, the few reconstructed markers are reprojected in their expected positions at the markers centers.

It is interesting to note that the reflection makes marker detection difficult, re-

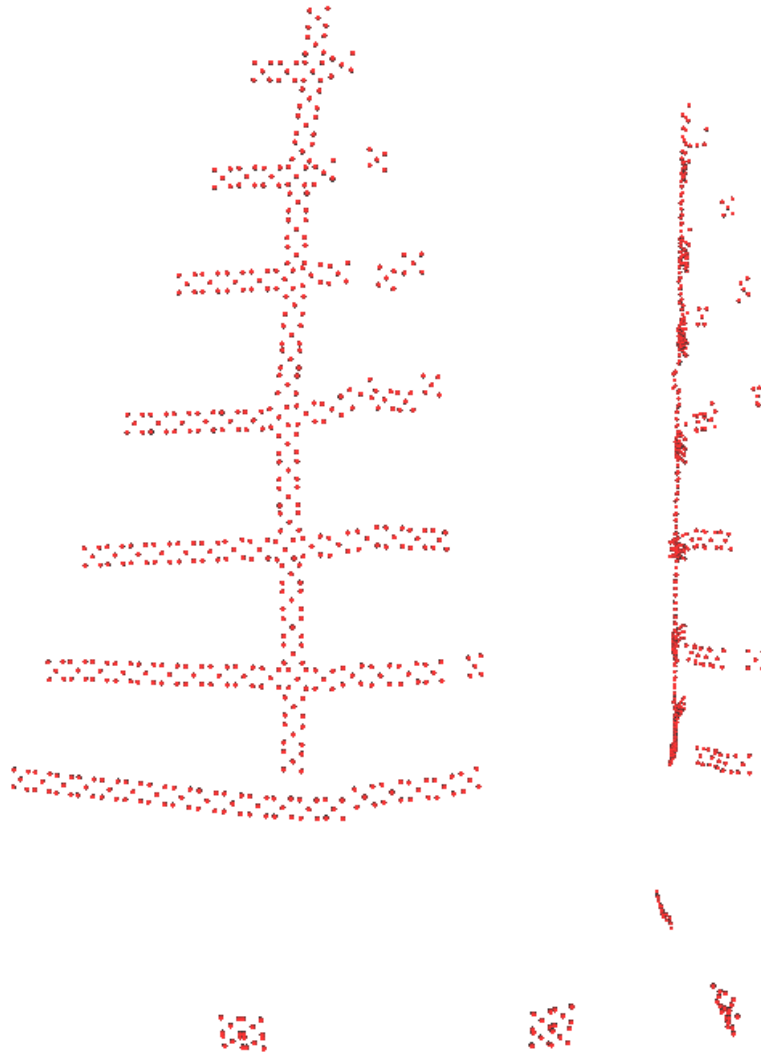


Figure 4.27: Visualization of the reconstruction of the clip “near\_4k\_08.mp4” centered in the frame 219 from two views.

ducing the reconstruction ratio, but it does not affect the quality of the reconstructed markers. Figures 4.33 and 4.34 compare the area and reprojeted errors respectively for the clips “near\_4k\_17.mp4” and “far\_4k\_14.mp4”. The charts show that the two clips present similar errors. For some criterion the clip “far\_4k\_14.mp4” presents even smaller values.

### Capture angle and distance issues

The capture angle is another issue that makes difficult the markers detection. Figure 4.35 shows the reconstructed markers centers of the frame 205 of the clip “near\_4k\_07.mp4” reprojeted on the respective frame. We note that, beyond the markers on the top that were obfuscated, the markers at the luff region were not detected due the bad capture angle. But as well as the example of the clip “far\_4k\_14.mp4” the few reconstructed points are correctly reprojeted to the markers centers.





Figure 4.28: Reconstructed points reprojected on the central frame 219 for clip “near\_4k\_08”.

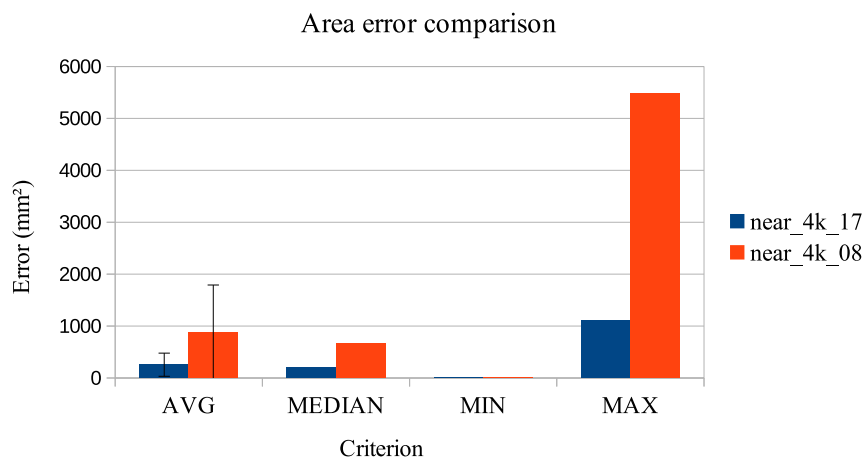


Figure 4.29: Area error criterion comparison between “near\_4k\_17.mp4” and “near\_4k\_08.mp4” clips.



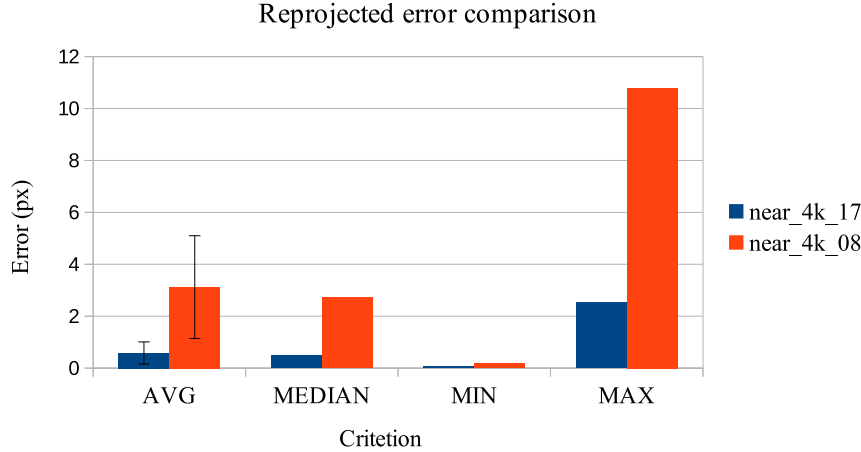


Figure 4.30: Reprojected error criterion comparison between “near\_4k\_17.mp4” and “near\_4k\_08.mp4” clips.

The reconstructed points of frame 205 of the clip “near\_4k\_07.mp4” are presented in Figure 4.36 from two views. The visual analysis of these points indicates they are correctly reconstructed.

Another issue that should be considered for our method is the capture distance. Markers can not be detected from videos recorder from a great distance. For the clips assigned as “Too far” in the Table 4.1 our reconstruction rate was zero or smaller than 10%. Thus, we conclude that the reflection, the capture angle and distance are important issues that influence the markers detection and, consequently, the detection ratio.

#### 4.3.4 Qualitative feedback from domain expert

Our results were analyzed by the naval architect and professor of UFRJ Alexandre Alho (alexandrealho@poli.ufrj.br), and the experienced sailor Jorge Rodrigues (jr-silva@gmail.com), in order to have feedback from domain experts. They analyzed results for the reconstruction of the clip “near\_4k\_17.mp4”. Figure 4.37 shows the profiles of the sail sections generated by the naval architect using the ANSYS [46] software from the markers centers of our reconstruction data. They observed that, in general, the shape of the profiles of the sail sections are very satisfactory. However, some distortions are observed near the boom (in red). Nonetheless, it is not clear if these are reconstruction errors or the sails actual shape since this region is subject to significant interference from the mast and the boom. Furthermore, some misalignments between the profiles are observed. The same observation was formulated about the initial and final points of the profiles. However, we note these misalignments result from the actual markers positioning on the sail. Therefore,

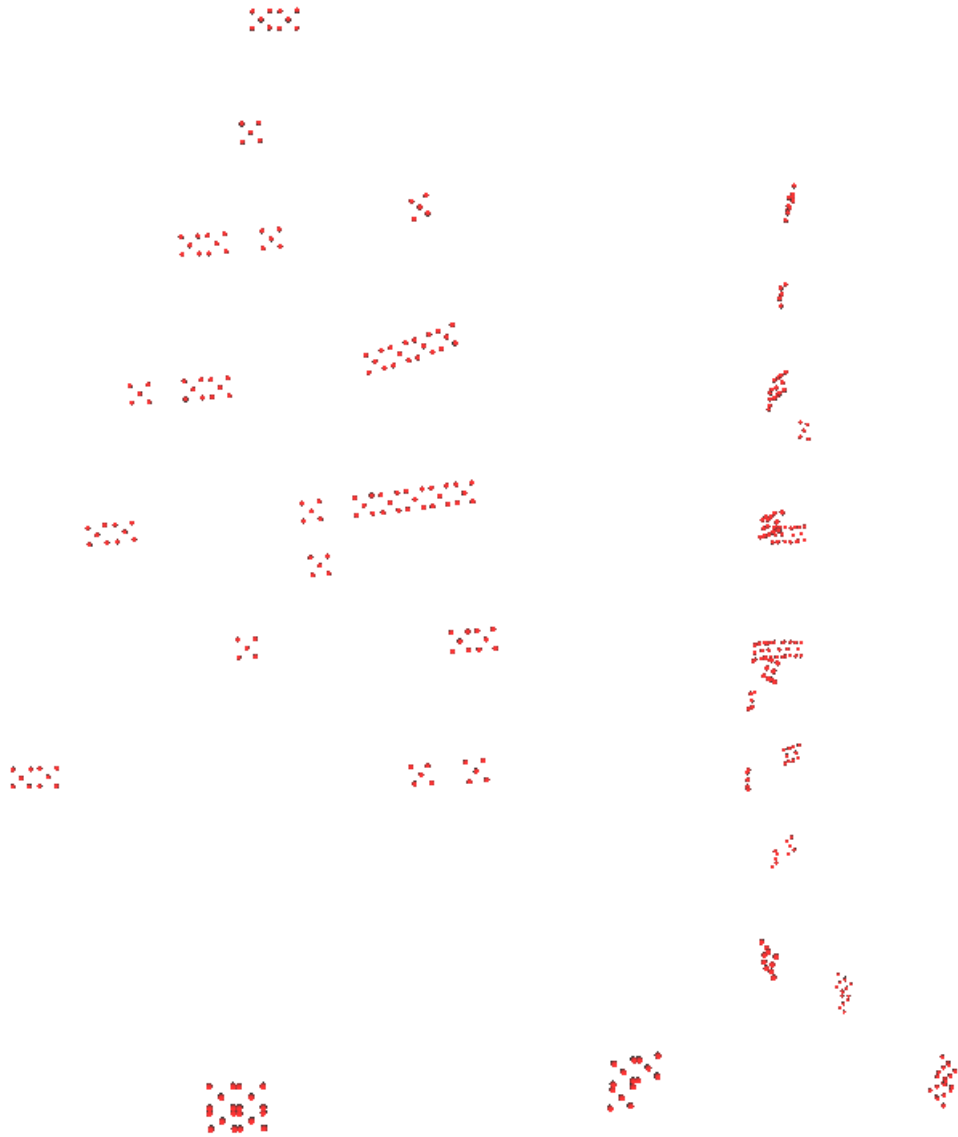


Figure 4.31: Visualization of the reconstruction of the clip “far\_4k\_14.mp4” centered in the frame 246 from two views.

the per points reconstruction quality is fully satisfactory to generate the sail shape. Nevertheless, it was suggested that it is important to achieve additional information about the sail bounds to have a more useful reconstruction for simulation and design evaluation purposes, and a more careful positioning of the markers to increase the profile reconstruction quality.

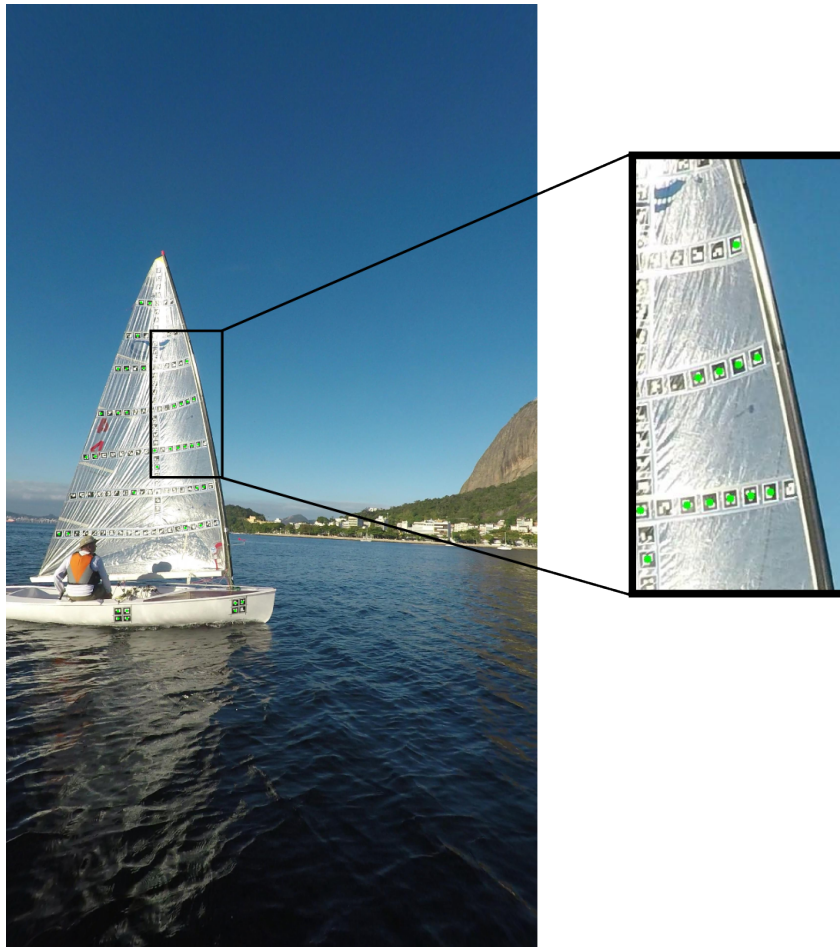


Figure 4.32: Reconstructed points reprojected on the central frame 246 for clip “far\_4k\_14”.

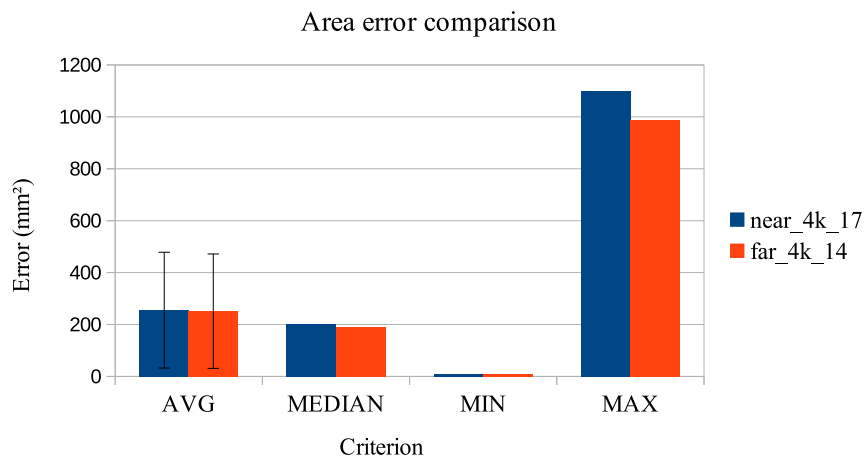


Figure 4.33: Area error statistics comparison between “near\_4k\_17.mp4” and “far\_4k\_14.mp4” clips.

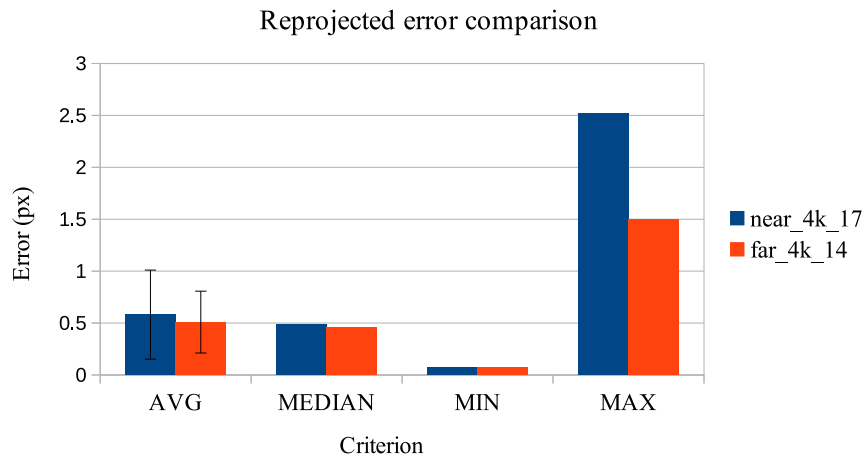


Figure 4.34: Reprojected error statistics comparison between “near\_4k\_17.mp4” and “far\_4k\_14.mp4” clips.

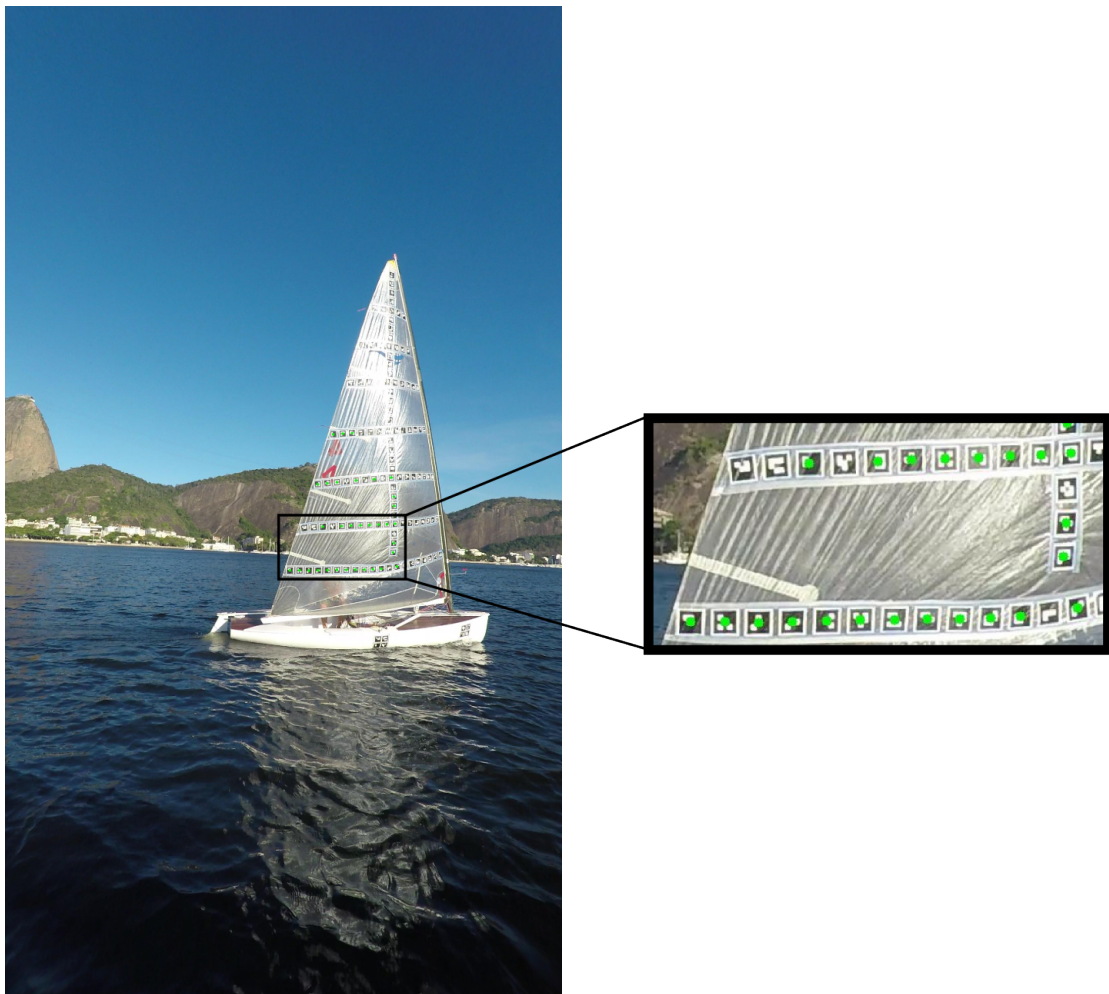


Figure 4.35: Reconstructed points reprojected on the central frame 205 for clip “far\_4k\_14”.

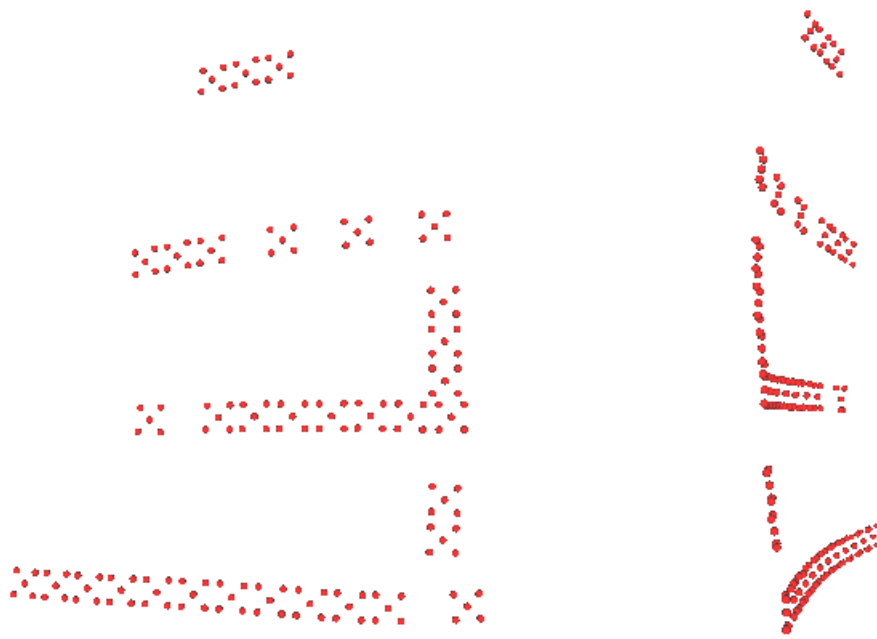


Figure 4.36: Visualization of the reconstruction of the clip “near\_4k\_07.mp4” centered in the frame 205 from two views.

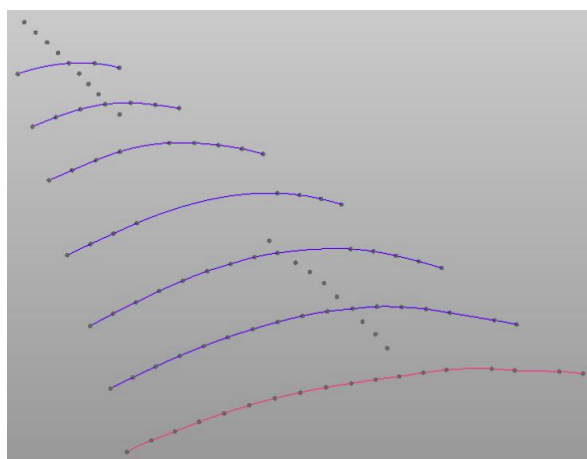


Figure 4.37: Profile of the sail sections. The lower curve in red presented some distortion, but we were unable to precise its source.

# Chapter 5

## Conclusion

In this work, we proposed a methodology for capturing the sail shape using a single video camera and passive markers. Our method is mostly non-invasive, as even though we still have to stick the markers onto the sail, we do not interfere with the sailing. For naval architecture purposes it is important to achieve the sail mean shape during a time interval, during which the sailor maintains the sail as stable as possible. Our main hypothesis is that the sail shape does not change significantly during the time period used for the reconstruction. Based on this, we proposed a method to estimate the sail mean shape from the markers position extracted along the interval. Our method is simple and very low-cost, since we need only passive markers and a single camera. Furthermore, our reconstruction is sparse by design, since just a few points on the sail surface are enough for the naval architects to extract reconstruct its shape. In fact, they point out that a few well placed and well recovered points is a much better input for them, than a dense reconstruction from which they would have to manually remove many points that where not precisely reconstructed.

After fixing the markers on the sail surface, they are captured from another boat. The markers information is extracted from the images and the selected frames are registered to a global reference system. This registered data is used to estimate the mean position of each marker during the interval, and is further optimized by a Bundle Adjustment algorithm, resulting in our final reconstruction.

Preceding our real sailing tests, we performed several laboratory experiments. These experiments allowed us to evaluate the accuracy and limitations of the marker detection method. Furthermore, we ran many tests against rigid objects, where the ground truth is known and precise reconstruction metrics could be extracted. We achieved an error smaller than 1 mm in this controlled environment.

For the real validation of our method, we recorded several videos of a Finn class sail in two situations: ashore and sailing. These videos compose our dataset, which we have made available at <http://www.lcg.ufrj.br/sail3D>. In addition to the

original videos, we also split them into smaller clips and separated them by capture distance. We also divided the clips in categories presented wind changes, reflection and capture angle. We believe that the creation of such dataset may be of immense value for other researchers in this area.

The dataset clips were tested using our method and the results were quantitatively evaluated by analyzing the markers' areas and the reprojected errors. We noticed that for stable videos the maximum area error was around 10% regarding the marker area, and the maximum reprojected error was around 2.5%. Qualitatively, we note that the reconstructed points were correctly reprojected at the central frame. Furthermore, we estimated the sail rigid motion between two reconstructions and observed that the movement is coherent with the sail dynamics.

Some videos presented wind changes, which modifies the sail shape. As expected, the quantitative and qualitative results for these videos were worse than the stable case, which confirm our hypothesis that the sail needs to be as stable as possible during the capture. Other limitations of our method are the reflection, and the capture distance and angle. Markers that are obfuscated by sun light, or recorded from a large distance or in a bad angle are not detected from the images and, consequently, are not reconstructed. However, even in a video that presents these issues, the markers captured from good conditions are correctly reconstructed. Moreover, we admit that the reflection problem was mostly due to a design issue, since a simple polarization filter could have been of great aid.

Our reconstruction result was evaluated by domain experts and were considered very satisfactory, and we conclude that our reconstructions were sufficiently accurate to be used for a real application. Moreover, our system can be easily applied on other types of boats, and even other kinds of surface, such as the boat hull.

Albeit the promising results, there are many possible improvements. We can improve the positioning of the markers on the sail, and fix markers on the sail bounds (the foot, the luff and the leech) to improve the final profile reconstruction. We can use filters to deal to the reflection issue. Our capture step can be modified to adapt to different boats and sails. We can capture a large sail or more than one sail by simultaneously use two or more auxiliary boats, or by capturing them separately. Finally, it would be possible to use a drone to record the sail from a better angle, but that would imply in increasing the cost of the system.

Several improvements can be implemented in our reconstruction method. We can introduce specific conditions for the Bundle Adjustment, using known constraints of our problem. Another criterion to search optimized frames for the reconstruction can be evaluated, selecting frames based in their reconstruction quality instead of quantity of markers. Actually, our system is very modular, and we could try new methods for each step without many issues.

# Bibliography

- [1] JOJIC, N., HUANG, T. S. “Estimating Cloth Draping Parameters from Range Data”. In: *Proceedings of International Workshop on Synthetic-Natural Hybrid Coding and 3-D Imaging*, pp. 73–76, 1997.
- [2] PRITCHARD, D., HEIDRICH, W. “Cloth motion capture”. In: *Proceedings of Computer Graphics Forum*, v. 22, pp. 263–271. Wiley Online Library, 2003.
- [3] SALZMANN, M., PILET, J., ILIC, S., et al. “Surface deformation models for nonrigid 3D shape recovery”, *Transactions on Pattern Analysis and Machine Intelligence*, v. 29, n. 8, pp. 1481–1487, 2007.
- [4] HAYASHI, T., DE SORBIER, F., SAITO, H. “Texture overlay onto non-rigid surface using commodity depth camera”. In: *Proceedings of International Conference on Computer Vision Theory and Applications*, 2012.
- [5] PILET, J., LEPETIT, V., FUA, P. “Fast non-rigid surface detection, registration and realistic augmentation”, *International Journal of Computer Vision*, v. 76, n. 2, pp. 109–122, 2008.
- [6] HILSMANN, A., EISERT, P. “Tracking and retexturing cloth for real-time virtual clothing applications”. In: *Proceedings of International Conference on Computer Vision/Computer Graphics Collaboration Techniques and Applications*, pp. 94–105. Springer, 2009.
- [7] BRUNET, F., HARTLEY, R., BARTOLI, A., et al. “Monocular template-based reconstruction of smooth and inextensible surfaces”. In: *Proceedings of Asian Conference on Computer Vision*, pp. 52–66. Springer, 2010.
- [8] BARTOLI, A., GÉRARD, Y., CHADEBECQ, F., et al. “On template-based reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces”. In: *Proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 2026–2033. IEEE, 2012.



- [9] NGO, D. T., ÖSTLUND, J., FUA, P. “Template-based monocular 3D shape recovery using laplacian meshes”, *Transactions on Pattern Analysis and Machine Intelligence*, v. 38, n. 1, pp. 172–187, 2016.
- [10] YU, R., RUSSELL, C., CAMPBELL, N. D., et al. “Direct, Dense, and Deformable: Template-Based non-Rigid 3D Reconstruction from RGB Video”. In: *Proceedings of International Conference on Computer Vision*, pp. 918–926. IEEE, 2015.
- [11] LIU, Y., CHEN, Y. Q. “Joint reconstruction of 3D shape and non-rigid motion in a region-growing framework”. In: *Proceedings of International Conference on Computer Vision Workshops*, pp. 1578–1585. IEEE, 2011.
- [12] CLAUSS, G., HEISEN, W. “CFD analysis on the flying shape of modern yacht sails”. In: *Proceedings of International Congress of the International Maritime Association of the Mediterranean*, p. 87, 2006.
- [13] LE PELLE, D., MODRAL, O. “V-Spars: A Combined Sail and Rig Shape Recognition System Using Imaging Techniques”. In: *Proceedings of High Performance Yacht Design Conference*, pp. 2–4, 2008.
- [14] GRAF, K., MÜLLER, O. “Photogrammetric investigation of the flying shape of spinnakers in a twisted flow wind tunnel”. In: *Proceedings of Chesapeake Sailing Yacht Symposium*, 2009.
- [15] MAUSOLF, J., DEPARDAY, J., GRAF, K., et al. “Photogrammetry based flying shape investigation of downwind sails in the wind tunnel and at full scale on a sailing yacht”. In: *Proceedings of Chesapeake Sailing Yacht Symposium*, pp. 33–43, 2011.
- [16] FOSSATI, F., MAINETTI, G., MALANDRA, M., et al. “Offwind Sail Flying Shapes Detection”. In: *Proceedings of High Performance Yacht Design Conference*, 2015.
- [17] DEPARDAY, J., BOT, P., HAUVILLE, F., et al. “Full-scale flying shape measurement of offwind yacht sails with photogrammetry”, *Ocean Engineering*, v. 127, pp. 135–143, 2016.
- [18] FERREIRA, P., CAETANO, E., PINTO, P. “Real-time flying shape detection of yacht sails based on strain measurements”, *Ocean Engineering*, v. 131, pp. 48–56, 2017.

- [19] LOMBARDI, M., CREMONESI, M., GIAMPIERI, A., et al. “A strongly coupled fluid-structure interaction model for wind-sail simulation”. In: *Proceedings of Performance Yacht Design Conference*, 2012.
- [20] DURAND, M., LEROYER, A., LOTHODÉ, C., et al. “FSI investigation on stability of downwind sails with an automatic dynamic trimming”, *Ocean Engineering*, v. 90, pp. 129–139, 2014.
- [21] CARCERONI, R. L., KUTULAKOS, K. N. “Toward recovering shape and motion of 3D curves from multi-view image sequences”. In: *Conference on Computer Vision and Pattern Recognition*, v. 1, pp. 192–197. IEEE, 1999.
- [22] CARCERONI, R. L., KUTULAKOS, K. N. “Multi-view scene capture by surfel sampling: From video streams to non-rigid 3D motion, shape and reflectance”, *International Journal of Computer Vision*, v. 49, n. 2-3, pp. 175–214, 2002.
- [23] LOWE, D. G. “Object recognition from local scale-invariant features”. In: *Proceedings of International Conference on Computer Vision*, v. 2, pp. 1150–1157. Ieee, 1999.
- [24] SALZMANN, M., FUA, P. “Linear local models for monocular reconstruction of deformable surfaces”, *Transactions on Pattern Analysis and Machine Intelligence*, v. 33, n. 5, pp. 931–944, 2011.
- [25] SHIMIZU, N., YOSHIDA, T., HAYASHI, T., et al. “Non-rigid surface tracking for virtual fitting system”. In: *Proceedings of International Conference on Computer Vision Theory and Applications*, 2013.
- [26] PERRIOLLAT, M., HARTLEY, R., BARTOLI, A. “Monocular template-based reconstruction of inextensible surfaces”, *International journal of computer vision*, v. 95, n. 2, pp. 124–137, 2011.
- [27] FURUKAWA, Y., PONCE, J. “Accurate, dense, and robust multiview stereopsis”, *Transactions on Pattern Analysis and Machine Intelligence*, v. 32, n. 8, pp. 1362–1376, 2010.
- [28] FABBRI, R., KIMIA, B. “3D curve sketch: Flexible curve-based stereo reconstruction and calibration”. In: *Proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 1538–1545. IEEE, 2010.

- [29] USUMEZBAS, A., FABBRI, R., KIMIA, B. B. “From multiview image curves to 3D drawings”. In: *Proceedings of European Conference on Computer Vision*, pp. 70–87. Springer, 2016.
- [30] USUMEZBAS, A., FABBRI, R., KIMIA, B. B. “The Surfacing of Multiview 3D Drawings via Lofting and Occlusion Reasoning”. In: *Proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 2980–2989, 2017.
- [31] LIU, Y., CHEN, Y. Q. “3D tracking of deformable surface by propagating feature correspondences”. In: *Proceedings of International Conference on Pattern Recognition*, pp. 2202–2205. IEEE, 2012.
- [32] BLANZ, V., SCHERBAUM, K., SEIDEL, H.-P. “Fitting a morphable model to 3D scans of faces”. In: *Proceedings of International Conference on Computer Vision*, pp. 1–8. IEEE, 2007.
- [33] ZOLLHÖFER, M., NIESSNER, M., IZADI, S., et al. “Real-time non-rigid reconstruction using an RGB-D camera”, *Transactions on Graphics*, v. 33, n. 4, pp. 156, 2014.
- [34] NEWCOMBE, R. A., FOX, D., SEITZ, S. M. “Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time”. In: *Proceedings of Conference on Computer Vision and Pattern Recognition*, pp. 343–352, 2015.
- [35] BRONSTEIN, M. M., BRUNA, J., LECUN, Y., et al. “Geometric deep learning: going beyond euclidean data”, *Signal Processing Magazine*, v. 34, n. 4, pp. 18–42, 2017.
- [36] LE MAÎTRE, O., HUBERSON, S., DE CURSI, J. S. “Application of a non-convex model of fabric deformations to sail cut analysis”, *Journal of Wind Engineering and Industrial Aerodynamics*, v. 63, n. 1-3, pp. 77–93, 1996.
- [37] AMBROZIAK, A., KŁOSOWSKI, P. “Polyester sail technical woven fabric behaviour under uniaxial and biaxial tensile tests”, *Journal of Theoretical and Applied Mechanics*, v. 56, n. 1, pp. 227–238, 2018.
- [38] RENZSCH, H., GRAF, K. “An experimental validation case for fluid-structure-interaction simulations of downwind sails”. In: *Proceedings of Chesapeake Sailing Yacht Symposium*, 2013.
- [39] TRIGGS, B., MCLAUHLAN, P. F., HARTLEY, R. I., et al. “Bundle adjustment—a modern synthesis”. In: *Proceedings of International Workshop on Vision Algorithms*, pp. 298–372. Springer, 1999.

- [40] KATO, H., BILLINGHURST, M. “Marker tracking and hmd calibration for a video-based augmented reality conferencing system”. In: *Proceedings of International Workshop on Augmented Reality*, pp. 85–94. IEEE, 1999.
- [41] GARRIDO-JURADO, S., MUÑOZ-SALINAS, R., MADRID-CUEVAS, F. J., et al. “Automatic generation and detection of highly reliable fiducial markers under occlusion”, *Pattern Recognition*, v. 47, n. 6, pp. 2280–2292, 2014.
- [42] ROUSSELON, N. “Optimization for sail design”. In: *Proceedings of Mode Frontier Conference*, 2008.
- [43] COHEN-OR, D., GREIF, C., JU, T., et al. *A Sampler of Useful Computational Tools for Applied Geometry, Computer Graphics, and Image Processing: Foundations for Computer Graphics, Vision, and Image Processing*. CRC Press, 2015.
- [44] KÜMMERLE, R., GRISETTI, G., STRASDAT, H., et al. “g 2 o: A general framework for graph optimization”. In: *Proceedings of International Conference on Robotics and Automation*, pp. 3607–3613. IEEE, 2011.
- [45] DAQRI. “ARToolKit Documentation”. 2015. Disponível em: <<http://artoolkit.org/documentation/>>. Acesso em: 01 ago. 2016.
- [46] MECHANICAL, A. R. “ANSYS”. Disponível em: <<https://www.ansys.com/>>. Release 18.1.