Delft University of Technology

Monocular 3D reconstruction of sail flying shape using passive markers

Maciel, Luiz ; Marroquim, Ricardo; Vieira, Marcelo ; Ribeiro, Kevyn ; Alho, Alexandre

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

ORIGINAL PAPER

Monocular 3D reconstruction of sail flying shape using passive markers

Luiz Maciel[1,2,3] · Ricardo Marroquim[3,4] · Marcelo Vieira[1] · Kevyn Ribeiro[1] · Alexandre Alho[3]

Abstract

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 an interval where the sailor maintains the sail as stable as possible. The average is further improved by a Bundle Adjustment algorithm. 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.

Keywords 3D Reconstruction · Sail · Monocular · Passive Markers

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 sailing yacht design application domain, more specifically, on recovering the flying shape of a sail under sailing conditions.

✉ Ricardo Marroquim
  r.marroquim@tudelft.nl

  Luiz Maciel
  luiz.maurilio@ice.ufjf.br

  Marcelo Vieira
  marcelo.bernardes@ufjf.edu.br

  Kevyn Ribeiro
  swhants@ice.ufjf.br

  Alexandre Alho
  alexandrealho@poli.ufrj.br

1   Universidade Federal de Juiz de Fora, Juiz de Fora, Brazil

2   Instituto Federal de Minas Gerais, Ouro Branco, Brazil

3   Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil

4   Computer Science Department, Delft University of Technology, Delft, The Netherlands

The understanding of the aerodynamics of sails is a fundamental key to predict the performance of a racing yacht. In the last decades, a great research effort was employed for a better knowledge of the behavior of sails in response to the ongoing demand for higher racing performance. Among all the aspects related to sail design, the prediction of the aerodynamic forces produced and its correlation with the flying shape and the pressure distribution along the sail are by far the topics of most interest.

Sail performance predictions have been historically made through semi-empirical methods, based largely on experimental data obtained from both wind tunnel and full-scale testing. Efforts were mostly concentrated on developing aerodynamic force models to be implemented in a velocity prediction program (VPP) [4,33]. Even though VPP is a well-established technology, it is basically applied to predict performance under steady-state conditions. In recent years, it became evident the claim of more well-founded deductions of forces on sails, not only in full-scale but also under real sailing conditions.

Recent developments in sail design are extensively based on computational fluid dynamics (CFD) simulations. CFD models can be used to test a number of candidate designs under different trims configurations and environmental conditions, in less time and with lower costs in comparison with experimental methods. However, there are a number

of aspects related to the simulation of the flow around yacht sails that demand a more complex approach. The intrinsic unsteady regime of the wind loads coupled with the flexibility characteristics of sails' materials impose that the computations of the aerodynamic forces on a sail consist essentially in a fluid-structure interaction problem. Notwithstanding the benefits of CFD simulations, some authors [2] point out the importance of full-scale experiments for the validation of the fluid-structure interaction models. Full-scale testing allows obtaining the real shape of the sail under the action of the wind loads and, consequently, helps towards a better understanding of the behavior of the sail material and its influence on the flying shape.

Another observation that contributes to the importance of full-scale testing is that the sail design, including computer-aided, 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 is discussed in Sect. 2. Furthermore, it has been likewise noted that tests performed in the controlled environment of a wind tunnel, even at full-scale, do not reflect precisely the true conditions of a real scenario.

In spite of the flying shape being essential to predict the actual performance of a sail, obtaining sail section profiles under sailing conditions has never been an easy task to accomplish. Yacht motion and variations in wind speed and direction, for example, are all factors that affect the sail shape and, consequently, the measurement of its section profiles. Some of these factors have high variability in time, hence, instantaneous configurations are too noisy for individual analysis. In fact, the so defined flying shape represents a snapshot, and in terms of sail performance analysis the results are founded in steady-state aerodynamics. In order to minimize the influence of the natural fluctuations in environmental conditions, the sail shape can be averaged over a period of time, raging from few to tens of seconds. Despite the obvious limitations, such an approach is accurate enough to allow the prediction of the sail performance for both design and optimum trimming analysis purposes, being currently of common use in the design of large racing yachts.

Nevertheless, sailing racing is a very competitive sport, and the demand for high racing performance is not exclusive of large yachts racing, being observed in sailing dinghy racing as well. Technology developments over the years have also affected many aspects of the modern sailing dinghy racing, including hull design, sail materials and sail plan. With this in mind, the current research proposes a simple low-cost technique for full-scale measurements of sail flying shapes for sailing racing dinghies. Such a technique offers a wider range of application for both design and sail trimming, allowing researchers and designers to carry out full-scale tests at reasonable costs, in order to improve racing performance.

To summarize, our main contribution is a low-cost and simple acquisition method for sail shape reconstruction that uses a single camera and passive markers. We provide a detailed and systematic description of the system. The precision of the method was verified against a battery of tests using rigid objects for which ground truth data is available. In addition, a series of parameters were thoroughly evaluated in regards to commonly employed reconstruction metrics. This parameterization and error analysis were paramount to pinpoint the origin of errors and fine tune the method. Finally, the system was tested in full-scale and in real sailing conditions with feedback from domain experts. Our method was tested for a Finn class sail, but it can be adapted for other sails and boats by modifying the capture setup accordingly.

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.

1.1 Boat terminology

We briefly 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 Fig. 1:

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

2 Related works

Jojic and Huang [17] 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 [3,7,16,27–29,34,35,38], others proposed multi-view approaches [22,23,30], while some made use of RGB-D devices [5,15,26,36,39]. 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 [3,7,22,23,27–30,34–36,38], requiring that the object presents a highly textured surface with distinguishable elements. In our work, we opted for passive markers, which
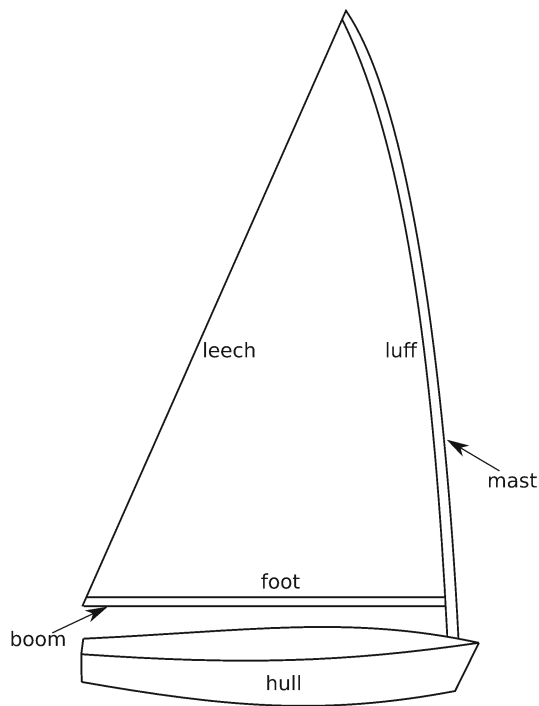
Fig. 1 Boat diagram indicating the terminology of some parts

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. [15] 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 [3,7,27,28,30,34,35]. 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 [1,20], hence, such specific evaluation goes far beyond the scope of the current work.

Other reconstruction approaches apply a temporal smooth constraint [5,22,23,26,38,39]. This constraint assumes the object deforms as minimum as possible over time and is often accompanied by an as rigid as possible constraint [5,22,23,26,27,29,38,39], 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 (Sect. 3.4). There are works that employ machine learning methods, such as PCA [5,15,34–36] and deep learning [6], 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.

Besides generic surface reconstruction methods, some specific approaches for sails were introduced in recent years. Clauss and Heisen [8] proposed to capture the flying shape of the sails of an yacht DYNA by fixing 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 along the boat. After identifying the markers on the images, their location 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.

The Visual Sail Position And Rig Shape (VSPARS) software, popular among sail designers, was presented by Le Pelley and Modral [21]. They determine the 3D localization of colored stripes on the sails and colored points on the rig using three cameras fixed on the boat deck. The targets are extracted and their positions in a global coordinate system are estimated based on the hypothesis that the stripes are parallel to a horizontal plane when flying. This is, nonetheless, a strong hypothesis, which is not true for several apparent wind angles, according to some recent works [10,12]. 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 a wind tunnel. The sail is covered by coded passive markers and four cameras are arranged outside the boat. After preprocessing the images, they recover the markers' 3D positions using the *Photo Modeler Pro* photogrammetry software. 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 with the design shape. Mausolf et al. [24] 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. 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 [31] 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,

Table 1 Features of the sail reconstruction works

	[8]	[21]	[14]	[24]	[31]	[12]	[10]	[11]	Our method
Monocular									•
Multi-view	•	•	•	•	•		•		
Internal cameras	•	•					•		
External cameras			•	•	•				•
Wind tunnel	•	•	•	•	•				
Full scale	•	•		•			•		•
Black markers	•								
Colored markers		•					•		
Colored stripes		•							
Coded markers			•	•	•				•
Photo-grammetry	•		•	•	•		•		•
Active capture						•			
Error correction	•								•
Strain sensors								•	
Bundle Adjustment									•

the movement occurs mainly at the luff, but the paper does not give further detail on the reconstruction evaluation.

Fossati et al. [12] introduced another method to measure flying shapes in a wind tunnel at full-scale. They built an active capture device that rotates around an axis, brushing the whole sail area. This device retrieves a point cloud which is used to recover the sail corners, edges and sections. Precision and accuracy were verified by preliminary tests using known reference objects achieving the application requirements. The reconstructed sail shape was evaluated comparing the measurements retrieved against those provided by the design sail tool, achieving significant differences. As also noted by Mausolf et al. [24], these differences were associate with the trim adjustment. Unfortunately, the authors do not explicit any quantitative results in their paper.

Deparday et al. [10] 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 Modeler* software, which recovers the 3D positions of the markers using photogrammetry algorithms. 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 and concluded that a simulation using the designed shape is not representative of the real sailing conditions.

Recently, Ferreira et al. [11] 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 achieve the curvature of the sections. The estimated curvature may be sent to mobile devices and seen by the sailor in real time. They validate their method in laboratory conditions using a rigid model, but are still studying the influence of the sensors material on the aerodynamic of real flexible sails.

Table 1 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 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.

3 Proposed method

In this Section, the proposed method for solving the problem of sail shape estimation is described. It is composed of five steps, as depicted in Fig. 2, which will be presented in the next sections:

1. **Markers fixation** (Sect. 3.1): markers are chosen, printed and fixed on the sail. The fixation should ensure the markers will not drop during the sailing and the sail can be properly captured.
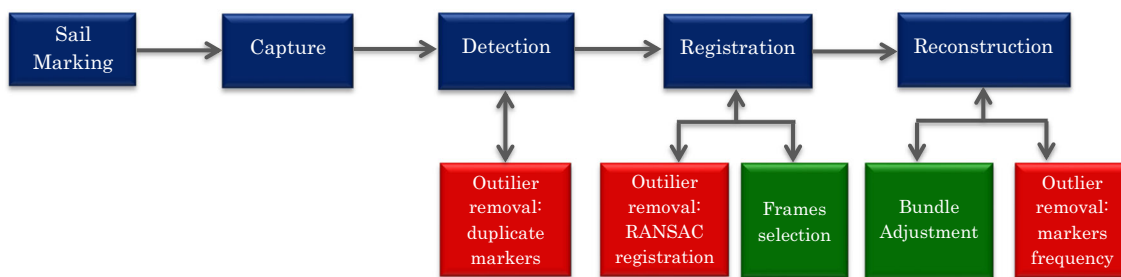
Fig. 2 Proposed method steps. Blue: main steps. Red: outlier removal internal steps. Green: internal selection and optimization steps

2. **Capture** (Sect. 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** (Sect. 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 provides 2D points on the image and the corresponding 3D points in the camera coordinate system.

4. **Registration** (Sect. 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 interval that will be used to estimate the mean sail shape. Furthermore, our registration performs a filtering step to remove outliers.

5. **Reconstruction** (Sect. 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 [37].

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 does not adjust any configuration in order for the boat to be as stable as possible, hence, 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.
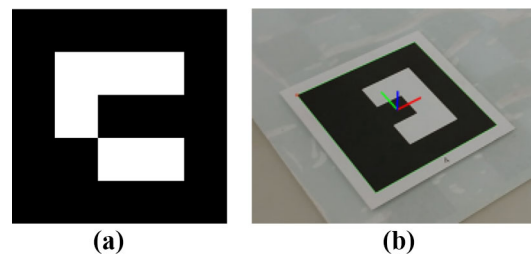

Fig. 3 ArUco Markers. **a** Example of used marker. **b** Detail of marker detection

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. We previously compared the detection robustness between two libraries: ARToolKit [18] and ArUco [13]. ArUco presented the best results for our tests. ArUco markers are square and binary, and the library allows creating a configurable markers dictionary by defining the number of markers and the number of bits for the inner pattern and the border. Markers are generated maximizing the inter-marker distance and the number of bit transitions. We performed experiments with different numbers of bits for the inner pattern and border size and achieved the best results for our application with 9 internal bits and 1 bit for the border. From the markers, it is possible to extract their image contours, 3D center positions and orientations in the camera's coordinate system (Fig. 3). The extracted data are the input for the next steps, as described in the following sections.

For naval architecture purposes, it is important to retrieve horizontal sections along the sail, since they convey well its general shape. For simulation and design purposes, the sail surface is mainly defined by horizontal curves [32]. Thus, markers were placed forming horizontal stripes in strategic positions pointed out by the naval architects. We also placed markers along a vertical line 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.

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 $V = \{v_i \mid v_i$ is the marker with index $i\}$ is the vertices set and $E = \{e_{ij} \mid e_{ij}$ is the edge connecting the vertices v_i and $v_j\}$ is the edges set. We established the adjacencies as shown in Fig. 4: markers on horizontal lines are connected to the markers on the 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 (Sect. 3.3). We define the topological distance between two vertices as the number of edges connecting them. The smaller the number of edges between two vertices, closer they are. For example, in Fig. 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.

It is important to emphasize that the total number of markers depends on the project (design) and the analysis objectives. More markers provide a more detailed graph and reconstruction, and redundancy may help in overcoming detection errors. The markers fixation should be performed carefully to avoid losing them during sailing, and 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 also fixed scotch tape along the markers border. However, if a marker does fall off, our method considers that this marker was not detected and the reconstruction proceeds normally. For our tests, the fixation of about 122 markers took around two hours.

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. Considering the Finn class, 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. This setup has disadvantages, however, such as the need for more cameras in order to capture the whole sail surface [8,21], and the perspective distortion of the images, especially on the sail top [10]. Positioning cameras in another boat allows to record the sail at a more perpendicular angle. It is a more generic and simple setup that can be used for a broader range of boats and can be arranged as to not interfere with the sailing of the tracked boat. The main challenge of capturing the sail is to keep the camera at a distance that
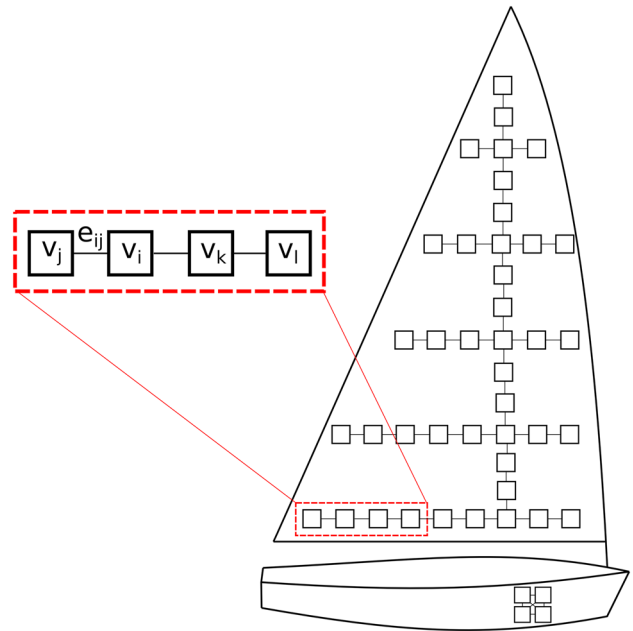

Fig. 4 Markers graph and a topological distance example

allows a good marker detection while avoiding illumination problems.

3.3 Detection

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 $\mathbf{t}_{k,0} \in \mathbb{R}^3$ and rotation $\mathbf{R}_k \in SO(3)$) is 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 $\mathbf{t}_{k,0}$. 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:

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

Fig. 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

Commonly, false positives arise during detection. Artifacts on the image may be confused with a marker, and markers may be mislabeled, as shown in Fig. 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 Sect. 3.1. For each index k and frame f, we have a set of candidate markers $C_{k,f} = \{M_{k,f}^i \mid M_{k,f}^i$ is a candidate for marker k 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 Fig. 5, the incorrect detection of 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 (Sect. 3.4.1) and reconstruction (Sect. 3.5) are more effective to remove outliers. Thus, we have chosen to handle only the duplicate markers in the detection step.

> **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**

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 i is discarded at a frame f, $M_{i,f}$ is removed from D_f.

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 Fig. 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 Fig. 6b.

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, since frames with small time differences are very similar and

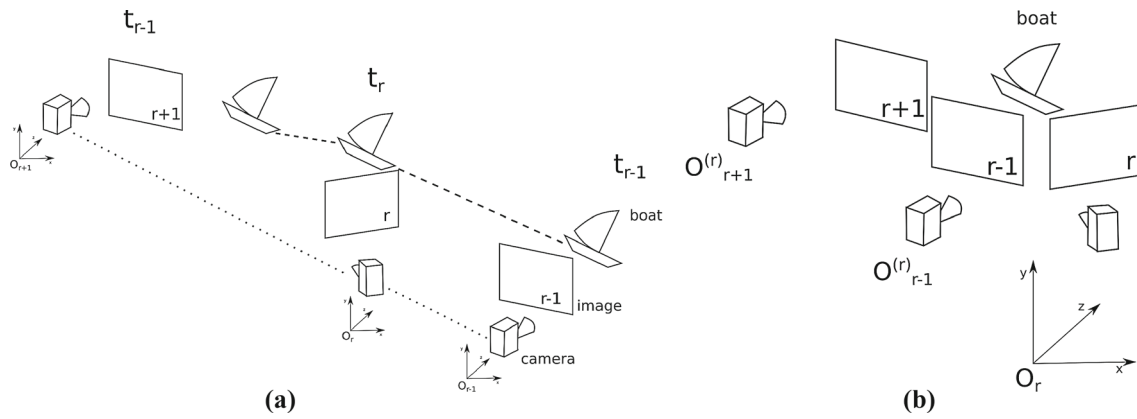**Fig. 6** Camera and sail dynamics during different time frames. **a** Before registration the camera, at each time step t_i a frame i is generated with an independent camera coordinate system O_i. Frame r is the reference frame at instant t_r, while $r - 1$ and $r + 1$ are frames before and after r, respectively (not necessarily consecutive). **b** After registration, all coordinate systems are registered against a global reference O_r at time t_r. This allows to transform all extracted 3D points to the same reference frame in order perform the averaging step

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, lets define the set that contains the selected $2n + 1$ frames as:

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

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 $\mathbf{p}_{k,0}^{(r)}$:

$$(\mathbf{R}_{f,r}, \mathbf{v}_{f,r}) = \arg\min_{\mathbf{R},\mathbf{v}} \sum_k ||(\mathbf{R} \cdot \mathbf{p}_{k,0}^{(f)} + \mathbf{v}) - \mathbf{p}_{k,0}^{(r)}||^2, \quad (1)$$

where k represents all markers indices such that $M_{k,f} \in D_f$ and $M_{k,r} \in D_r$, $\mathbf{R}_{f,r} \in SO(3)$ is the rotation and $\mathbf{v}_{f,r} \in \mathbb{R}^3$ is the translation that align f's reference system with r's. Eq. (1) is a least square problem which can be solved by Singular Value Decomposition (SVD) [9]. 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 Eq. (1) searches for a solution that best fits all markers, these outliers disturb the solution $(\mathbf{R}_{f,r}, \mathbf{v}_{f,r})$. It is important to filter out 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 Eq. (1) and find $(\mathbf{R}_{f,r}, \mathbf{v}_{f,r})$.

This RANSAC strategy is also used to select the n frames before and after frame r. Starting from frame r, we skip s frames backward 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 Sect. 4.3.

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

$$\mathbf{P}'_{k,f} = \mathbf{R}_{f,r} \cdot \mathbf{P}_{k,f} + \mathbf{v}_{f,r}, \quad (2)$$

where $\mathbf{R}_{f,r}$ and $\mathbf{v}_{f,r}$ are the rotation and translation between f and r obtained by Eq. (1) using D_f with hindering markers removed. Points from $\mathbf{P}'_{k,f}$ are at 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 $(\mathbf{R}_{f,r}, \mathbf{v}_{f,r})$,

but we opted to register them using Eq. (2) and evaluate them by the weighted average described in Sect. 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 $f \in S$.

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 the marker of index k was correctly detected. A marker needs to appear in a minimum number of frames β so that its position can be correctly optimized by the Bundle Adjustment (BA) algorithm [37] described in Sect. 3.5.1. To avoid optimization problems, if $|Q_k| < \beta$, $M_{k,f}$ is removed from D_f, for all $f \in S$. The threshold β is our frequency tolerance, and its value will be discussed in Sect. 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 filtering, the updated sets D_f are used to estimate the mean positions $\bar{\mathbf{P}}_k$ of the marker k. Notice that up to this point all positions were computed using the frame r as reference. These mean positions are iteratively computed from the initial mean (iteration 0):

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

where $k \in I$. After computing this initial mean, we start an iterative algorithm to compute a weighted averaged position [9] 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 [9]. 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 estimate by progressively penalizing points far from the mean. Thus, we can define a set:

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

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

3.5.1 Bundle adjustment

We further refine the mean points estimate using Bundle Adjustment (BA) [37]. It optimizes the points reconstructed in world space and the cameras poses by minimizing 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 transformations between pairs 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 $C = \{(\mathbf{R}_j, \mathbf{t}_j) \mid \mathbf{R}_j \in SO(3) \text{ and } \mathbf{t}_j \in \mathbb{R}^3\}$ of camera poses, and a set $Y = \{\mathbf{y}_{ij} \in \Omega \mid \mathbf{y}_{ij} \text{ is the image of point } \mathbf{w}_i \text{ by camera } j\}$. In our case,

$$W = \bar{P},$$

where $\bar{P}$ is the set of mean points defined in Eq. (3). Notice that $\bar{P}$ is a refined set of points in the world coordinate system that is the camera system of reference frame r.

For each $f \in S$, we need to find an initial estimate for the camera pose $(\mathbf{R}_f, \mathbf{t}_f)$ in relation to the world system. This estimate pose can be obtained 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 Eq. (1):

$$(\mathbf{R}_f, \mathbf{t}_f) = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{k \in D_f} \|(\mathbf{R} \cdot \bar{\mathbf{p}}_{k,0} + \mathbf{t}) - \mathbf{p}_{k,0}\|^2, \tag{4}$$

where $\mathbf{p}_{k,0} \in P_{k,f}$ is the center position of marker $M_{k,f}$, $k \in I$ before registration, $\bar{\mathbf{p}}_{k,0}$ is the world position of this

marker center, $\mathbf{R}_f \in SO(3)$ and $\mathbf{t}_f \in \mathbb{R}^3$. Solving Eq. (4) for each $j \in S$, we find our camera poses set:

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

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

Our image points set Y is defined as:

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

Thus, we apply the BA algorithm implemented by g2o library [19] using the sets W, C and Y 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 $\mathbf{t}_j \in C^*$ 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.

4 Experiments

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 stripes. Furthermore, 8 markers were fixed on the boat hull as depicted in Fig. 7a. 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.

4.1 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 (Fig. 7a). This scenario allowed more control over the capture distance and the illumination. We recorded a total of 20 ashore sequences, including 4K, 2.7K and 12MP camera resolutions. After this controlled scenario, we captured the sail in a real sailing environment

(Fig. 7b and c), totaling 28 sequences. Our original videos were divided in these two categories: ashore and sailing.

The original videos were split and classified into two main classes based on the capture distance to the sail: near or far. This division resulted in 39 clips with 4K resolution. Each clip presents particular features as listed in Table 2.

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. Figures 7b and c present examples of these situations. It is important to note that both reflection types can occur in the same clip. For future registrations, we can soften this issue using filters in the camera.

In some clips, the wind changes during the capture, modifying the sail shape. They were classified as "Wind change". Fig. 8 shows three frames of the clip "near_4k_08.mp4" with wind changes.

Some clips were recorded from a great distance, which makes markers detection very difficult. These clips are classified as "Too far". Furthermore, some clips were captured from an almost parallel angle in relation to the sail. They were classified as "Bad angle". Ideally, the capture should have an angle as perpendicular as possible to the sail.

## 4.2 General parameters evaluation

We used three evaluators to quantitatively assess 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 $\mathbf{p}$ 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.

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 Sect. 3.5, we described our iterative weighted average of the markers points. This average uses a Gaussian weight with parameter σ. We tested some σ values in the interval [0.1, 5.0] and observed its influence on the error evaluators. We noticed that σ is not sensitive for the reconstruction ratio, and values $\sigma \geq 0.6$ do not disturb e_a and e_r. Thus, we set

Fig. 7 Dataset video frames examples. **a** Ashore video frame. **b** Sailing video frame with weak reflection. **c** Sailing video frame with strong reflection

(a)   (b)   (c)
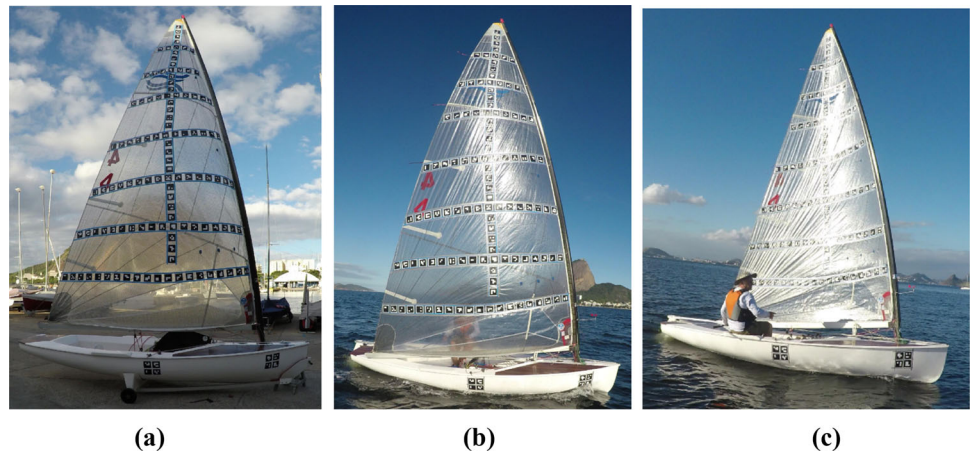
$\sigma = 0.6$ for our experiments. We also analyze the threshold β for the frequency filter by varying its value between 10% and 40%. Small values increase the number of reconstructed markers, but also increases e_a and e_r. The value $\beta = 30\%$ presented the best trade-off between reconstruction ratio and errors. We performed 30 iterations, which were enough for convergence in Eq. (3).

For the RANSAC strategy described in Sect. 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 mm slightly decrease the errors but considerably reduces the number of reconstructed markers. On the other hand, values above 100 mm increase the reconstruction ratio at the cost of increasing errors.

4.3 Reconstruction results

The clip "near_4k_17.mp4" is the longest sequence recorded in sailing conditions from a reasonable distance. This clip presents a good sail stability, parts with weak and strong reflection. Thus, it is considered the best baseline for the dataset reconstruction and all results presented in this section use this clip. It will be used in the next subsections for comparing the results with difficult clips.

Figure 9a presents a visualization of the reconstruction centered in the frame 457 from two view points. It shows all points for each reconstructed marker.

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

– For each reconstruction 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 frame
  – Compute the average, standard deviation, median, minimum and maximum over all e_a and e_r obtained in the frame;

– Compute the mean of e_a and e_r over all markers from all frames;
– Compute the mean of $\frac{n_r}{N}$ over all frames.

Our frame selection procedure described in Sect. 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 10a, b and c show the resulting statistics for the three evaluators. Figure 10 presents the mean statistics in function of total number of selected frames $|S|$. Notice that the error decreases with the increase in selected frames, but after 41 frames the variation is small. The average error was around 250 mm^2, which represents 2.5% of the marker area and the maximum around 1000 mm^2, which is 10% of the area. For the reprojected error, the error slightly increases as the number of frames increases. This is expected since we have more frames to be adjusted by the Bundle Adjustment. Despite this increase, the maximum error slightly changes after $|S| = 50$, stabilizing at around 2 pixels. The reconstruction ratio decreases with the total of selected frames, varying from 56.9% for 11 frames to 40.1% for 101 frames, i.e., as more frames are used for the reconstruction, fewer markers are reconstructed. The decrease is more accentuated after $|S| = 31$. Thus, we can summarize the analysis of Fig. 10a, b and c 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 $|S| = 31$.

Table 2 Sail dataset video features

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

Based on this analysis, we opted to use $n = 20$, i.e., selecting $|S| = 41$ frames for reconstruction. This value ensures small area errors without penalizing the reconstruction ratio.

Figure 11a, b and c presents the evaluation results in regards to the skip size s. By analyzing Fig. 11a, we note that small values perform poorly. This is explained by the frames similarity, since the frame rate is high in relation to the scene motion. If s is increased, a longer clip is necessary to select the frames, since the interval between two selected frames will be larger. But keeping the sail stable for an extended period is usually not a trivial task. Furthermore, Fig. 11c shows that the reconstruction ratio decreases by increasing s. Regarding the reprojected error (Fig. 11b), the behavior was similar to Fig. 10b, i.e., the error slightly increase by
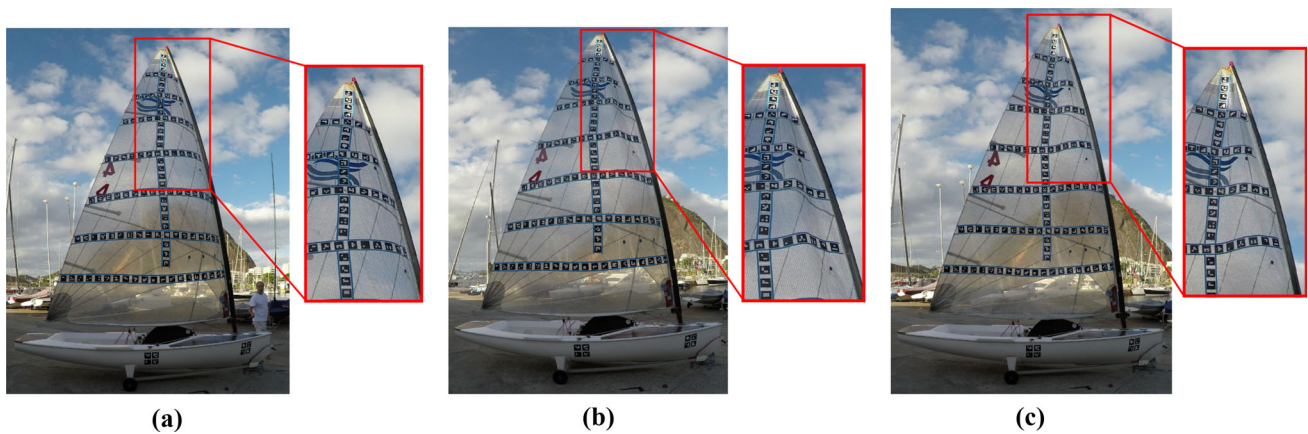
Fig. 8 Sail shape variation due wind changing for clip "near_4k_08.mp4". **a** Frame 60. **b** Frame 210. **c** Frame 330
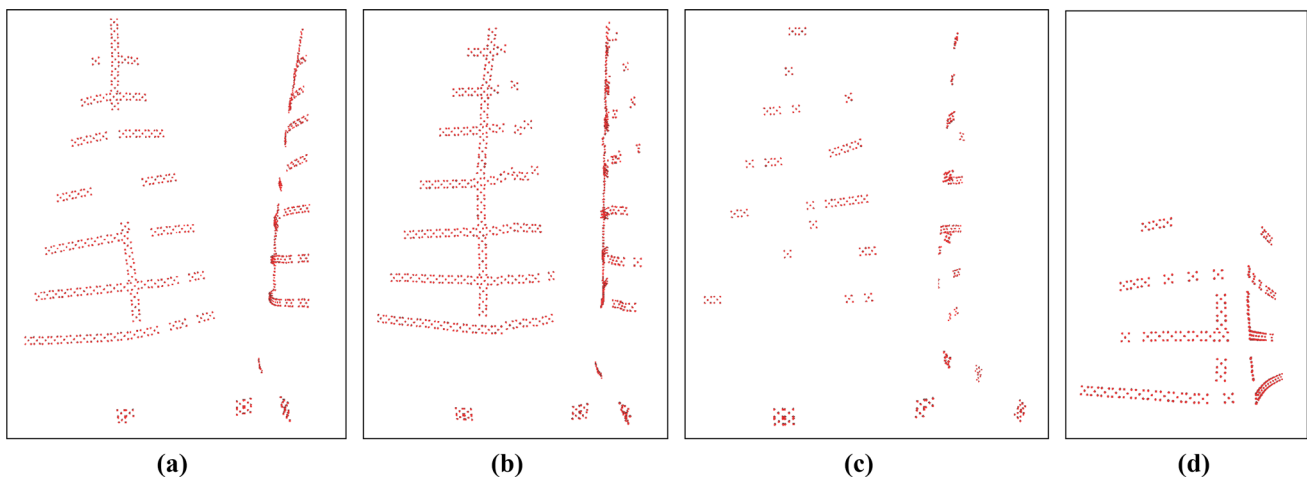

Fig. 9 Visualization of the reconstruction from two view points. **a** Frame 457 for clip "near_4k_17". **b** Frame 219 for clip "near_4k_08". **c** Frame 246 for clip "far_4k_14". **d** Frame 205 for clip "near_4k_07"

increasing s. The explanation is also similar. Since the Bundle Adjustment should adjust frames with more variability between them, it is expected an increase in the mean error to adjust all frames. We observed that $s = 10$ is a good choice for videos recorded at 30 FPS.

We also observed that the neighborhood size m has no significant impact on the errors, but increasing m also increases the reconstruction ratio. This occurs because more frames are used to find inliers to register with the central frame. The 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 s of video at 30 FPS is necessary. However, larger videos allow us to also vary the central frame.

Figure 12a presents the histogram of the markers area by using $n = 20$, $s = 10$ and $m = 5$. This histogram considers the area of the markers reconstructed in all frames. Notice that the markers area tend to be close to the expected value of 10,000 mm^2.

Figures 10c and 11c 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 12b shows the reconstruction ratio for each frame from 202 to 1502 for the clip using $n = 20$, $s = 10$ and $m = 5$. The reconstruction ratio is around 70% before the frame 600, i.e., before 20 s of the video. After this frame, the ratio decreases, only rising again near to the clip end. This behavior is justified by the increase in the capture distance which difficults the markers detection.

Figure 13a shows the reprojection of the reconstructed points (Fig. 9a) on the central frame 457. The points are projected on the expected positions, i.e., at the center of the markers.

Figure 14 presents the rigid motion of the sail markers in relation to the hull markers. This motion is computed by
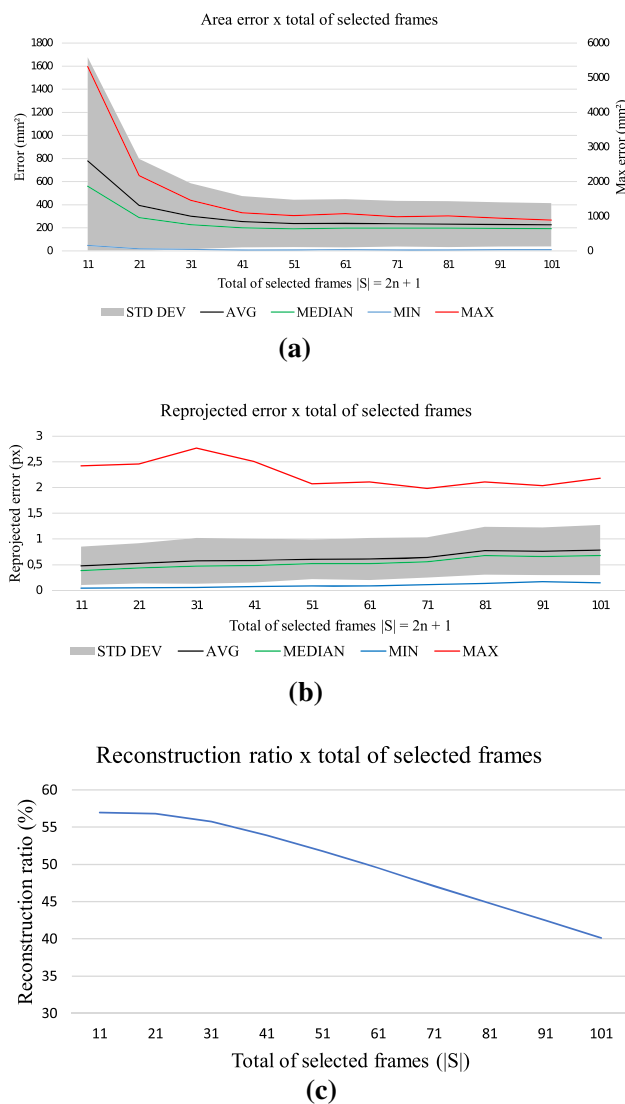
Fig. 10 Evaluators in function of the total number of selected frames. The maximum error scale is at right. **a** Area error. **b** Reprojected error. **c** Reconstruction ratio
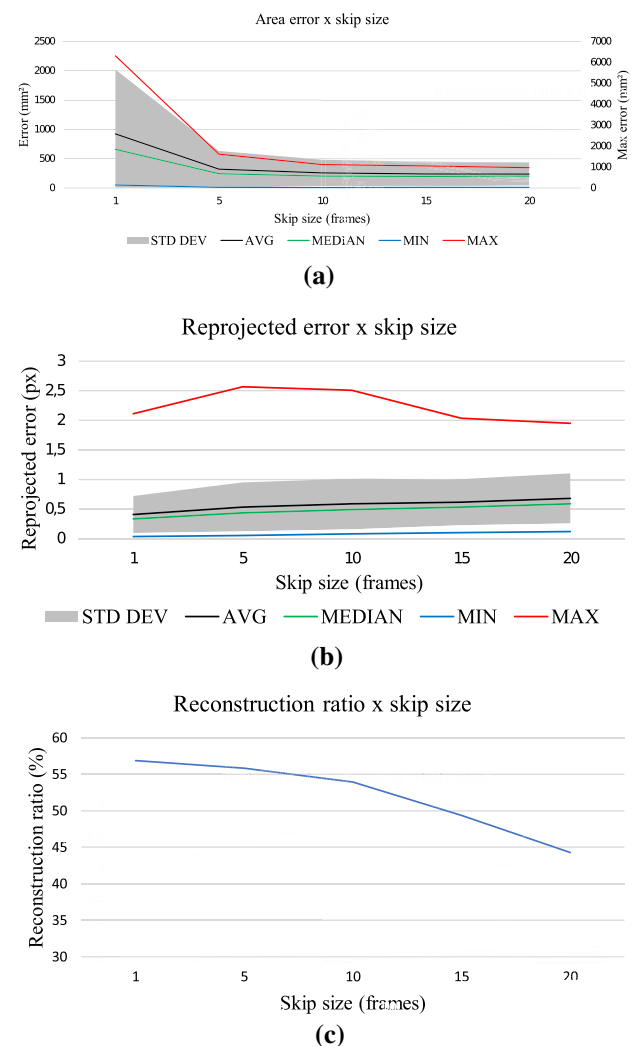

Fig. 11 Evaluators in function of skip size. The maximum error scale is at right. **a** Area error. **b** Reprojected error. **c** Reconstruction ratio

aligning two reconstructions centered in different frames in relation to the hull markers. The distance between reconstructions is 15 frames, i.e., 0.5 second (frames 457 and 472). Notice 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.

4.3.1 Results for videos with wind changes

Our goal is to estimate the mean sail shape during a time interval. Therefore, the sail shape should be as stable as possible during this period. However, the wind changes during some videos, modifying the sail shape (Fig. 8). In this Section, we discuss the results of our method for the clip "near_4k_08.mp4", which presents wind changes. The

reconstruction was performed using the parameters previously chosen ($n = 20$, $s = 10$ and $m = 5$).

Figure 9b presents a visualization of the reconstruction of the frame 219 from two views. We note that the sail region near the luff (right side) is incorrectly reconstructed. This is due a region of the sail that was significantly deformed by the change of wind, as depicted in Fig. 8. Figure 13b shows the reconstructed markers centers reprojected on the frame 219. We observe that the centers are not reprojected in the expected positions where the sail shape changes.

Figure 15a and b presents the comparison between the clips "near_4K_17.mp4" and "near_4k_08.mp4" for the area and reprojected errors, respectively. All errors were greater for the clip "near_4k_08.mp4", confirming 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 mean reconstruction ratio (85%) since the

Fig. 12 **a** Histogram of the markers area. **b** Reconstruction ratio by frame for clip "near_4k_17.mp4"
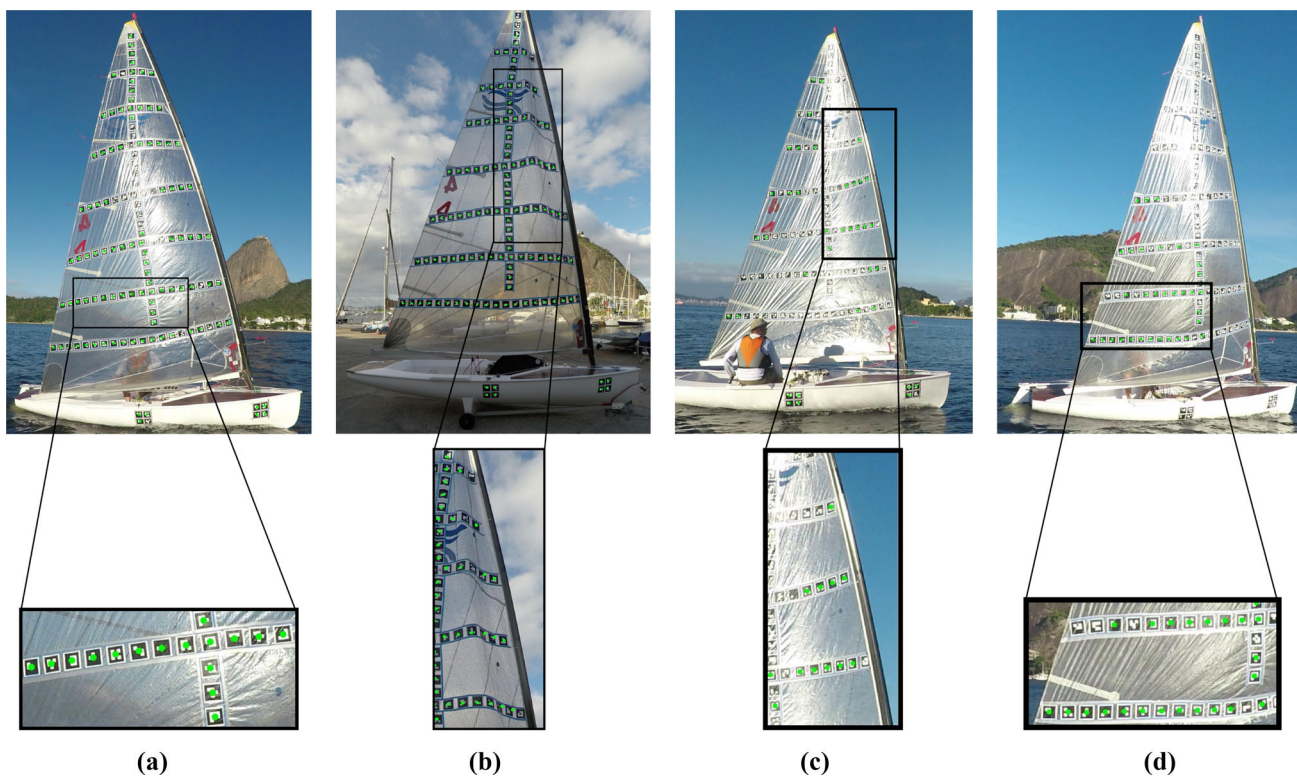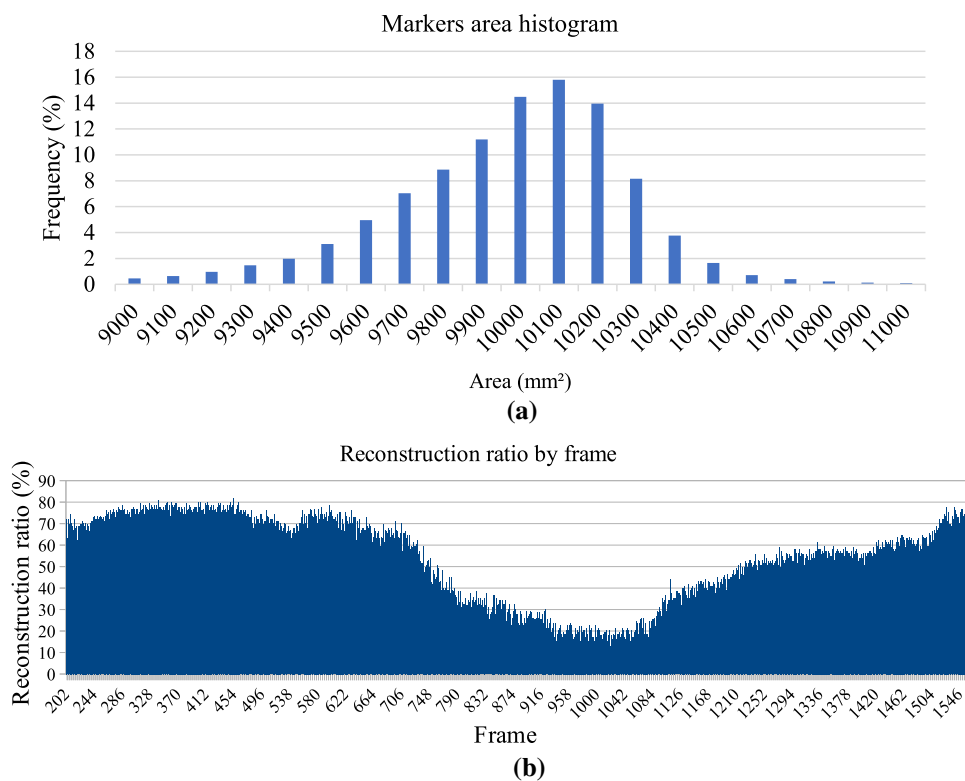

Markers area histogram

(a)

Reconstruction ratio by frame

(b)

(a)   (b)   (c)   (d)

Fig. 13 Reconstructed points reprojected on the central frame. **a** Frame 457 for clip "near_4k_17". **b** Frame 219 for clip "near_4k_08". **c** Frame 246 for clip "far_4k_14". **d** Frame 205 for clip "near_4k_07"
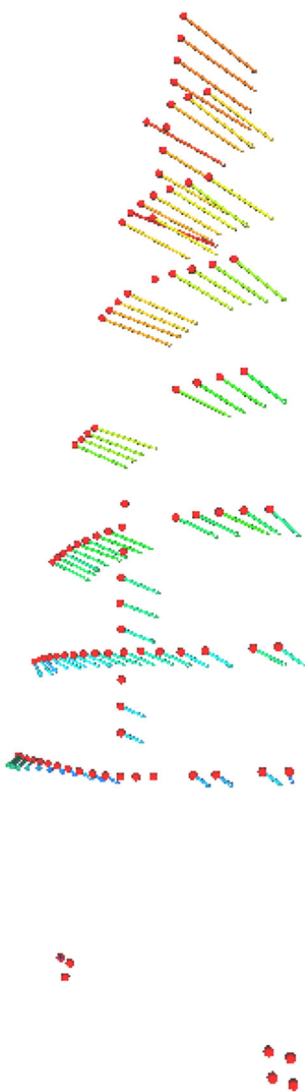
Fig. 14 Sail markers rigid motion in relation to the hull markers. The central frames are separated by 0.5 s (frames 457 and 472). The movement is consistent with the expected behavior, that is, larger motion at the top of the sail

conditions of distance and illumination are favorable. Summarizing, the sail shape stability is essential for the correct working of our method.

4.3.2 Results for videos with strong reflections

Our sailing videos were recorded under natural illumination condition, which are not controllable. As described in Table 2, several videos presented a strong reflection. To illustrate the effect of this issue in our method, Fig. 9c 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 to the reflection.
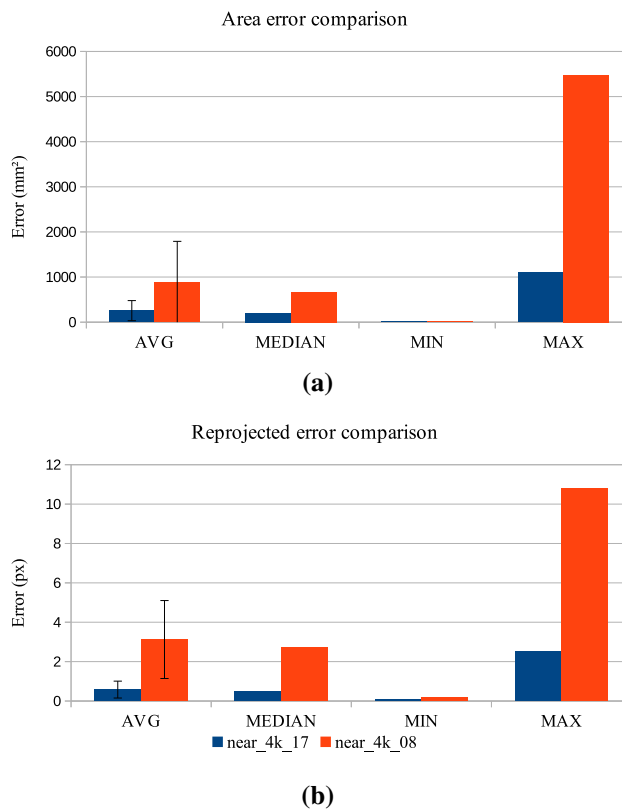
Fig. 15 Error comparison between "near_4k_17.mp4" and "near_4k_08.mp4" clips. **a** Area error. **b** Reprojected error

Figure 13c shows the markers centers reprojected on the central frame. 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, reducing the reconstruction ratio, but it does not affect the quality of the reconstructed markers. Figure 16a and b compares the area and reprojected 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 criteria, the clip "far_4k_14.mp4" presents even better averages.

4.3.3 Capture angle and distance issues

The capture angle is another element that influences the markers detection. Figure 13d shows the reconstructed markers centers of the frame 205 of the clip "near_4k_07.mp4" reprojected on the respective frame. Besides the markers on the top that were obfuscated, the markers at the luff region were not detected due the bad capture angle. The reconstructed points of frame 205 of the clip "near_4k_07.mp4"
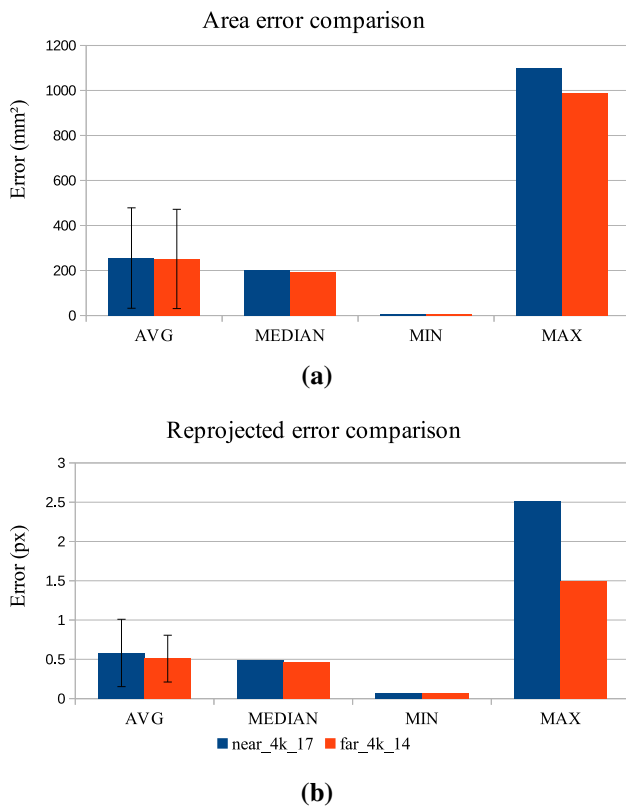
Area error comparison

Reprojected error comparison

Fig. 16 Error comparison between "near_4k_17.mp4" and "far_4k_14.mp4" clips. **a** Area error. **b** Reprojected error

Fig. 17 Markers over a static frame support

Table 3 Reconstruction of the controlled videos (Expected values: distance $= 150$ mm, area $= 6400$ mm^2)

	AVG	STD DEV	STD ERROR
Static curved surface			
Distance	154.03	1.18	0.009
Distance error	4.03	1.18	0.009
Area	6399.63	63.82	0.397
Area error	46.58	43.63	0.272
Curved surface with wind			
Distance	154.31	1.59	0.005
Distance error	4.32	1.58	0.004
Area	6398.91	130.13	0.306
Area error	74.02	107.03	0.252

are presented in Fig. 9d 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 cannot be detected from videos recorded from a great distance. For the clips assigned as "Too far" in Table 2, 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.

4.4 Controlled experiments

To evaluate the precision and accuracy of our method in a controlled environment, we fixed 33 80×80 *mm* markers in a flexible plastic surface. Consecutive markers in a row are separated by 150 *mm* (Fig. 17). The surface was fixed on a slightly cylindrical wood frame. We recorded 48 videos in 4K resolution of this pattern under two situations: static (24 videos) and with wind generated by a fan (24 videos). The camera was slowly moved in all axis, some videos at 2 and some videos at 4 meters from the surface, to represent motion.

We applied our method to reconstruct the surface points using our best parameters for sail reconstruction ($n = 20$, $s = 10$ and $m = 5$). We performed 400 reconstructions
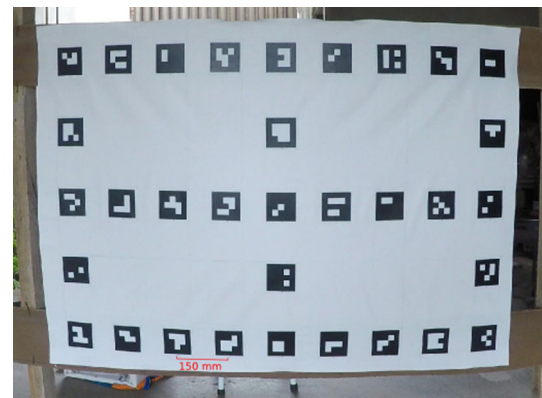
centered in consecutive frames for each video. The distance of horizontally adjacent markers and the markers area were computed for each reconstruction. The statistics of area, distance and respective errors using all 400 reconstructions of the 24 videos in each situation is shown in Table 3.

Table 3 shows that the error for the detected distance between markers was below 3% of the expected value and the area error was around 1% of the expected value. This setup is useful to assess the averaging properties of our method, by using the same parameters tuned for sailing conditions. Notice that the area error average is high if compared to the area average, which is fairly close to 6400 *mm*^{2}. This is due to mistakenly detected marker areas (outliers), leading to a heavy-tailed distribution. However, the standard-deviation can be reduced by filtering out the outliers by thresholding.

Finally, we performed an experiment to evaluate the reconstruction against curvature variation. For this purpose, we used four cylindrical surfaces with different radii. For each one, we used a pattern of 15 markers, varying their dimensions and inter-marker distances to better fit the surfaces. We also adjusted the RANSAC threshold accordingly due to different scales, but all other parameters were fixed. In partic-

Table 4 Information of patterns fixed on cylinders

Cylinder radius	Marker size	Distance between markers	RANSAC threshold
224 mm	60 × 60 mm	80 mm	25 mm
150 mm	40 × 40 mm	50 mm	25 mm
101 mm	30 × 30 mm	40 mm	25 mm
75 mm	20 × 20 mm	25 mm	10 mm
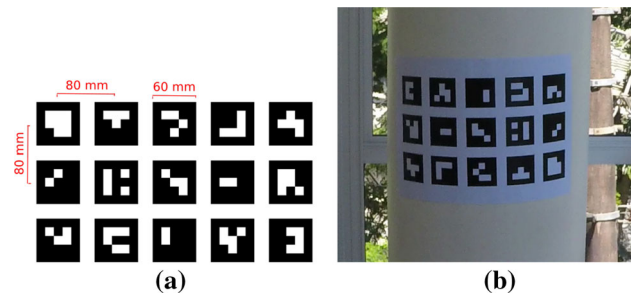
Fig. 18 Cylindrical surface experiment. **a** Planar pattern fixed. **b** Cylindrical surface with the pattern fixed

Table 5 Reconstruction cylindrical surfaces

Surface with radius 224 mm			
Distance error (%)	1.054	0.726	0.004
Area error (%)	1.611	1.323	0.011
Surface with radius 150 mm	AVG	STD DEV	STD ERROR
Distance error (%)	1.320	1.294	0.008
Area error (%)	2.216	2.140	0.018
Surface with radius 101 mm	AVG	STD DEV	STD ERROR
Distance error (%)	1.410	1.386	0.011
Area error (%)	2.857	2.690	0.026
Surface with radius 75 mm	AVG	STD DEV	STD ERROR
Distance error (%)	1.840	1.186	0.004
Area error (%)	2.090	1.800	0.015

ular, we used our best parameters for the sail reconstruction ($n = 20$, $s = 10$ and $m = 5$). Table 4 shows the settings for each surface, and Fig. 18 illustrates the settings for the surface with largest radius.

For every surface, we recorded a video in 4K resolution by moving the camera along all axes at a distance of approximately 2 meters from the surface. We then performed 1000 reconstructions centered in consecutive frames. Since the cylinder radius and the geodesic distances between markers are known, we calculated the real euclidean distance between markers and compared them to the reconstructed data. We also calculated the real planar area formed by the markers' corners and compared to the estimated markers. The results are presented in Table 5. The average distance error was less than 2% for all cases, and all area errors were below 3%, which is compatible, and predominantly better, than the approximately 2.5% for the sail reconstruction. Figure 19 shows the reconstructed points around the ground truth cylinders.

These experiments show the ability of our method to reconstruct surfaces with different curvatures without affecting the performance. It is important to mention that the distance verification takes into account distances between non-adjacent markers, where the difference between the geodesic and euclidean distances are larger. We even included the distance between markers at opposite extremities of each row. Moreover, all cylinders provide a curvature much larger than any expected configuration of the sail, better supporting our results for the sail reconstruction.

4.5 Runtime discussion

As described in Sect. 3, our reconstruction method is composed of five steps: markers fixation, capture, detection, registration, and reconstruction. Each step takes a different time to be performed and depends on different factors. As mentioned in Sect. 3.1, markers fixation takes about two hours, even though we expect this time to reduce significantly with more practice. The video capture depends on how long we want to analyze the sail behavior. However, as exposed in Sect. 4.3, less than 30 s of footage is already enough to achieve a suitable reconstruction.

The detection time depends on the video resolution and how many frames are analyzed for the reconstruction. For a reconstruction using our best parameters ($n = 20$, $s = 10$ and $m = 5$), we examine at most 601 frames. In the worst case, the distance between the reference frame and the first and last selected frames will be of 300 frames, since we have in this case 20 frames separated by 15 frames (10 of the skip size and 5 of the neighborhood). Notwithstanding, the detection is performed before the frame selection step, thus, we need to detect markers in all 601 frames. The detection of 601 frames for a 4K video takes around 221 s in a Intel®
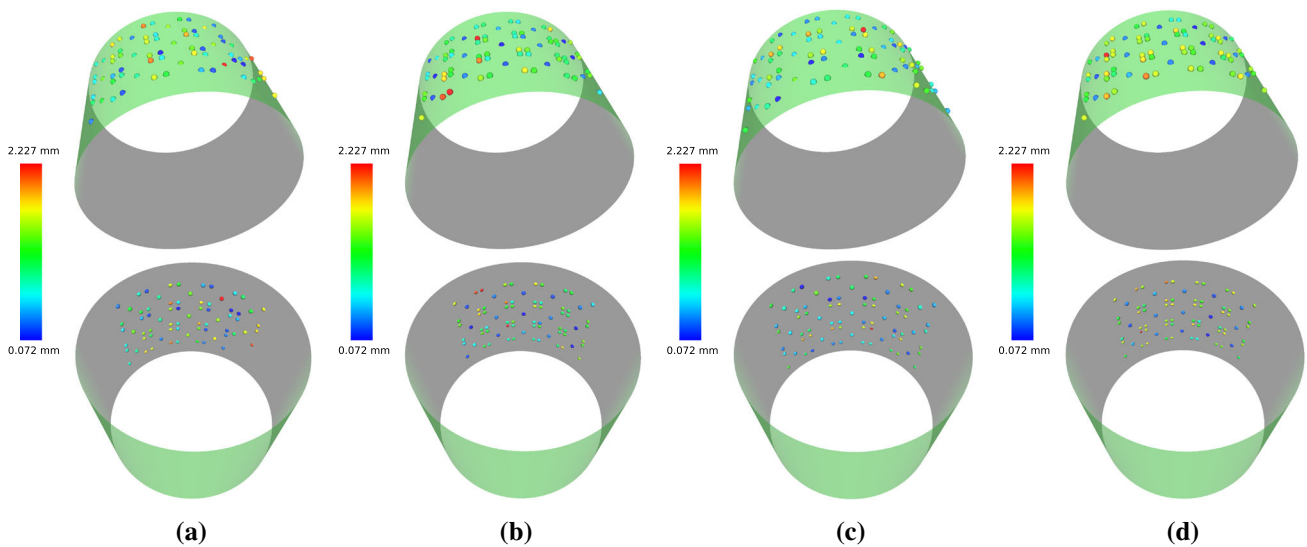
Fig. 19 Visualization of the error between the reconstructed points and the real cylindrical surface. **a** Cylinder with radius 224 *mm* (average error: 0.990 *mm*). **b** Cylinder with radius 150 *mm* (average error: 0.966 *mm*). **c** Cylinder with radius 101 *mm* (average error: 0.652 *mm*). **d** Cylinder of radius 75 *mm* (average error: 0.688 *mm*)

Core™ i7-5500U 2.40GHz processor with 8GB of memory. It is important to note that the detection needs to be performed only once for each video, as it can then be reused to compute reconstructions centered at different frames and using different parameters.

The registration and reconstruction steps depend on the number of markers. For the reconstruction of frame 457 of video "near_4k_17.mp4" (Fig. 9a), the registration and reconstruction took 21 and 41 s, respectively. This reconstruction is composed of 92 markers (460 points). The total processing time was 283 s, considering detection, registration and reconstruction.

We performed some extra tests by artificially removing some markers. In this case, markers with odd indexes were discarded. We analyzed the quality and runtime of this sparser reconstruction. As expected, the area and reprojected error were comparable to the complete reconstruction for video "near_4k_17" (Fig. 20), as presented in Sects. 4.3.1, 4.3.3 and 4.3.2 for issue cases. This means that a sparse reconstruction can present a satisfactory result and it is possible to reduce the number of markers according to the application needs. In terms of runtime, the sparser reconstruction took 10 s for the registration and 25 s for the reconstruction. However, the detection step time did not present a significant reduction since the ArUco library still runs through the entire image to detect markers, i.e., it depends on the image resolution and not on the number of markers. Thus, using fewer markers obviously reduces the fixation time, but does not improve significantly the processing time and does not hinder the reconstruction.
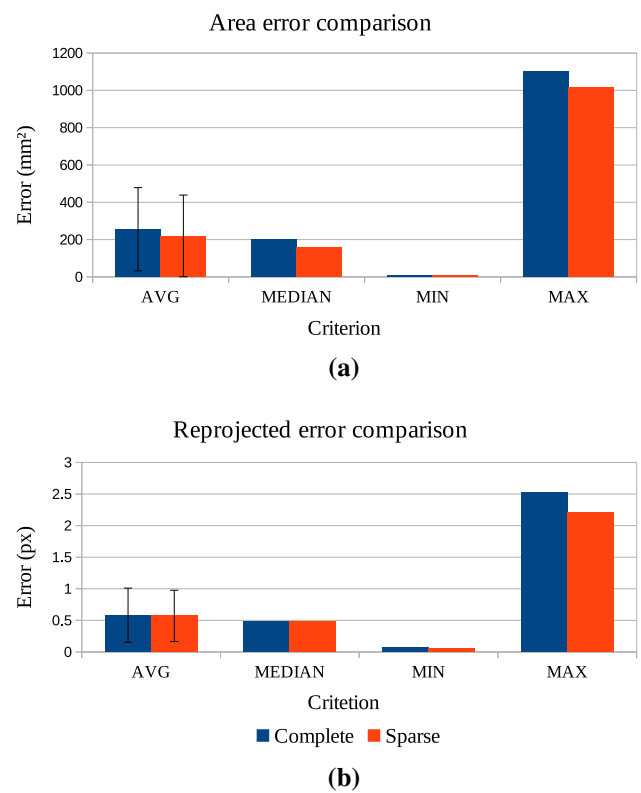

Fig. 20 Error comparison between complete and sparse reconstructions achieved by artificially removing markers for the "near_4k_17.mp4" clip. **a** Area error. **b** Reprojected error
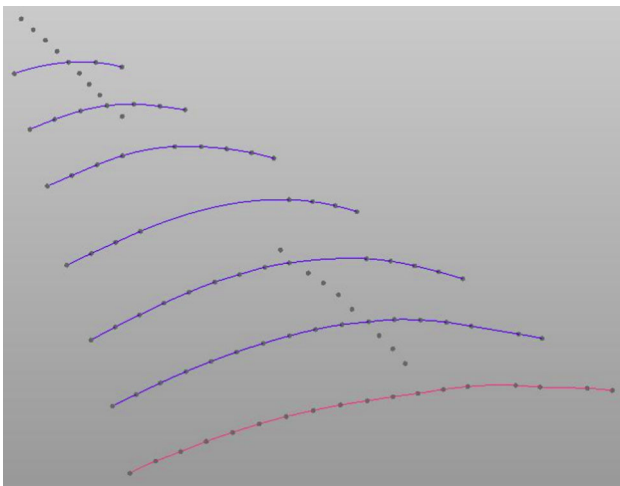
Fig. 21 Profile of the sail sections. The lower curve in red presented some distortion, but we were unable to precise its source

4.6 Qualitative discussion

The results for the reconstruction of the clip "near_4k_17.mp4" were submitted to domain experts and experienced sailors for analysis. Figure 21 shows the profiles of the sail sections generated by naval engineers using the ANSYS [25] software from the markers centers of our reconstruction data. They observed that, in general, the shape of the profiles of the sail sections is 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 misalignment between the profiles is observed. The same observation was formulated about the initial and final points of the profiles. We noticed that these misalignments result from the actual markers positioning on the sail. Therefore, the per points reconstruction quality was considered satisfactory to generate the sail shape. Nevertheless, it was suggested that additional information about the sail bounds would entail more useful reconstructions for simulation and design evaluation purposes, and a more careful positioning of the markers would also increase the profile reconstruction quality.

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 noninvasive, even though we still have to stick the markers onto the sail we do not interfere with the sailing. For sail design and analysis purposes, it is important to achieve the sail mean shape during a time interval, while the boat is as stable as possible. Our main premise 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 to setup 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 naval architects to 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.

To validate 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. We believe that the creation of such dataset may be valuable 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 px. Qualitatively, we notice 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. Limitations of our method include the reflection, the capture distance and view 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, 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 was 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. Filters attached to the camera can be useful to deal with the reflection issue. It is possible to capture a large sail or more than one sail by simultaneously using two or more auxiliary boats. 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. The Bundle Adjustment step can be tuned by using known specific constraints of our problem. In addition, another criterion to search optimized frames for the reconstruction can be evaluated, selecting frames based in their reconstruction quality instead of quantity of markers.

References

1. Ambroziak, A., Kłosowski, P.: Polyester sail technical woven fabric behaviour under uniaxial and biaxial tensile tests. J. Theor. Appl. Mech. **56**(1), 227–238 (2018)
2. Augier, B., Bot, P., Hauville, F., Durand, M.: Experimental validation of unsteady models for wind/sails/rigging fluid structure interaction. In: International Conference on Innovation in High Performance Sailing Yachts, Lorient, France (2010)
3. Bartoli, A., Gérard, Y., Chadebecq, F., Collins, T.: On template-based reconstruction from a single view: Analytical solutions and proofs of well-posedness for developable, isometric and conformal surfaces. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 2026–2033. IEEE (2012)
4. Battistin, D., Ledri, M.: A tool for time-dependent performance prediction and optimization of sailing yachts. In: Proceedings of the 18th Chesapeake Sailing Yacht Symposium, Annapolis, MD, pp. 90–101 (2007)
5. Blanz, V., Scherbaum, K., Seidel, H.P.: Fitting a morphable model to 3d scans of faces. In: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, pp. 1–8. IEEE (2007)
6. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond euclidean data. arXiv preprint arXiv:1611.08097 (2016)
7. Brunet, F., Hartley, R., Bartoli, A., Navab, N., Malgouyres, R.: Monocular template-based reconstruction of smooth and inextensible surfaces. In: Computer Vision–ACCV 2010, pp. 52–66. Springer (2010)
8. Clauss, G., Heisen, W.: Cfd analysis on the flying shape of modern yacht sails. In: Maritime Transportation and Exploitation of Ocean and Coastal Resources: Proceedings of the 11th International Congress of the International Maritime Association of the Mediterranean, Lisbon, Portugal, p. 87 (2006)
9. Cohen-Or, D., Greif, C., Ju, T., Mitra, N.J., Shamir, A., Sorkine-Hornung, O., Zhang, H.R.: A Sampler of Useful Computational Tools for Applied Geometry, Computer Graphics, and Image Processing: Foundations for Computer Graphics, Vision, and Image Processing. CRC Press, Boca Raton (2015)
10. Deparday, J., Bot, P., Hauville, F., Augier, B., Rabaud, M.: Full-scale flying shape measurement of offwind yacht sails with photogrammetry. Ocean Eng. **127**, 135–143 (2016)
11. Ferreira, P., Caetano, E., Pinto, P.: Real-time flying shape detection of yacht sails based on strain measurements. Ocean Eng. **131**, 48–56 (2017)
12. Fossati, F., Mainetti, G., Malandra, M., Sala, R., Schito, P., Vandone, A.: Offwind sail flying shapes detection. In: Proceedings of the 5th High Performance Yacht Design Conference. Auckland (2015)
13. Garrido-Jurado, S., Noz Salinas, R.M., Madrid-Cuevas, F., Marín-Jiménez, M.: Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recognit. **47**(6), 2280–2292 (2014). https://doi.org/10.1016/j.patcog.2014.01.005
14. Graf, K., Müller, O.: Photogrammetric investigation of the flying shape of spinnakers in a twisted flow wind tunnel. In: Proceedings 19th Chesapeake Sailing Yacht Symposium (2009)
15. Hayashi, T., De Sorbier, F., Saito, H.: Texture overlay onto non-rigid surface using commodity depth camera. In: VISAPP (2), pp. 66–71. Citeseer (2012)
16. Hilsmann, A., Eisert, P.: Tracking and retexturing cloth for real-time virtual clothing applications. In: Computer Vision/Computer Graphics CollaborationTechniques, pp. 94–105. Springer (2009)
17. Jojic, N., Huang, T.S.: Estimating cloth draping parameters from range data. In: In International Workshop on Synthetic-Natural Hybrid Coding and 3-D Imaging, pp. 73–76 (1997)
18. Kato, H., Billinghurst, M.: Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality, 1999. (IWAR'99) , pp. 85–94. IEEE (1999)
19. Kümmerle, R., Grisetti, G., Strasdat, H., Konolige, K., Burgard, W.: g 2 o: A general framework for graph optimization. In: 2011 IEEE International Conference on Robotics and Automation (ICRA), pp. 3607–3613. IEEE (2011)
20. Le Maître, O., Huberson, S., De Cursi, J.S.: Application of a non-convex model of fabric deformations to sail cut analysis. J. Wind Eng. Ind. Aerodyn. **63**(1–3), 77–93 (1996)
21. Le Pelley, D., Modral, O.: V-spars: A combined sail and rig shape recognition system using imaging techniques. In: Proc. 3rd High Performance Yacht Design Conference Auckland, New Zealand, Dec, pp. 2–4 (2008)
22. Liu, Y., Chen, Y.Q.: Joint reconstruction of 3d shape and non-rigid motion in a region-growing framework. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1578–1585. IEEE (2011)
23. Liu, Y., Chen, Y.Q.: 3d tracking of deformable surface by propagating feature correspondences. In: 2012 21st International Conference on Pattern Recognition (ICPR), pp. 2202–2205. IEEE (2012)
24. Mausolf, J., Deparday, J., Graf, K., Renzsch, H., Böhm, C.: Photogrammetry based flying shape investigation of downwind sails in the wind tunnel and at full scale on a sailing yacht. In: Proceedings of the 20th Cheasapeake Sailing Yacht Symposium. Annapolis, pp. 33–43 (2011)
25. Mechanical, A.R.: Ansys. https://www.ansys.com/. Release 18.1
26. Newcombe, R.A., Fox, D., Seitz, S.M.: Dynamicfusion: reconstruction and tracking of non-rigid scenes in real-time. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 343–352 (2015)
27. Ngo, D.T., Östlund, J., Fua, P.: Template-based monocular 3d shape recovery using laplacian meshes. IEEE Trans. Pattern Anal. Mach. Intell. **38**(1), 172–187 (2016)
28. Perriollat, M., Hartley, R., Bartoli, A.: Monocular template-based reconstruction of inextensible surfaces. Int. J. Comput. Vis. **95**(2), 124–137 (2011)

29. Pilet, J., Lepetit, V., Fua, P.: Fast non-rigid surface detection, registration and realistic augmentation. Int. J. Comput. Vis. **76**(2), 109–122 (2008)
30. Pritchard, D., Heidrich, W.: Cloth motion capture. Comput. Graph. Forum **22**(3), 263–271 (2003)
31. Renzsch, H., Graf, K.: An experimental validation case for fluid-structure-interaction simulations of downwind sails. In: 21st Chesapeake Sailing Yacht Symp (2013)
32. Rousselon, N.: Optimization for sail design. In: ModeFrontier Conference (2008)
33. Roux, Y., Huberson, S., Hauville, F., Boin, J.P., Guilbaud, M., Ba, M.: Yacht performance prediction: towards a numerical vpp. In: High Performance Yacht Design Conference, Auckland, pp. 11–20 (2002)
34. Salzmann, M., Fua, P.: Linear local models for monocular reconstruction of deformable surfaces. IEEE Trans. Pattern Anal. Mach. Intell. **33**(5), 931–944 (2011)
35. Salzmann, M., Pilet, J., Ilic, S., Fua, P.: Surface deformation models for nonrigid 3d shape recovery. IEEE Trans. Pattern Anal. Mach. Intell. **29**(8), 1481–1487 (2007)
36. Shimizu, N., Yoshida, T., Hayashi, T., De Sorbier, F., Saito, H.: Non-rigid surface tracking for virtual fitting system. In: VISAPP (2), pp. 12–18. Citeseer (2013)
37. Triggs, B., McLauchlan, P.F., Hartley, R.I., Fitzgibbon, A.W.: Bundle adjustment—a modern synthesis. In: International workshop on vision algorithms, pp. 298–372. Springer (1999)
38. Yu, R., Russell, C., Campbell, N.D., Agapito, L.: Direct, dense, and deformable: Template-based non-rigid 3d reconstruction from rgb video. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 918–926. IEEE (2015)
39. Zollhöfer, M., Nießner, M., Izadi, S., Rehmann, C., Zach, C., Fisher, M., Wu, C., Fitzgibbon, A., Loop, C., Theobalt, C., et al.: Real-time non-rigid reconstruction using an rgb-d camera. ACM Trans. Graph. (TOG) **33**(4), 156 (2014)

Luiz Maciel is Ph.D. in Systems Engineering and Computer Science at Federal University of Rio de Janeiro (UFRJ - 2018). He holds a M.Sc. degree in Computer Science from the Federal University of Juiz de Fora (UFJF - 2014) and graduated in Computer Science at the same university (2011). He is a professor at UFJF since 2019. His main knowledge area is Computer Science focusing on Image Processing and Computer Vision.

Ricardo Marroquim is an Assistant Professor at Delft University of Technology, The Netherlands. He has a M.Sc. (2001) and a PhD (2008) in Computer Science from the Federal University of Rio de Janeiro. His main interests is on Appearance Capture and Modelling and Rendering algorithms.

Marcelo Vieira is Ph.D. in Image & Signal Processing (2002) from ENSEA/UCP in France and Computer Science (2002) from UFMG. He has M.Sc. in Computer Science (1998) from UFMG and graduated in Computer Science in PUC-MG (1995). He is associate professor at UFJF with interests in computer vision, computer geometry and fluid simulation.
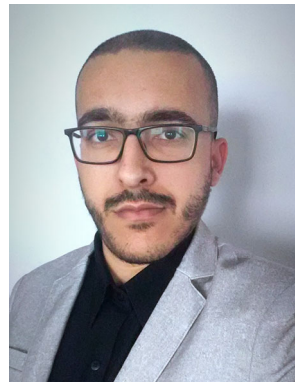
Kevyn Ribeiro is a master's student of Computer Science at the Federal University of Juiz de Fora (UFJF). He is a member of the Group for Computer Graphics, Image and Vision (GCG) at UFJF, where he has interests on research and development of computer vision and computer graphics.

Alexandre Alho is an Associate Professor at the Department of Naval and Ocean Engineering of the Federal University of Rio de Janeiro (UFRJ). He has M.Sc. and D.Sc. in Ocean Engineering from COPPE/UFRJ. His professional expertise is focused on the design of sailing and motor yachts, with an emphasis on the design of hull lines and propulsion systems. He is the Head of the Nautical Centre Lab of the Federal University of Rio de Janeiro.