AMBRA VANDONE

MEASUREMENT AND ANALYSIS OF FREE-FORM OBJECTS DEVELOPMENT OF A SOLUTION FOR FLYING SAIL SHAPE RECONSTRUCTION

MEASUREMENT AND ANALYSIS OF FREE-FORM OBJECTS DEVELOPMENT OF A SOLUTION FOR FLYING SAIL SHAPE RECONSTRUCTION

Doctoral dissertation of

AMBRA VANDONE

Politecnico di Milano
Department of Mechanical Engineering
Doctoral Program in Mechanical Engineering

Supervisor: Prof. Remo Sala
Tutor: Prof. Fabio Fossati
Co-Tutor: Prof. Marco Bocciolone
Co-Tutor: Michele Malandra (North Sails Group)

March 2016

To my tutor, Professor Fabio Fossati.

ABSTRACT

In this research some novel methods for the acquisition and the analysis of free-form shapes have been proposed.
In particular for this study, the flying sail shape has been consider as free-form object to be properly measured and geometrically analyzed. Large differences, in fact, exist between a computer based design shape and the resulting flying shape due to different factors such as pressure distribution, sail trim controls and fluid structure interaction forces. Nowadays, Computational Fluid Dynamics (CFD) codes assess the yacht performances starting from the design shape; however, the actual flying shape is the only one truly related to moments, heeling and thrust forces. Hence the reason for this case study. Moreover, analyzing the reconstructed shape in terms of geometrical features can be of interest for sail makers to redesign the sail and for crew members to adjust the trim.
The main goal of this work was the development of a methodology for data acquisition and elaboration. A Time-Of-Flight based device has been realized ensuring non-contact, wide range and outdoor measurement. A laser pulse is emitted, and measuring the time it takes to hit the target and return to the sensor, distance estimation can be computed. The laser beam is properly deflected to be able to scan a three-dimensional object. Its metrological qualification has been conducted, including tests to assess the in influence of the target-to-sensor distance, incident angle, target material and lighting conditions.
Furthermore, most efforts have been addressed to the development of a custom post process algorithm. Raw data are acquired in terms of point cloud and several steps are required to lead to the sail surface reconstruction. Those are primarily: registration of the different scans into a common reference system, scene interpretation - i.e. segmentation of the cloud to extract the sail cluster -, filtering of the data to remove outliers and to reduce acquisition noise, and modeling - i.e. the creation of a surface -.
The procedure was validated onto synthetic data sets representing simple scenes and onto design sail shapes provided by sail-makers. Finally, the algorithm was exploited to reconstruct sails during wind tunnel campaigns and even for few tests on field leading to promising results.

CONTENTS

1 INTRODUCTION

RESEARCH CONTENTS

There is no doubt that, nowadays, competitiveness in all branches of the manufacturing industry is extremely high. It follows the need of implementing processes that reduce the time-to-market and optimize the product performances, tailoring them to particular customer requirements.
In many industrial applications, functional product characteristics depend tightly on the product's shape. Shapes that have to be handled are often free-form and cannot be described through traditional regular features or as a composition of well-known geometry solids.
The problem is twofold: it may be necessary either to design free-form shapes or to recover information from a free-form real object. This work focuses mainly on the second aspect. The idea, in fact, is the implementation of a procedure able to estimate the product performances through the analysis of its shape. So different object shapes could be tested in order to achieve the most performing item.

In particular, this study integrates into the aero- and hydrodynamics field. It aims at providing a reliable evaluation of a sailing yacht performances related to its hull and sail shapes.
While Computational Fluid Dynamic (CFD) analysis has become routine in hull design for the investigation of hydrodynamic forces, its application in predicting the aerodynamics of a yacht is not yet widely exploited, since modeling what happens above the waterline is always a complex work. The aerodynamic pressure distribution acting on the surface of the sail, the forces resulting from rig design and trim, and the elasticity of the sail material influence its actual flying shape in varying wind conditions. The combination of all these factors leads to a virtually infinite number of flying shapes and, this is especially true considering off-wind sails because of the lightweight construction materials and their relatively unconstrained nature. On the other hand, the geometry of the sails determine the aerodynamic flow around them and the resulting forces.
Thus, it is of fundamental importance for the reliability of sail force predictions using CFD to know the real geometry of the sail for given wind conditions. It is possible, although computationally expensive, to model the aero-structural coupling by combining CFD codes with Finite Element Analysis (FEA) of the sail.
Up to date numerical codes still need a massive validation work

thanks to both wind tunnel tests as well as on water testing at full scale. Direct data collection can give a meaningful contribution, supposing that reliable flying shapes are provided, reflective of the realistic sailing and trim conditions.

In the past years, several methods have been developed by the scientific community for a better understanding of the flying sail behavior.
Preliminary studies have been made on scaled yacht models using photogrammetry based techniques or Coordinate Measuring Machine (CMM). As a general comment, it can be noted that photogrammetry was judged accurate and relatively fast during the tunnel occupancy phase. Its main disadvantages are that it requires an intensive data post-processing and that it suffers from occlusions. To overcome the problem, a large number of cameras is required, leading to relevant difficulties in the system set up. On the other hand, CMM process requires a longer period during the tunnel occupancy phase, but yields results within moments after the digitizing process is complete. Its principal disadvantage is that the CMM and its operator can potentially influence the shape of the sail.
Full scale tests have been carried out lately using numerous digital cameras to record the position of a marker grid or the shape assumed by some colored stripes placed onto the sail surface at certain heights. The same technique is used by some computer programs, which are nowadays available on the market, aiming at analyzing sail pictures and providing some sail shape parameters in a certain number of sail sections (for example: North Sails proprietary software Advanced Sail Analyzer (ASA) or Visual Sail Position and Rig Shape (V-SPARS)). Their main drawback is that these systems can recover information just for few sail sections, assuming, moreover, that these colored stripes remain in a horizontal plane, which is questionable for downwind sails.

Considering the above, an innovative tool for the sail shape acquisition based on Time Of Flight (TOF) technology has been proposed in this thesis. It allows for performing three-dimensional (3D) measurement of the entire sail shape, overcoming most of the limits of the above mentioned techniques. Moreover, it is suitable for both upwind and downwind sails and for both indoor and outdoor environment, as it is barely sensitive to the environmental light conditions.
In particular, the acquisition device is equipped with a laser scanner, a gear-motor unit and other auxiliary sensors. The scanner emits a laser pulse in a certain direction, and estimates the target distance by evaluating the time the pulse takes to return to the sensor. An internal mirror deflects the pulse allowing the acquisition of sample points lying in a 2D (planar) section of the scene. The motor unity

allows for performing the controlled rotation of the measurement device around an axis perpendicular to the mirror rotational axis, ensuring, therefore, the acquisition of wide 3D scenes. A proximitor is used to identify the initial scanning position for each scan.
An in house software has been developed allowing for the handling unit control strategy. Through the realized user interface it is possible to: set velocity and acceleration for motor rotation, acquire data from the TOF sensor, set the scanning parameters, such as starting angle and angle to be scanned, receive an external trigger to automatically run the scanning (useful to synchronize the acquisition of different devices), and visualize the scene acquired by means of a webcam placed onto the sensor.
This acquisition unit was subject to metrological qualification to verify the measurement uncertainty and the quality of the entire scan. Various parameters that could affect the sensor performances, such as warm-up time, target properties (color and material), and target position (distance and orientation) are investigated. Moreover, since the sensor has been designed to work in a wide outdoor environment, the effect of direct sun light was taken into account. The order of magnitude for the measurement uncertainty obtained was around one centimeter. Considering the nautical application, the sensor was judged adequate.
To verify the quality of the geometrical feature of the data acquired, simple geometries and dimensional known objects were scanned and we compared the acquisition to the reference dimensions. Differences were in the order of magnitude of few millimeters, leading to satisfying conclusions.

Acquired data comes in the form of dense point clouds representing the spatial positions of thousands of object surface sampled, and realized properly piling several planar scans. The idea, in fact, is to consider the flying sail as a free-form object since its shape can be barely described using primitive geometries or by a combination of them. Thus, several points have to be measured to correctly reconstruct the sail shape. A dedicated software for the elaboration of the acquired point clouds has been realized.
Dealing with a custom-made device, common point cloud algorithms presented in literature had to be modified to accept as input non-organized and non-uniformly dense point clouds. The first step of the point cloud elaboration pipeline is about the registration of different clouds into a common reference coordinate system. In fact, a single scanner unit is required for each sail one intends to measure; thus, to recover the entire sail plan and to match it with the hull and rigging Computer Aided Design (CAD) model, the clouds have to be properly moved and oriented. A common practice is to let the user select corresponding points in different clouds and compute the

transformation matrix based on these point coordinates.
For this work, instead, the knowledge of some scene geometrical features is exploited. The algorithm extracts three planar surfaces from each cloud (for example: ceiling, floor and wall of the wind tunnel chamber) and computes the spatial pose of each plane. The cloud to register is rotated and moved so that its detected surfaces overlap the reference cloud ones. This strategy allows for reducing the user interaction and to limit at most the influence of the measurement uncertainty since it computes the transformation matrix based on the orientation of the planes that best fit thousands of points.
Once the clouds are oriented, segmentation is performed to isolate the sail cluster to analyze. Point spatial proximity and normal orientation of local fitting surfaces are the principal criteria to identify points belonging to different objects in the scene. Denoising, statistical outlier removal and point cloud smoothing are performed setting the required inputs according to what was carried out after many tries: the goal is to allow for spatial relocation of each sample according to the characteristic of its neighborhood to compensate the acquisition noise but being very careful not to alter the overall acquired shape. Starting from this clean sail cluster, a triangular mesh is realized to link all the points.
Then, the algorithm detects automatically the sail vertexes and edges (thanks to some heuristics and/or to the knowledge of the design shape) and divides leech and luff in as many segments as the user desires. Planes that passes through two corresponding luff and leech segment extremities and perpendicular to the meshed surface at the points of maximum camber are identified (iterative process). These planes are used to section sail surface and the resulting curves are analyzed in terms of geometrical parameters such as camber, draft or twist, which can be used for a quick evaluation of the sail shape. They provide a synthetic idea about the way the sail is flying, allowing a yacht crew to eventually adjust the trim or sail-makers to conceive a redesign of the shape.
Finally, the surface modeling problem is faced. A linear piece-wise approximation of the surfaces can be obtained by connecting sample points to form a polygonal mesh. Because of the non-uniform cloud density, this approach sometimes lead to holes or non-realistic piece of surface. Looking for a smoother and continuous surface, different algorithm based onto NURBS geometry have been tested. The results look nicer but the surface might depart considerably from the sample points, especially at the edges.
Thus, some 3D pattern matching algorithms, usually exploited in the industrial field, have been considered. They require a CAD model for the object to look for and they extract from it particular features such as sharp variation in surface curvature or identify geometrical primitives. Then, they try to fit these features onto the points in the

acquired cloud, and if a good correspondence is found, they superimpose the CAD model to the matched cluster of points. Thus, for this application the idea is to look for the sail CAD model into the scanned scene, but difficulties came out since no evident features could be detected in the sail shape - it is just a smooth curvature variation surface - and deformation of the CAD model has to be allowed to reproduce the flying shape. This approach needs to be further investigated.

The algorithm has been validated using a CAD model given by the sail-maker as an input and comparing both geometrical parameters computed and sail surface reconstructed to the design ones. Differences could be considered negligible leading to assess good performances of post processing tools.
The device has been used in various wind tunnel tests on scaled model and in few full scale tests. For example, a first campaign has been conducted on both symmetric and asymmetric spinnakers for an Offshore Racing Congress project aimed at revising off-wind sails aerodynamic coefficients and ORC VPP aerodynamic model. Another experimental research has been realized to support the sail inventory development for a high-performance super-yacht and in particular aiming to evaluate the opportunity to install a sail designed for furling. Comparison between design shapes and acquired flying shapes and potential improvements to the off-wind sail design process by developing an off-wind reference sail design database have been faced.
Moreover, full-scale tests have been performed onto the Laboratory boat realized by Politecnico di Milano staff. Upwind and downwind sail shapes have been correlated to values acquired by six load cells and other installed devices. Yacht performances have been assessed and reliable CFD analysis could have been performed starting from the reconstructed shapes. Since results obtained seemed promising, the tool presented in this research has been patented[1]and the possibility of a product industrialization has been considered.

RESEARCH AIMS AND STRUCTURE

The aim of this research is to provide methods and tools to measure and analyze free-form shapes.
Particular attention is addressed to the flying shapes assumed by the sails during navigation that are quite difficult to be handled, because of their high deformability and variability in time.
However, the work maintains a wide generality, as the flying sail shape assessment presents all the typical major problems of free-form

1 *Time of Flight (TOF) System for Sail Flying Shape Detection*
2014 - Italian Patent application n° PD2014A000249 (accepted)
2015 - International extension n° n. PCT/EP2015/072278 (pending)

shape analysis.

First of all, a definition of free-form shape is provided in Chapter 2, followed by a discussion above the most suitable measurement techniques to acquire this kind of shape and by a brief presentation of the principal methods to mathematically model a free-form object.
In Chapter 3, the reasons for the interest of the nautical scientific community in studying the flying sail shapes are reported. Different contributions on this topic presented in literature are described and focus is as much addressed to the the goodness of the results obtained as to the weak aspects.
Then, concept for a new tool able to acquire and elaborate the data regarding the flying sail shape is introduced justifying the selection of the TOF technology and the choice of looking at the sail as a free-form object. Advantages and disadvantages of the proposed idea are anticipated and discussed in detail in the following chapters.
In Chapter 4 the hardware acquisition system is described along with the dedicated software that enable the user to control the scanning. Metrological qualification of the acquisition unit follows. Then, in Chapter 5 the data elaboration pipeline is presented. A dedicated software receives as input the point cloud acquired and ends providing synthetic geometrical parameters and a 3D reconstruction of the sail surface. Each step of the data elaboration process is faced starting from standard algorithms and modifying them to suit the application requirements.
In Chapter 6, the experimental activity is presented regarding wind tunnel tests onto different upwind and downwind sails and full-scale tests done onto the Laboratory Boat of Politecnico di Milano.
Finally, conclusions about the work done and possible further development are reported.

2 FREE-FORM SHAPE

In order to be able to analyze an object, it is fundamental to measure adequately its shape. This is particularly true when the object shape can not be easily described by regular geometries or by a combination of them.
In this chapter a definition of free-form shape is provided, along with an overview about various measurement techniques the most suited for the acquisition of this kind of objects. Finally a brief discussion about the most common surface modeling methods is reported.

2.1 FREE-FORM SHAPE ISSUE

There is no doubt that, nowadays, competitiveness in all branches of the manufacturing industry is extremely high. It follows the need of implementing processes that, on the one hand, reduce the time-to-market and, on the other hand, optimize the product performances tailoring it to particular customer requirements. In many industrial applications, such as aerodynamics and fluid dynamics, functional product characteristics depend tightly on the product's shape. Often, shapes that have to be handled cannot be described through traditional regular features or as a composition of well-known geometry solids; thus, they get the name of free-form shape.
Even though the word shape is commonly used in everyday language and there is a common understanding about what a shape of an object is, it is quite hard to provide a formal definition of it. Two intuitive definitions of shape have been suggested in [1]. The first one states: "Shape is all the geometrical information that remains when location, scale and rotational effects are filtered out by an object". Instead, if it is of interest to retain the scale information (size) as well, a better suited definition might be: "Shape is all the geometrical information that remains when location and rotational effects are filtered out by an object". The second definition actually says that two objects have the same shape if they are rigid body transformations of each other, instead the first one considers a scale transformation possible too. Also the difference between regular and free-form shapes is more intuitive rather than formal. Often free-form is a general characterization of an object whose surfaces are not of a more easily recognized class such as planar and/or natural quadric surface. Hence, a free-form object is often assumed to be composed of one or more non-planar, non-quadric surfaces (defined as free-form surfaces) [2].

This definition was inspired in [3]: "A free-form surface has a well-defined surface normal that is continuous almost everywhere except at vertices, edges and cusps". Sculptures, car bodies, ship hulls, air planes, human faces, organs, terrain maps are just some examples of free-form objects.
Dealing with free-form shapes is a twofold problem: in different situations, it may be necessary to design free-form shapes, in many others to recover information from a free-form real object.
Focusing on the latter issue, a growing need to deal with free-form shapes has been emerging over the last few years in different fields. For example, in the field of safeguard and registration of the cultural heritage, the main aim is the restoration and the reconstruction of small parts or of complete objects of art, but other applications exist as well, like their registration, the creation of 3D virtual museums, the analysis of ancient manufacturing and design techniques.
A comparable need has been manifested in the bio-medical research field, that nowadays has a huge importance in society, in order to model organs and tissues, acquired basically by means of X-ray or CT scan techniques. Handling biological surfaces is even more complicated than dealing with mechanical or artistic free-form surfaces, because they are absolutely unique and are not manufactured.
In the present work, the free-form problem is faced in response to a more engineering need (reverse engineering application). The aim is to recover the object shape to generate a 3D model to be used, for example, in support of fluid dynamics tests or stress analysis. The shape can be correlated to its performances and thanks to Computer Aided Design (CAD) software can be easily adjusted and tested again to improve its efficiency.
Over the last twenty-thirty years many efforts have been made by the international research community to face the 3D reconstruction and shape analysis problems and much progress has been achieved, above all in surface modeling thanks especially to NURBS (Non-Uniform Rational B-Splines). Nevertheless, the problem, despite this soaring demand for both reconstruction and geometrical feature extraction tools for free-form shapes, is far from being solved.

2.2 FREE-FORM SHAPE ACQUISITION

Measuring a shape consists in its sampling by using a proper technology. In case of regular geometry shapes fewer points need to be acquired, because, since the shape is a priori known, just its dimensions have to be measured. Instead, free-form shapes require a large number of points in order to get a proper reconstruction, and consequently, the measurement process must be optimized in terms not only of accuracy but also of speed.
Basically there exist two families of methods allowing user to mea-

sure a shape: one of them requires the contact between a probe and the surface to be sampled (contact techniques), the other one does not (non-contact techniques). In both cases, the result of the measurement process is a set of point sampled onto the external object surface whose spatial positions are known. This set of data is called *point cloud*.

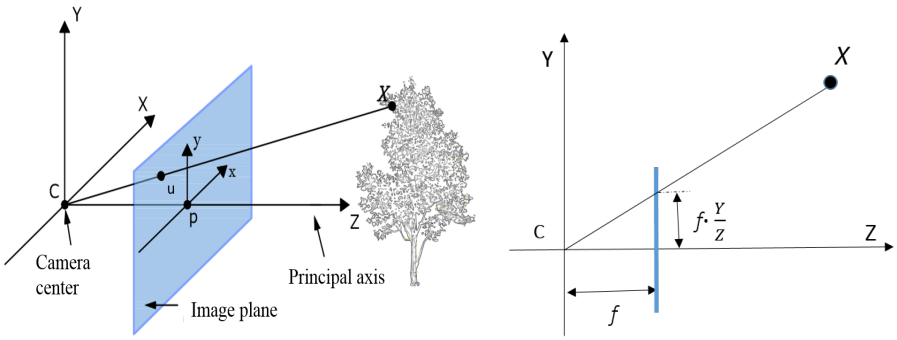
2.2.1 *Contact VS non-contact techniques*

Nowadays almost all the industrial metrology applications requiring very high accuracy are handled by using computer controlled Coordinate Measurement Machine (CMM) [4]. Basically, a probe is mounted on a traversing frame with three orthogonal axes embodying a spatial coordinate system and the points are sampled using that probe, whose position is known with very high accuracy in the coordinate system. In spite of its high measure performances, this technique has two main drawbacks: it is time-consuming, because just one point is sampled at a time, and it introduces intolerable load effects dealing with deformable shapes. Non-contact techniques overcome this limit. Among all the non-contact measurement techniques able to get a complete 3D information [5], it is possible to discern between methods suitable for measuring small objects, such as active triangulation, interference fringes or light field, and methods more suitable for medium or big shapes, such as stereo vision, phase variation, and Time of Flight (TOF) technology. The latter are described briefly in the next sections.

2.2.2 *Stereo vision*

Stereoscopic passive measurement systems determine the 3D scene coordinates by using the information contained in at least two images acquired synchronously.
The simplest way to describe the image formation process is the thin lens model [6] that is found on the hypothesis that light rays propagate in straight lines in an homogeneous medium, and all the rays getting to the sensor pass through a single point that correspond to the optical center of the lens and acts as projection center (Figure 2.1). A physical point $\boldsymbol{X}(X,Y,Z)$ in the 3D space is projected onto the image plane placed at a distance f - focal length- from the camera center C, and visualized by the sensor pixel $u(x,y)$. The similitude of the triangles implies the following ideal perspective projection:

$$\begin{aligned} x &= f \cdot \frac{X}{Z} \\ y &= f \cdot \frac{Y}{Z} \end{aligned} \tag{2.1}$$



(a) (b)

Figure 2.1: Pinhole camera model (a), and image formation process (b).

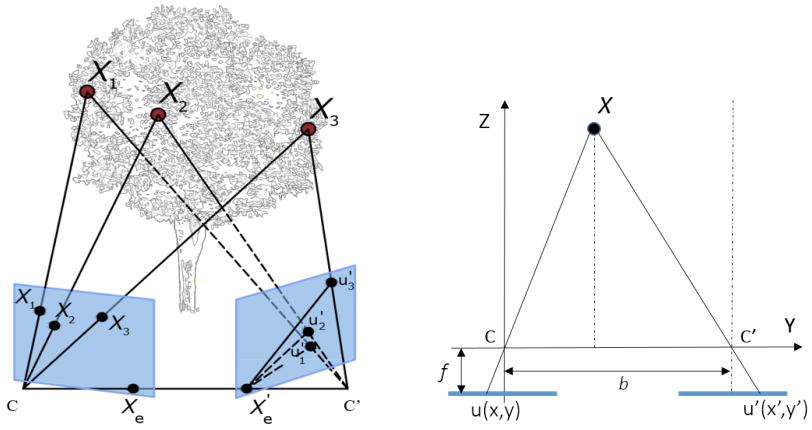
and the coordinate transformation from the camera reference system to the image reference system can be written as:

$$Z \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.2}$$

From equations 2.2 it follows that it is always possible to get *u(x,y)* knowing *X*(X,Y,Z), but the opposite is not true, in fact, knowing just *u(x,y)*, *u(x,y)* is defined up to a scalar value that is the Z point coordinate. It means that a digital camera alone is not able to retrieve a complete 3D information, then it must be integrated with other devices to do so.
Stereo vision generally uses two cameras separated by a distance, in a physical arrangement similar to the human eyes. In the stereoscopic approach, at least two images of the same physical point *X* are required to compute its 3D coordinates in the reference system of one camera (Figure 2.2a). Knowing the intrinsic camera parameters, such as the focal length *f*, the extrinsic camera parameters, such as their relative spatial position *b*, and observing the projections *u(x,y)* and *u'(x',y')* of the point *X* onto the image planes, the principle of triangulation leads to estimate the depth dimension Z (Figure 2.2b, Equation 2.3):

$$\begin{cases} Z = f \cdot \dfrac{b1}{x} \\ Z = f \cdot \dfrac{b2}{x'} \\ b = b1 + b2 \end{cases} \tag{2.3}$$

The main difficulty of the method is the identification of the homologous points in the different images. The homologous points are the points identifying the same point in the scene. These landmarks can coincide with points easy to be identified, like edges or corners.



(a) (b)

Figure 2.2: Stereo vision technique (a), and principle of triangulation (b).

A major challenge in stereo vision is solving the correspondence problem, that means detecting the homologous points in the different images. The homologous points are the points identifying the same point in the scene. These landmarks most of the times coincide with object edges or corners. Until the correspondence can be established, depth cannot be determined. Solving the correspondence problem require complex, computationally intensive algorithms for feature detection and template matching. Feature extraction and matching also require sufficient intensity or color variation in the image. This requirement render stereo vision less effective if the subject lacks these variations — for example, measuring the distance to a uniformed colored wall. In these cases, if possible, features have to be "artificiality created" by using markers or projecting laser spots (active stereo vision).

Accuracy below one millimeter on the maker spatial position can be reached for a well structured environment. Failures of this method might occur while acquiring in bad or variable illumination conditions: critical situations are for example, sun light dazzling the cameras, or indistinguishable objects in a dark room, or light reflections onto polished surface that might appear in an image but not in the other impairing the point correspondence matching. Time Of Flight (TOF) technology does not have this limitation because it does not depend on color or texture to measure distance, as explained in the next section.

However, the stereo vision is considered one of the quickest acquisition methods, and it is relatively affordable to implement, as most common off-the-shelf cameras can be used.

2.2.3 *Time of flight technology*

Another optical technique, suitable especially for medium and big targets, is based on the Time-of-Flight technology. A TOF based device operates by illuminating the scene with a light source, and observing the reflected light. Phase shift or time lapse between the illumination and the reflection is measured and translated to distance. Figure 2.3 illustrates the basic TOF concept. Typically, the illumination is pro-

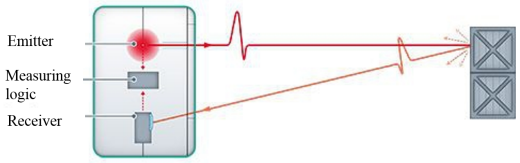


Figure 2.3: Pulsed Time Of Flight principle

vided by a solid-state laser or a LED operating in the near-infrared (NIR) invisible to the human eyes. An imaging sensor designed to respond to the same spectrum receives the light and converts the photonic energy to electrical charges. Note that the light entering the sensor has both the ambient light component and the reflected component.
In order to detect the phase shift between the illumination and the reflection, the illuminating source can be pulsed or modulated by a continuous-wave (CW), typically sinusoid or square wave. Square wave modulation is popular because it can be easily realized with digital circuits [7]. Pulsed modulation can be achieved by integrating photoelectrons from the reflected light, or by starting a fast counter at the first detection of the reflection. The latter require a fast photodetector, usually a single-photo avalanche detector (SPAD). This counting approach requires very fast electronics, since to achieve 1 millimetre accuracy require timing a light pulse of 6.6 picosecond in duration. This level of accuracy is nearly impossible to achieve in silicon at room temperature [8]. The pulsed method is straightforward. The light source illuminates for a brief period of time Δt, and the reflected energy is sampled at every pixel in parallel using two out-of-phase windows, $C1$ and $C2$, of the same Δt duration. Electrical charges accumulated during these samples, $Q1$ and $Q2$, are measured and used to compute distance using the formula:

$$d = \frac{1}{2} c \Delta t \cdot \frac{Q_2}{Q_1 + Q_2} \tag{2.4}$$

The CW method takes multiple samples per measurement, with each sample phase-shifted 90 degrees, for a total of 4 samples. Using this

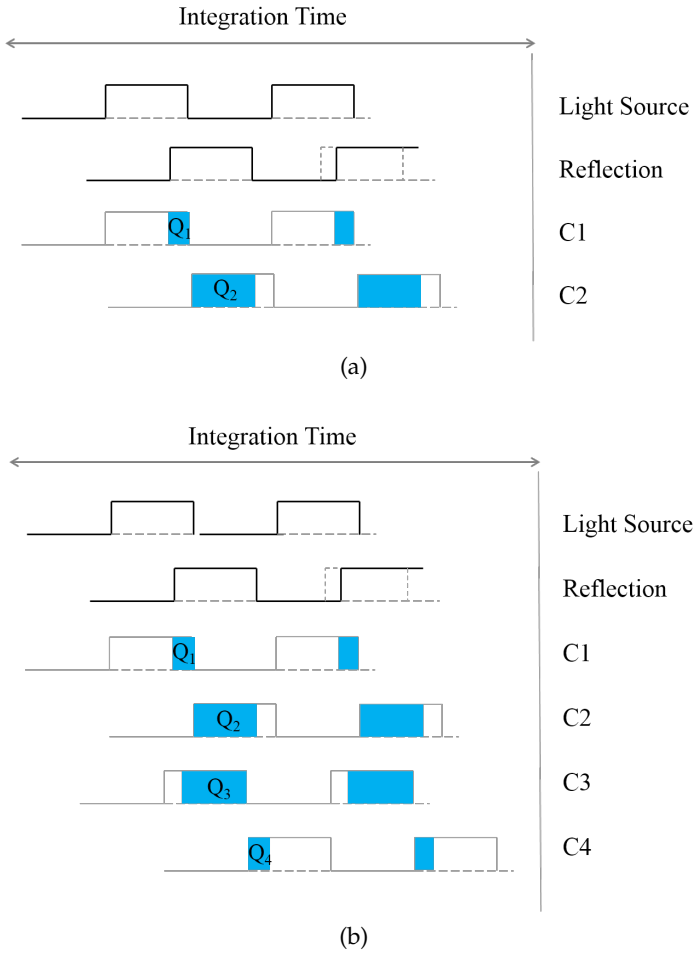


Figure 2.4: Time of flight methods: pulsed (a) and continuous wave (b)

technique, the phase angle between illumination and reflection, φ, and the distance d, can be calculated by:

$$\begin{aligned} \varphi &= \arctan\left(\frac{Q_3 - Q_4}{Q_1 - Q_2}\right) \\ d &= \frac{c}{4\Pi f} \cdot \varphi \end{aligned} \tag{2.5}$$

It follows that the pixel intensity amplitude A and offset B can be computed by:

$$\begin{aligned} A &= \frac{\sqrt{(Q_1 - Q_2)^2 + (Q_3 - Q_4)^2}}{2} \\ B &= \frac{Q_1 + Q_2 + Q_3 + Q_4}{4} \end{aligned} \tag{2.6}$$

In all equations above, c is the speed of light constant. At first glance, the complexity of the CW calculations as compared to the pulsed method may seem unjustified, but a closer look at the CW equations reveals that the terms ($Q3$–$Q4$) and ($Q1$–$Q2$) eliminate the effect of any constant offset from the measurements, such as that from the ambient light. Furthermore, the quotient of these in the phase equation

reduces the effect of any constant gains from the distance measurements, such as circuit amplification and attenuation, or the reflected intensity. The reflected amplitude A and offset B do have an impact the depth measurement accuracy. The depth measurement variance can be approximated by:

$$\sigma = \frac{c}{4\sqrt{2}\Pi f} \cdot \frac{\sqrt{A+B}}{c_d A} \tag{2.7}$$

The modulation contrast, cd, describes how efficient the imaging silicon separates and collects the photo-electrons. The reflected amplitude, A, is a function of the optical power. The offset, B, is a function of the ambient light. One may infer from Equation 2.7 that high reflected amplitude, high modulation frequency and high modulation contrast will contribute to increased accuracy. High offset, on the other hand, can lead to saturation and reduces accuracy. The fact that the CW measurement is based on phase, which wraps around every 2Π, means the distance will also alias. The distance where the aliasing occurs is called the ambiguity distance, da, and is defined as:

$$d_a = \frac{c}{2f} \tag{2.8}$$

Since the distance wraps, da is also the maximum measurable distance. If one wishes to extend the measurable distance, one may reduce the modulation frequency, but at the cost of reduced accuracy, as is according to Equation 2.7.
Instead of accepting this compromise, advanced TOF systems deploy multi-frequency technique to extend the distance without reducing the modulation frequency. Multi-frequency techniques work by adding one or more modulation frequencies to the mix. Each modulation frequency will have a different ambiguity distance, but true location is the one where the different frequencies agree. TOF sensors described so far perform just a punctual measurement, but there exist devices able to scan large 3D scenes providing information about the spatial location of thousands of points. A description of them is reported in the next sections.

2.2.3.1 *TOF cameras*

In the last few years a new generation of TOF sensors has been developed, which allows to acquire 3D point clouds from just one point of view and at video frame rates.
The working principle is the measurement of the TOF of an emitted signal by the device toward the object to be observed, with the advantage of simultaneously measuring the distance information for each pixel of a matrix sensor. Many terms have been used in literature to indicate such devices, normally called TOF cameras, Range IMaging (RIM) cameras or depth cameras [9]. Several devices are now

commercially available, presenting different ranges of acquisition, different resolutions and, thus, different prices. For example, the Mesa Imaging SwissRanger 4000 (SR4000), is one of the first TOF camera that entered the market. It has a depth measurement range up to 5 meters, a field of view of 69° x 55° and a resolution of 176 x 144 pixels. It is able to acquire up to 30 frames per second, and is sold for approximately $ 10,000 [43]. Slightly better performances can be achieved by using, for example, a PMD Technologies CamCube 2.0, that presents a higher operating range (up to 7 meters) and a better resolution (204 x 204 pixels) for the same price order of magnitude. At the end of 2013, Microsoft proposed a device called Kinect V2 based on a low cost CW TOF sensor able to acquire object at a distance up to 4 meters with a resolution of 512 x 424 pixels and a field of view of 70° x 60°. The device was originally designed for game controlling but lately it has been exploited in various applications due to its accuracy in depth measurement (around 1 mm) and its very competitive cost ($ 150). It is also equipped with a high resolution RGB camera (1080 x 1920 pixels) that allows for acquiring also intensity images. Compared to

(a) (b) (c)

Figure 2.5: Examples of commercially available TOF cameras: Mesa Imaging SR4000 (a), PMD Technologies CamCube (b), Microsoft Kinect v2 (c).

other technologies to obtain scene depth, TOF cameras exhibit some interesting properties such as the possibility to register both dense depth and intensity images, the high frame rate, and the low weight and compact design. The main drawback is the limited range of acquisition (few meters) and the limited vertical and horizontal field of view (tens of degrees).

2.2.3.2 *TOF Laser Scanner*

No limitations in vertical and horizontal field of view there exist dealing with TOF laser scanners. The working principle, in fact, consists in deflecting a pulsed laser to reach any point in the 3D space and to measure its distance from the device.
A common laser scanner is composed at least by an emitter/receiver pulsed infrared laser and a rotating prism (2D laser scanner - Figure 2.6a -). The rotation of the prism allows the deflection of the laser

light, enabling to scan all the points belonging to a plane perpendicular to the prism rotation axis (tilt angle). Properly coupling this hardware with a driving motor and an encoder mounted onto an external housing, it is possible to provide an extra rotation (span angle) and ,thus, to acquire a 360° scene. Few commercially available TOF scanners are reported in Figure 2.6b and 2.6c. These devices present accu-

(a) (b) (c)

Figure 2.6: Examples of commercially available TOF laser scanners: operating principle (a), Faro Focus X130 (b), Stonex X300 (c).

racy in the range of 2 - 50 mm for distance measurement between 1 - 300 meters and acquisition rates of ≈ 50000 points per second. Their main drawbacks are the high cost ($ 50000 - 100000) and the large dimensions (≈ 250 x 200 x 100 mm, 6 kg) that often lead customers to discard them. Finally, Table 2.1 provides a synthetic comparison of all the techniques described so far.

	Stereo Vision	TOF camera	TOF scanner
Operating range	Medium	Short	Large
Accuracy	High	Medium	Medium
Light influence	High	Low	Low
Acquisition rate	High	Medium	Low
Cloud uniformity	Low	High	Medium
Dimensions	Medium	Small	Big
Cost	Low	Medium	High

Table 2.1: Comparison of non-contact measurement techniques

2.3 FREE-FORM SHAPE MODELLING

Whatever the technique used for measuring a free-form object, the results is point cloud, i.e a discrete set of information regarding lots of points belonging to the sampled surface. Thus, it has to be processed in order to provide a full representation of the real object.

The capability to represent objects through proper geometric models is obviously the key issue of the reverse engineering. The term model identifies a mathematical entity describing real or virtual objects or phenomena: in no scientific or technological application a human interacts directly with the real world, but only with the model that he or she has created.
The collection of methods used to define shape and other geometric characteristics is referred to as geometric modeling. Geometric modeling combines techniques characteristic of many different branches, between them: analytic and descriptive geometry, topology, numerical analysis, set theory and matrix computation. Its aim is to provide efficient, powerful and flexible tools to represent 3D objects, that is their shapes, and to manipulate such representations. [Requicha80] introduced terminology and definitions of geometric modeling that have hugely influenced this discipline to this day. The turning point in geometric modeling for free-form shapes happened in the 1960s, when the French engineer Pierre Bézier proposed a new way to define polynomials. He replaced the traditional polynomial power basis form with a new form based on the Bernstein polynomial definition and on geometric coefficients, called control points, which are points in the 3D space that determine the curve/surface shape. Both forms are mathematically equivalent, however the latter is stunningly more powerful in geometric modeling applications, basically for three reasons [10]:

1. interactive shape design becomes very natural: shapes are modified just changing the control point positions;

2. algorithms dealing with the Bézier form have a geometric flavor rather than an algebraic one, that instead is characteristic of the power basis form;

3. Bézier form is far less prone to round-off error.

Bézier form was the starting point for the B-Spline implementation, which, nowadays, are de facto the standard in geometric modeling. In reverse engineering applications the model of an object is constructed from a great number of measured points which can be acquired in several different ways, depending on the object's characteristics (Section 2.2). Those points, that actually are a sampling of the object, essentially are able to capture its shape. Yet, in most applications, information they provide is absolutely insufficient. If the points are interpolated by using polygonal simplices, the resulting model consists in a polygonal mesh and then a piece-wise linear object reconstruction is obtained. Instead B-Splines (and especially NURBS that are the most flexible type of B-Splines) fit the measured points and perform a piece-wise smooth reconstruction. According to the characteristic of the surface expected and considering the uncertainty in the data acquisition, one might decide to exploit the linear approximation (high

detailed objects and high reliability in the surface measured points), or the smoothed approximation (no sharp variation in object surface curvature).

3

FLYING SAILS AS FREE-FORM OBJECT

The study of the flying sail shape came as an attempt to better understand what happens above the waterline while a yacht is sailing. Fluid Structure Interaction (FSI) codes are considered valuable tools for analyzing the yacht aero and hydrodynamics but they still need a massive validation work. They receive as input CAD models of rigging and sails that might not be reflective of the realistic sailing and trim conditions. Direct sail acquisitions could give a substantial contribution to this analysis.

In this chapter, different contributions performed by the scientific community in the last decades are reported; and the concept for an innovative sail shape acquisition system is proposed. For the first time, the flying sail is considered as a free-form object and thus measured and processed as such.

3.1 RELEVANCE OF FLYING SAIL SHAPE ANALYSIS

The study of the flying sail shape came as an attempt to analyze what happens above the waterline while a yacht is sailing.
Nowadays, Computational Fluid Dynamics (CFD) analysis has become a valuable tool in yacht design, and compared to wind tunnel or towing tank testing it is a cost efficient tool for analyzing aero- and hydrodynamics of design variations, allowing to alter geometry systematically according to predicted forces. While, it has become routine in hull design for the investigation of hydrodynamic forces, its application in predicting the aerodynamics of a yacht is not yet widely exploited since modeling what happens above the waterline is always a complex work. The angles of attack of the onset flow vary greatly and regions of flow separation are inevitable. Moreover, the geometry of the sails is usually unknown. In fact, the design shape is just one special case of the set of possible flying shapes that a sail might assume. The aerodynamic pressure distribution acting on the surface of the sail, the forces resulting from rig design and trim, and the elasticity of the sail material influence its actual flying shape in varying wind conditions. The combination of all these factors leads to a virtually infinite number of flying shapes and, this is especially true considering off-wind sails because of the lightweight construction materials and their relatively unconstrained nature. On the other hand, the geometry of the sails determine the aerodynamic flow around them and the resulting forces. It is thus of fundamental importance for the reliability of sail force predictions using CFD to know the real

geometry of the sail at given wind conditions. It is possible, although computationally expensive, to model the aero-structural coupling by combining CFD codes with Finite Element Analysis (FEA) of the sail. Up to date numerical codes still need a massive validation work on both wind tunnel tests as well as on water testing at full scale. Direct acquisitions can give a substantial contribution, supposing that reliable flying shapes are provided, reflective of the realistic sailing and trim conditions.
This work aims to add its own contribution to this issue. In fact, it presents a tool for the acquisition, reconstruction and geometrical analysis of the shape assumed by the sails during navigation. Due to its high variability in time and its unconstrained nature (especially true for off-wind sails), it has been considered as a free-from object. Thus, a proper device has been developed, able to measures thousands of points onto the sail surface; and a proper software has been realized to process the acquired point cloud to provide a 3D reconstruction of the sail shape.

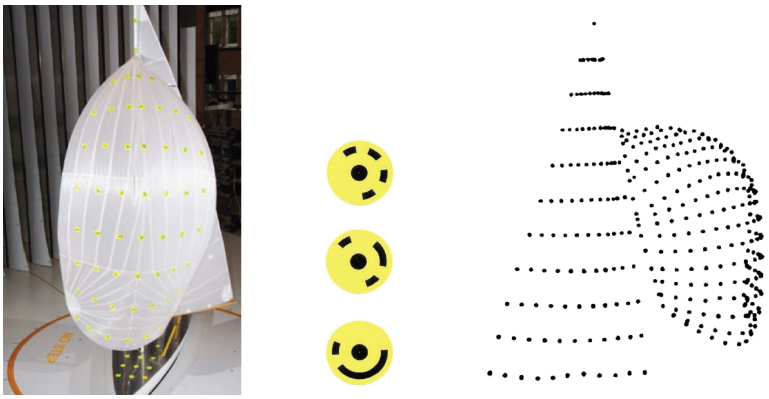
3.2 SAIL SHAPE MEASUREMENT TECHNIQUES

Sail aerodynamics is an active field of research in the scientific community. In the past years, several methods have been developed for a better understanding of the sail behavior. One of the topics of current interest is the knowledge of the flying sail shape.
Preliminary studies have been made on scaled yacht models during wind tunnel tests using different techniques. In 2002, Razenbach et al. [11] studied the resulting shape variation of a downwind sail shape in different wind conditions using a Coordinate Measuring Machine (CMM). CMM process requires a long period during the tunnel occupancy phase (about half an hour per configuration), but yields results within moments after the digitizing process is complete. Its principal disadvantage, however, is that the CMM and its operator can potentially influence the shape of the sail, moreover it requires the user to accurately select points directly upon the surface of the sail with the digitizing arm stylus.
To avoid the load effect, several authors exploited non-contact measurement techniques. For example, Fossati et al. [12], in 2008, used the photogrammetry technique and built an in house system composed of three cameras equipped with IR filters and IR coaxial lights (Figure 3.1b). Circular reflective markers were attached to the sail surfaces and images of the yacht viewed from different spacial positions were acquired simultaneously at high frame rates (Figure 3.1b). The IR light emitted illuminates the markers that can be easily visualized by the cameras. As explained in Section 2.2.2, knowing the camera calibration parameters, marker spacial positions are computed using triangulation with high accuracy thanks to the well structured envi-

(a) (b)

Figure 3.1: In house built stereoscopic camera systems for photogrammetry test on scale models.

ronment.
Photogrammetry has been exploited also at the YRU-Kiel Wind Tunnel, in 2009, by Graf et al. [13] that performed some tests on a set of spinnakers for an IMS600 custom design, focusing on the influence of wind velocity and wind twist on flying shape. More recently, in 2011, they presented other tests on two different asymmetric spinnakers [14]. Also for these tests several markers were attached to the sail surface: 60 coded targets per sail has been recorded by commercial cameras, identified and spatially located using a commercial software.
In 2012, Tahara et al. [15] considered the same measurement approach but using a more dense grid of markers: 100 through 200 target points were marked onto each sail surfaces. Aided by another commercial software and clicking on each corresponding target point indicated in several photographs, 3D point coordinates were computed (Figure 3.2c). A full scale test investigating the upwind sail aerodynamics was

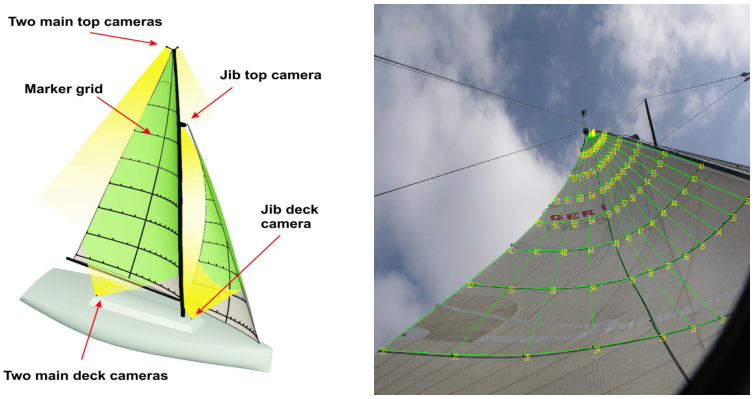


(a) (b) (c)

Figure 3.2: Dense marker grids for photogrammetry test on scale models.

performed by Clauss et al.on the 33-foot DYNA boat [16]. The hard-

ware mainly consists of a set of 6 digital cameras installed in fixed position on the boat and a grid of discrete markers applied to the sails forming a grid of horizontal and vertical lines is used to define their flying shape (Figure 3.3). As a general comment, it can be said that



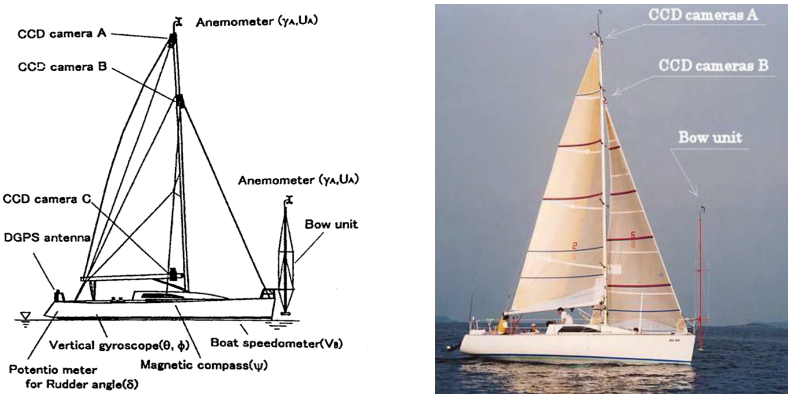
(a) (b)

Figure 3.3: Photogrammetry full scale test: camera positions (a),and marker grid identified (b).

the photogrammetry technique was judged accurate and relatively fast during the tunnel occupancy phase, requiring only that digital images were recorded from at least two vantage points. Its chief disadvantages are that it requires an intensive data post-processing and that suffers from occlusion problem. This means that sometimes some grid points are not properly recorded by the cameras. To overcome the problem, a large number of cameras is required, leading to relevant difficulties in system set up.

For full scale tests different authors had rather replace the marker grid with colored stripes drawn onto the sail surfaces. Masuyama et al. in 1997 [17] performed some pioneering tests on their dynamometer boat called Fujin. 6 CCD cameras have been used, 3 for the starboard side and 3 for the port side and placed as shown in Figure 3.4a. Few horizontal stripes, 5 cm in width, were drawn on mainsail and jib (Figure 3.4), and an image processing software retrieved shape parameters at these sail sections, such as maximum camber, maximum draft and twist angle values. The same procedure is used by several computer programs, which are nowadays commonly available on the market aiming at analyzing sail pictures taken on-board by means of a standard camera and providing some sail shape parameters in a certain number of sail sections.

For instance, North Sails has developed a proprietary software called Advanced Sail Analyzer (ASA) to digitize pictures deriving synthetic parameters for each section that has been marked by the horizontal stripes on the sails. In 2008, a method called Visual Sail Position and Rig Shape (V-SPARS) aimed at measuring also downwind sails has been presented [18]. It is based on single cameras placed on the

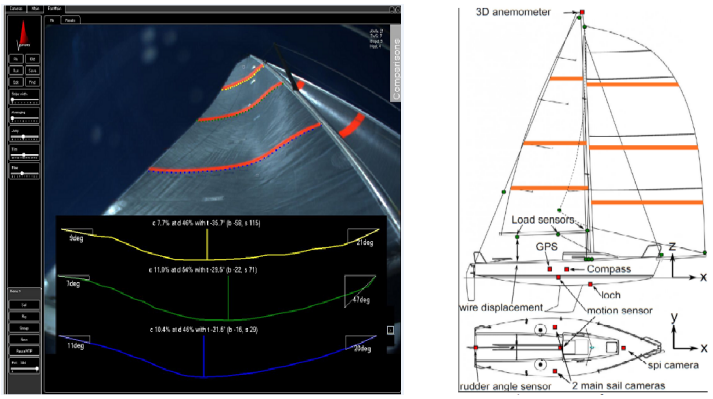


(a) (b)

Figure 3.4: Set up for the acquisition of colored horizontal stripe shapes

deck that capture fluorescent colored stripes on the sail and on an image processing software that produces the global coordinates of each stripe relative to a fixed datum position on the yacht. This method has been tested also for downwind sail aerodynamics investigation at full scale combining pressure and sail shape measurements [19].

The above mentioned background is extremely useful not only for



(a) (b)

Figure 3.5: V-SPARS system: software interface (a), and setup for downwind test

use while racing but also for sail design tool development which rely on a trustworthy validation process. One of the main drawback is that these systems can recover only few horizontal sections, deriving the relative angle between them but not being able to describe the three-dimensional position of the leading edge, therefore not being able to reproduce with accuracy the entire sail surfaces, which represent respectively the reference or the input in case of FSI or CFD calculations. Moreover, these systems rely on the assumption that the stripes remain in a horizontal plane, which is questionable for downwind sails especially when a large range of apparent wind angle is

considered.
Another intersting approach is the one proposed by Salzmann et al. [20] in 2007. They applied a surface deformation model, generally used for nonrigid 3D shape recovery, starting from the spinnaker CAD model provided by a sail-maker and images of its flying shape acquired from outboard (Figure 3.6). Results were promising but the

Figure 3.6

method always required a support boat to take photos from.

3.3 PROPOSED SAIL MEASUREMENT METHODOLOGY

Considering the above, a new tool for the sail shape acquisition has been developed. The idea is to try to overcome the limits of the techniques mentioned above.
First of all, the absence of contact between the hardware and the object during data acquisition was considered of particular interest for the application studied because non-contact instruments ensure no load effect onto the high deformable and extremely light sail surface. Moreover, with reference to the full scale situation, taking into account the great dimensions of the sails and the uncontrolled environmental situations, a tool presenting a wide measuring range and barely sensible to weather conditions is necessary. Finally, since the aim is to retrieve the overall sail shape and to analyze its surface as a free-form object, a relevant amount of data has to be acquired.
Thus, CMM, digital cameras and photogrammetry technique have been discarded. On the contrary, Time of flight technology has been chosen enabling to perform directly three dimensional measurements of the entire sail shape and not just bi-dimensional shape variation at few sail sections without impairing the flying shape. Moreover, it is suitable for outdoor tests because of its low sensitivity to light and environmental conditions and presents a wide measurement range (tens of meters) that perfectly meets the nautical field application.
A detailed comparison among all the techniques discussed so far is reported in table of Figure 3.7, highlighting weak and strong aspects of each.

Technique	Weak points	Strength points
Coordinate Measurement Machine	• Load effect: Alteration of the flying sail shape • Limited range of measurement: Suitable only for wind tunnel tests • Time consuming acquisition: Surface sampled point by point • Discrete measurements: Lots of points need to be measured	• 3D measurement: Retrieval of point spatial positions • Very accurate measurement: Micrometer precision
Photogrammetry Marker grid	• Time consuming set up: Tens of markers to be placed onto the sample surface 3 to 6 digital cameras required • Occlusion: Some markers can be lost • Influence of light condition • Discrete measurements: Lots of points need to be measured	• Instantaneous acquisition • High acquisition frequency • 3D measurement: 2D images + epipolar geometry to get 3D point coordinates • Cost affordable: Standard digital cameras required • Good accuracy: Millimeters precision
Colored stripes	• Discrete measurements Information only in correspondence of the stripes • Assumption stripes remain horizontal w.r.t the deck (2.5D measurement) - Questionable for off-wind sails • Influence of light condition	• Instantaneous acquisition • High acquisition frequency • Cost affordable Standard digital cameras required • Very compact design (easy to install onboard) • Good accuracy Millimeters precision
TOF cameras	• Limited range of measurement: Suitable only for wind tunnel tests Suitable only for flat sails • Influence of light condition • Influence of material properties • Limited field of view: Suitable only for flat sails	• Instantaneous acquisition • 3D measurement: Retrieval of point spatial positions • Cost affordable • Good accuracy: Millimeters precision • Dense point cloud obtained (thousands of points)
TOF laser scanner	• Non-instantaneous acquisition: Few seconds each acquisition • Measurement uncertainty of the order of the cm: For scale model tests might be unacceptable	• Very wide range of measurement: Tens of meters • No Influence of light condition • Dense 3D measurement: Retrieval of thousands of point spatial positions

Figure 3.7: Comparison of sail measurement techniques

3.4 SAIL SHAPE ANALYSIS

Whatever the measurement strategy involved, researchers and sailmakers use to quantitatively evaluate the sail shape extracting few geometrical information of various sail sections. These sections can be seen as airfoils and synthetic parameters such as chord length, camber and draft are extracted. Twist with respect to the boat center line is another interesting information providing an idea of the trim. Figure 3.8 shows the synthetic parameters computed. In details, for

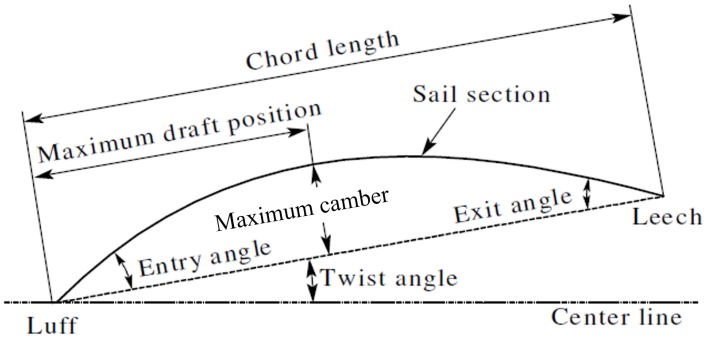


Figure 3.8: Sail section geometrical parameters

this work sail shape parameters are so defined:

1. *camber*: maximum deviation of the section arch to the chord;
2. *draft*: position along the chord (starting from the luff) of the camber point;
3. *twist*: angle between chord and boat centerline;
4. *entry angle*: angle between chord and tangent to the sail section at luff;
5. *exit angle*: angle between chord and tangent to the sail section at leech.

Sometimes it could be interesting to analyze the shape of the luff edge. Once more, it can be seen as an airfoil and the same parameters reported above are computed.

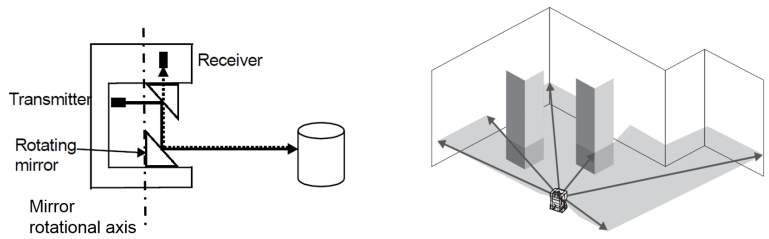
4

MATERIALS

In this chapter the tool developed for the acquisition and the analysis of the free-form sail shape is presented. Hardware components of the acquisition unit are described and a metrological evaluation of the measurement system is provided. Then, the data elaboration pipeline is presented, facing each step starting from raw data and ending with the evaluation of geometrical parameters and a 3D shape reconstruction. Validation of the data processing algorithm is provided too.

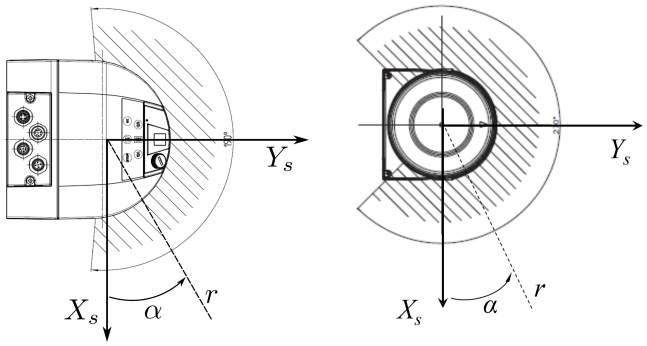
4.1 THE ACQUISITION SYSTEM

For this study two similar acquisition units have been realized. The goal is to reconstruct the entire sail plan, thus one unit has to be dedicated to the mainsail and another to the fore sail. Each unit is equipped principally with a TOF laser scanner, a gear-motor unit and other auxiliary sensors.
Two different TOF sensors by Sick have been tested (LMS 111 and LMS 511, detailed data-sheets are reported on Sick website [21]).
A schematic representation of their operating principle is reported in Figure 4.1a: a pulsed infrared laser beam at 905 nm is emitted and reflected on the target surface back to the sensor. The time between the transmission and the reception of the laser beam is used to quantify the distance between the scanner and the object. Thanks to a rotating mirror that deflects the laser ray with a regular angular step, the sensor samples points belonging to a plane 4.1b, providing 2D information for each of them in terms of polar coordinates, i.e. distance r and angle α. The origin of the polar coordinate system is located on the internal mirror rotational axis (Figure 4.2).



(a) (b)

Figure 4.1: 2D TOF laser scanner working principle.



(a) (b)

Figure 4.2: 2D TOF scanner reference systems: polar coordinate system (r, α) and correspondent Cartesian coordinates system (Xs, Ys) for scanner model LMS 511 (a) and LSM 111 (b)

Different scanning frequency can be set (25 to 100 Hz) depending on the angular resolution one desires (0.167° to 1°), for a maximum horizontal scanned angle α of 190° for a sensor and 270° for the other. This wide angular range has turned to be very useful in full scale tests since once placed the device at the bottom of the mast and aligned its Ys axis to the yacht center-line, it allows for scanning fore sails on both starboard tack and port tack without being moved.
Full scale tests are also feasible thanks to their operating range that covers distances up to 80 meters.

	Minimun	Typical	Maximum
Distance measuring range	0.5 m		20 m
Scan angle			270°
Scanning frequency	25 Hz		50 Hz
Angular resolution	0.25°	0.5°	0.5°
Systematic error		± 30 mm	± 50 mm
Statistical error (1σ)		± 12 mm	± 20 mm

Table 4.1: Data-sheet scanner Sick LMS 111.

Moreover, these scanners are based on the 5-echo technology that ensures the reliability of the measurements even in outdoor bad weather conditions: in Figure 4.3 each of the measurements 1 to 4 are above the trigger threshold, but none of them represents the actual object of interest. Conventional laser measurement sensors would fail in such a scenario. The Sick LMS series scanner selected detects the actual target with echo 5 no matter about fog, rain or dust.
Due to its intrinsic characteristics, a 2D laser scanner is able to measure only points lying in the same plane while in order to detect the

	Minimun	Typical	Maximum
Distance measuring range	0.7 m		80 m
Scan angle			190°
Scanning frequency	25 Hz		100 Hz
Angular resolution	0.25°	0.5°	1°
Systematic error		± 25 mm	± 35 mm
Statistical error (1σ)		± 7 mm	± 9 mm

Table 4.2: Data-sheet scanner Sick LMS 511.

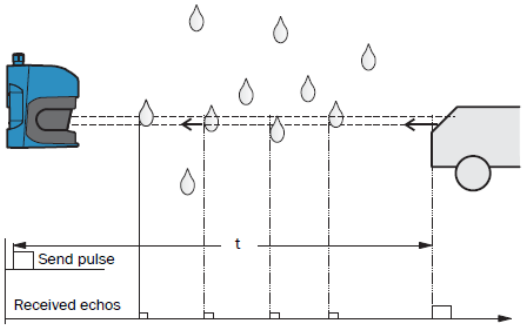


Figure 4.3: Sick laser scanner 5-echo technology

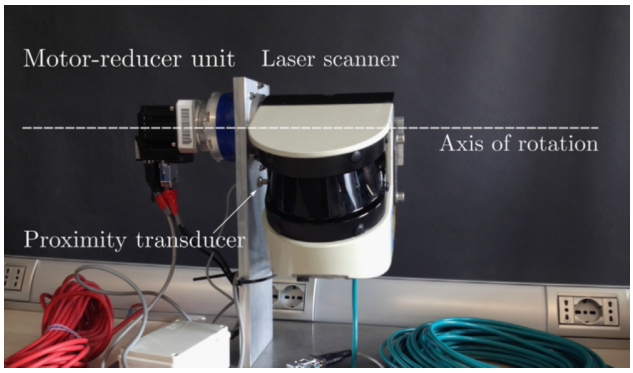
sail flying shape detection the entire sail surfaces has to be measured. To this aim, a custom made unit based on a brush-less motor and an epicyclical gear has been realized to enable the controlled rotation of the TOF sensor around an axis perpendicular to the mirror rotational axis. A proximitor is used to identify the initial scanning position for each data acquisition. Figure 4.4 shows the complete measurement systems. An in house computer program has been developed allowing for handling the acquisition phase: the brush-less motor is controlled via serial port and the measurement data stream is managed via Ethernet connection using TCP/IP protocol and stored onto a personal computer hard disk.
More in detail, the program allows the user to:

1. control the brush-less motor setting scanning velocity and acceleration;

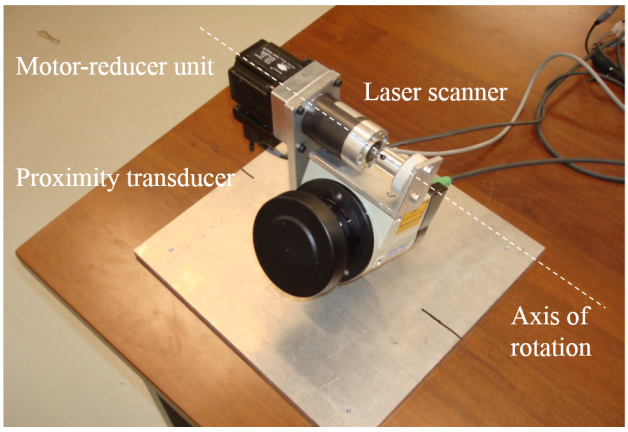
2. establish the connection to the laser scanner and gather the data from it;

3. set the scanning parameters, such as starting angle and vertical angle to scan;

4. receive and external trigger for running the scanning (useful for synchronization with other devices managed by a common control unit;



(a)



(b)

Figure 4.4: The TOF Flying Shape Detection Systems

5. visualize the scene scanned through a web-cam placed onto the TOF sensor.

Figure 4.5 shows the graphic user interface developed using LabView code. As explained so far, the TOF scanner provides as output a couple of arrays containing the polar coordinates (r and α) of the measured points. Cartesian coordinates in the scanning plane *Xs-Ys* (Figure 4.2b) are retrieved using the Equations 4.1:

$$\begin{cases} X_s = r\cos\alpha \\ Y_s = r\sin\alpha \end{cases} \tag{4.1}$$

A further coordinate transformation is necessary to interpret the data as three-dimensional points in the scene. For both TOF acquisition units, the origin of a world reference system has been placed in the middle of the TOF sensor, the X axis corresponds to the drive-shaft rotational axis, Z axis corresponds to the vertical quote - referred to the device basement - and Y axis completes the right-handed coordinate system. A representation for polar, Cartesian and absolute coordinate systems is shown in Figure 4.6; and the coordinate transformations is computed as reported in Equations 4.2.

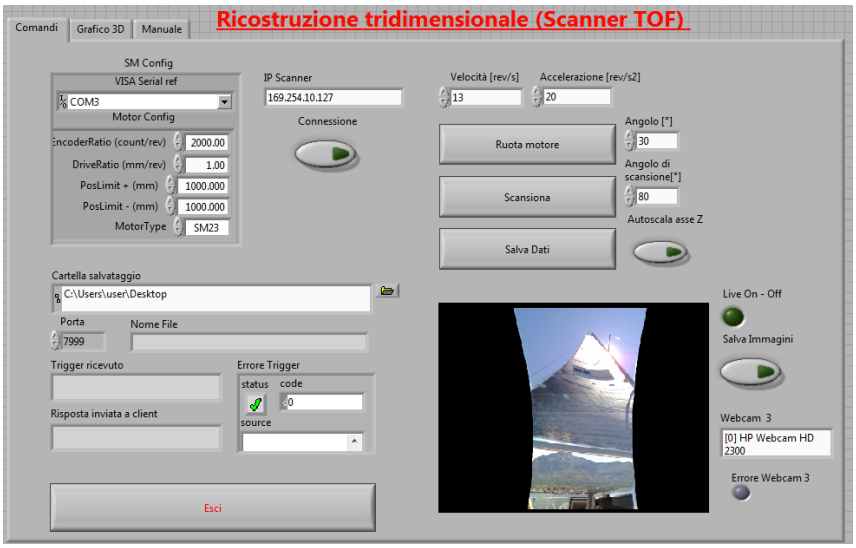
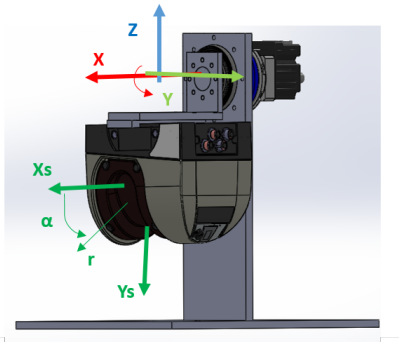


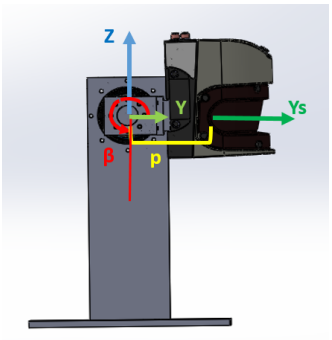
Figure 4.5: Acquisition software interface

$$\begin{cases} X = r\cos\alpha \\ Y = r\sin\alpha\cos\beta + p\cos(\beta+\theta) \\ Z = r\sin\beta\sin\alpha + p\sin(\beta+\theta) \end{cases} \tag{4.2}$$

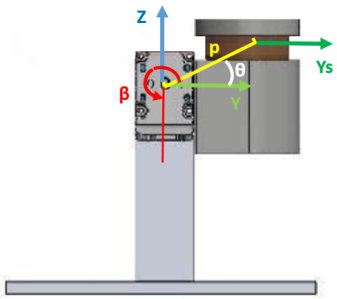
where: r and α are the polar coordinates provided by the scanner, β is the angle between the plane *Xs-Ys* and the Y axis (provided by the motor encoder), p and θ are geometrical constants. In particular, p is the segment that joins the origin of the absolute system to the origin of the scanner system while in the 90° rotated position, and θ is the angle between this segment and the Y axis (see Figure 4.6).
Several tests have been performed in order to accurately retrieve these geometrical parameters since there was no evidence of the exact positions of the scanner reference systems origins. By scanning know and regular geometry objects, p and θ have been carried out iteratively as results of the minimization of the difference between the measured values and the expected ones. Figure 4.7a reports the point cloud acquired scanning a plane wall of the wind tunnel chamber with one of the scanners involved. A least square based function minimized the distance between points and best fitting plane providing as a solution the geometrical parameters that satisfied the planarity condition the best. Other tests have been performed scanning cylinders of know diameters (Figure 4.7b) and verifying the points belonged to the reference cylindrical surface. Points departing respectively more than 7 and 9 mm from the planar and cylindrical reference surface are highlighted in red color. They are spread uniformly inside the whole scan leading to the conclusion that 3D world coordinates are correctly computed, and that these displacements from the reference surfaces have to be ascribed to the sensor measurement uncertainty, that is analyzed in details in the next sections.



(a)



(b)



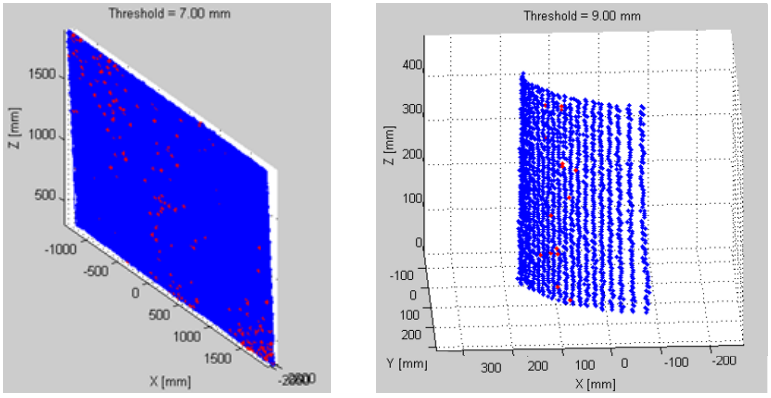
(c)

Figure 4.6: Polar (green: r and α), Cartesian (green: Xs and Ys) and Absolute (colored: X, Y, Z) coordinate reference systems for scanner LMS 511 (a,b) and scanner LMS 111 (c).

4.2 METROLOGICAL QUALIFICATION

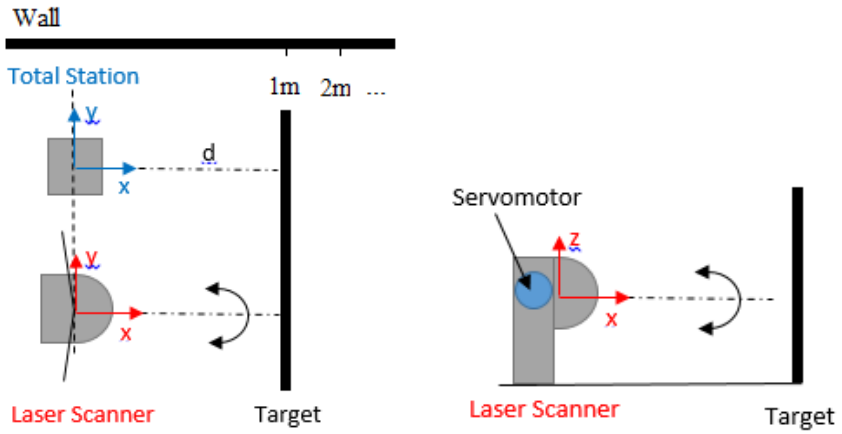
In order to estimate the measurement precision and accuracy of the TOF laser scanner, several tests have been carried out varying the distance, the orientation and the material of a planar target surface. Moreover, sensor warm-up time and mixed pixel problem have been faced. Finally, since the sensor has been designed to work in a wide outdoor environment, the effect of direct sun light has been taken into account.

Two different setups have been realized to test each TOF sensor. Setup *A* is shown in Figure 4.8. This setup was mainly used for the test regarding the influence of the target distance. Some papers reported in literature ([22], [23], [24]) suggest the use of a linear guide to increase the distance between target and sensors of few meters. Since in this case the range to be tested is by far wider (up to 20 meters), a different approach was adopted. A wall of the wind tunnel of the Politecnico di Milano - which is planar and smooth to avoid any turbulence - was considered as linear reference, and marks on the adjacent floor at different distances have been placed by using Leica Total Station measurements for a maximum distance of 15 m. The scanner was placed parallel to the wall and the planar target perpendicular to it. In order



(a) (b)

Figure 4.7: Scanning of known geometries: a planar surface (a), and a cylindrical surface (b). Points in red depart more than the threshold value from the reference surface.



(a) (b)

Figure 4.8: Setup *A*. Schematization of the setup components viewed from the top (a) and from a side (b).

to properly measure the distance to the target, a whole scan in the *XY* plane was registered and the deflected beam measuring the minimum distance value was detected (Figure 4.8a). Then, the tilt angle of the scanner was adjusted moving the servomotor 0.5° each step up or down since a minimum value for the beam selected before was found (Figure 4.8b). This procedure guaranteed that the scanner measured always the desired distance, and that the target plane was always perpendicular to the wall while moving away from the scanner. In other words, the scanner was used as a punctual distance-meter.
1000 scans have been acquired for each scenario. Distance values followed Gaussian distributions (Figure 4.9) and mean values were compared to those obtained through the Total Station aligned to the scanner and whose accuracy (0.25 mm at 35 m) is by far higher. Displacement between reference distance and measured distance are evalu-

ated in terms of systematic error, i.e. difference between reference distance and mean value for all the 1000 measured distances, and in terms of statistical error, i.e. the standard deviation for the 1000 samples.

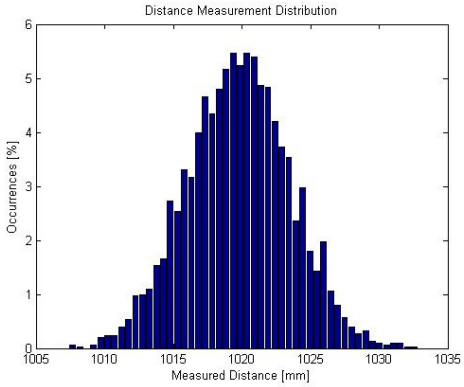


Figure 4.9: Example of data distribution for 1000 measurements for a planar target placed at a nominal distance of 1020 mm.

$$\begin{aligned} \overline{d} &= \frac{\sum_{i=1}^{N} d_i}{N} & \text{Systematic error} &= d_{ref} - \overline{d} \\ \sigma &= \sqrt{\frac{\sum_{i=1}^{N} (d_i - \overline{d})^2}{N}} & \text{Statistical error} &= \sigma \end{aligned} \tag{4.3}$$

A second setup has been built up for tests on the dependency of the angle of incidence between target surface and laser ray direction. Figure 4.10 shows it.

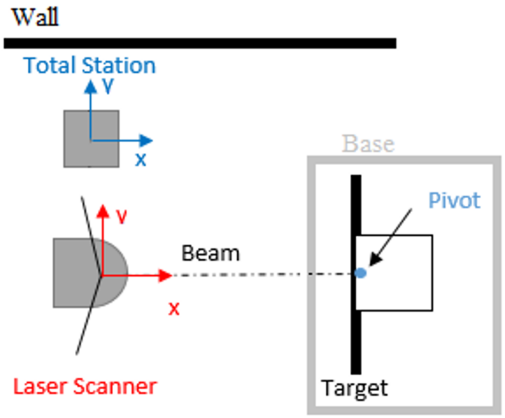


Figure 4.10: Setup *B*. Schematization of the setup viewed from the top.

This setup was composed of a flat basement and a planar target free to rotate over it thanks to a small shaft that joined the two elements. No translation was allowed. A goniometer was used to perform a

coarse rotation, while the exact angle value was obtained by computing the orientation of the plane that fits 9 points sampled on the target surface and acquired by means of the Total Station.
Depending on the test to be done, Setup *A* rather than Setup *B* could be exploited. For sake of brevity, in the next sections only the metrological qualification for scanner LMS 511 is described but same tests have been done with scanner LMS 111 leading to comparable results.

Drift effect

It is know from literature that various TOF sensors require minutes or even hours to provide a reliable measurement. Warm-up time has been investigated for the sensors used for this research.
Since the warm-up time is expected to increase by increasing the measured distance - as suggested in [25] -, the drift effect was analyzed using Setup *A* and placing the target at the maximum distance of interest (15 m). The scanner sampled 13500 full scans, over a stretch of time of about 2 hours. The results are visualized in Figure 4.11. Note

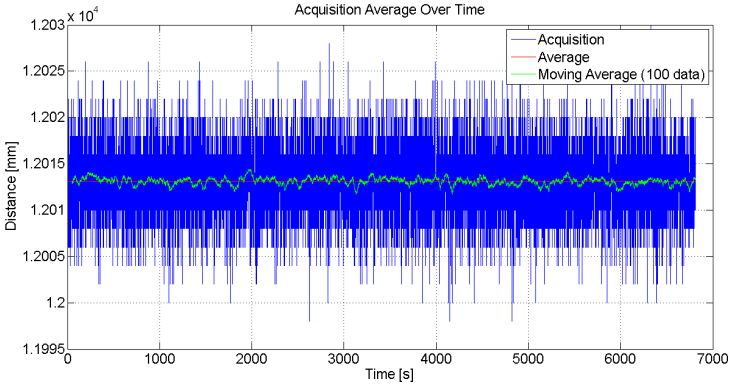


Figure 4.11: Drift test over 2 hours: there is no evident trend for distance mean values (green line).

that there is no evident drift effect, neither on the average distance values nor on the standard deviations. This means that these TOF scanners are prompt sensors and do not need a warm-up time. This result is in contrast with the majority of the previously mentioned papers in which the authors highlighted an hour-minimum warm-up time. The test was thus repeated at a different distance to verify the results, confirming the above. Moreover, to reject the hypothesis of a warm-up time longer than two hours, an experimental campaign of 9 hours was conducted for a shorter distance (≈3 m). Table 4.3 presents the trend of the average distance for each hour and the correspondent standard deviation.

Once again, results have been confirmed: no significant trend has been evinced.

Hour #	Mean [mm]	Standard Deviation [mm]
1	2971.8	4.2
2	2971.2	4.0
3	2971.1	3.9
4	2971.0	3.9
5	2971.1	4.0
6	2971.2	4.0
7	2971.2	4.0
8	2971.2	3.9
9	2971.1	3.9

Table 4.3: Drift test over 9 hours: no evident trend.

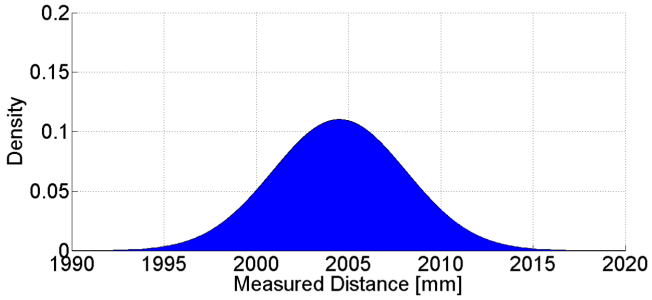


Figure 4.12: Distribution of the distances measured over 9 hours.

Distance effect

The laser beam presents a nominal output diameter of few millimeters (8 to 13 mm) and a nominal divergence of few milliradiants (4.6 to 15 mrad); this means that the size of the laser spot hitting a target increases by increasing the distance between target and sensor. Figure 4.13 schematizes the above and Equation 4.4 explicates the proportional relation between beam diameter and target distance.

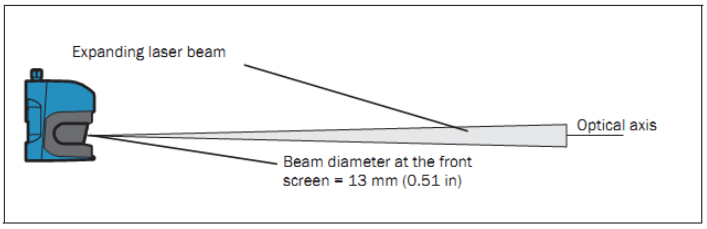


Figure 4.13: Laser beam divergence

$$\phi_t = \phi_o + d \cdot div \tag{4.4}$$

ϕ_t represents the spot diameter on the target surface; d the distance from the laser scanner, ϕ_o the output beam diameter, and *div* the

beam divergence in radiant unit. For example, measuring an object 10 m far away with a scanner having a laser beam of 13 mm and a divergence of 4.6 mrad means having a spot diameter at the target surface of 59 mm. The higher the beam diameter, the less accurate the measure.
In order to investigate the influence of the distance on the measurement accuracy, a planar target has been placed in front of the scanner at different distances: from 2 m to 12 m with steps of 2 m each, following the procedure described above for Setup *A*. Figures 4.14 and 4.15 show the absolute and percent displacements from the reference distance (taken by means of the Total Station) and the standard deviations. Measured distances depart from the reference value of few

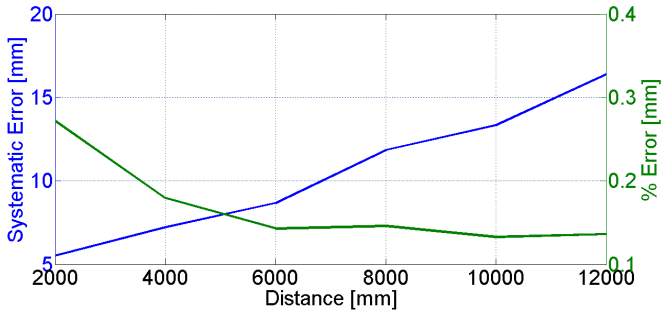


Figure 4.14: Distance test: errors in terms of absolute value increases almost linearly (blue line) while decreases exponentially considered in percentage over the target distance (green line).

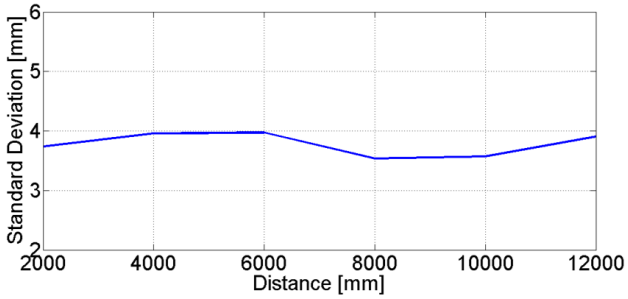


Figure 4.15: Distance test: the standard deviation trend does not vary significantly with the distance.

millimeters for the shortest distances. Differences reach tens of millimeters for wide range measurement, that are still acceptable considering the nautical application on full scale yacht.
The standard deviation varies approximately between 3.5 mm and 4 mm. These absolute values, that meet the manufacturer's specifications, once more can be considered not relevant if compared to the distance measured.

Influence of the target material

The TOF scanners considered are able to detect targets located at a distance approximately between 0.5 and 80 m. The measuring range depends not only on the type of sensor, but also on the reflectivity of the surfaces: the higher the reflectivity, the greater the measurable distance.
The majority of surfaces reflect the laser beam diffusely in all directions. The reflection of the laser beam varies as a function of the surface structure and color. Light surfaces reflect the laser beam better than dark surfaces and can be detected over larger distances. On very rough surfaces, part of the energy is lost due to shading and consequently the scanning range is reduced. At mirror surfaces the laser beam is almost entirely deflected. Thus, no measuring can be performed. Figure 4.16 schematizes the reflection of a laser beam that hits three different type of surfaces. Dependency on the target mate-

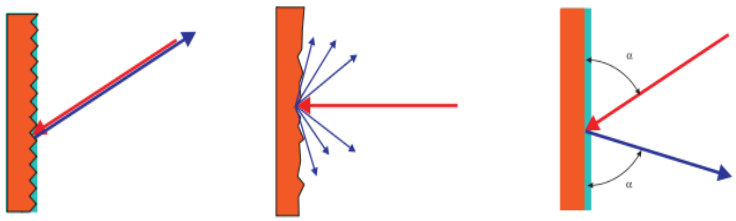


Figure 4.16: Laser reflection for different surfaces: perfect reflection (reflective materials, a), diffusion in all directions (rough surfaces, b), and total deflection (mirror surfaces, c).

rial properties has been investigated using Setup *A* and placing different planar target in front of the sensor at a fixed distance. To allow the comparison of our results with those obtained by other researchers, experiments have been carried out both for common materials as:

- wood;
- cardboard;
- aluminum;
- plastic.

and, considering our specific application, either for:

- reflective tissue;
- sail tissue.

For each material, the planar target has been placed in front of the scanner at a distance of around 2 m. The beam perpendicular to the

target was identified by searching for the minimum measured distance among those corresponding to the target. 1000 whole scans have been acquired for each material. Only distances registered by the perpendicular beam have been considered. Figure4.17 shows the distribution of the 1000 scans.

Note that the material presenting the lowest standard deviation (3,3

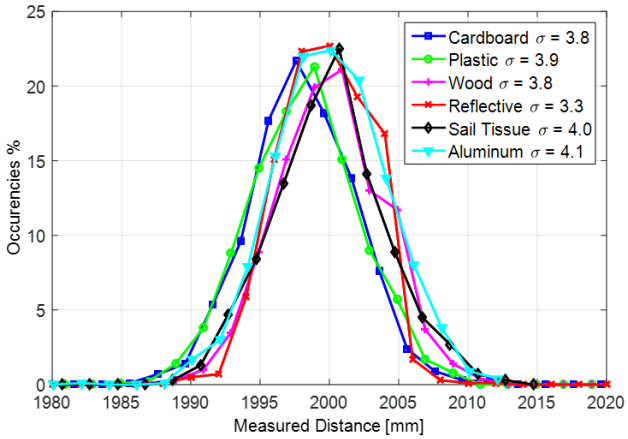


Figure 4.17: Material test: distributions of distance values acquired for different material targets placed at 2000 mm away.

mm) is the reflective one (red line). This could be due to the higher signal intensity that reaches the receiver. The worst material came out to be the sail tissue: standard deviation (4,1 mm, black line). This is probably due to the transparency of the mold that did not reflect properly the laser signal. Figure 4.18 reports a picture of the sail highlighting the transparency of the surface (note the cloud beyond the sail).

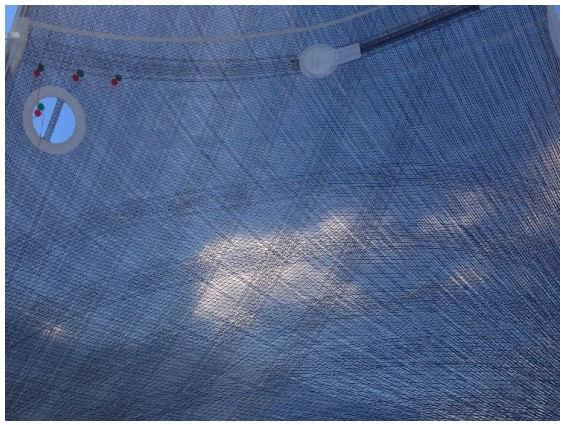


Figure 4.18: Transparency of the sail tissue.

Target orientation

Considering Setup *B*, a wooden planar target placed at a distance of 1000 mm is rotated from 0° (target facing frontally the sensor) to 80°, with step of 10° each. The aim is to investigate the influence of the angle of incidence between the laser beam and the target surface. 1000 scans for each angle have been acquired, and only the distances retrieved by the central ray were considered. Afterward, the target has been moved away from the scanner to reach the distances of 2000 mm and 3000 mm, and the procedure was repeated to support the first test. Figure 4.19 presents the standard deviation trends for the three distances as a function of the angle of incidence between the laser beam and the target surface. Standard deviation trends looked

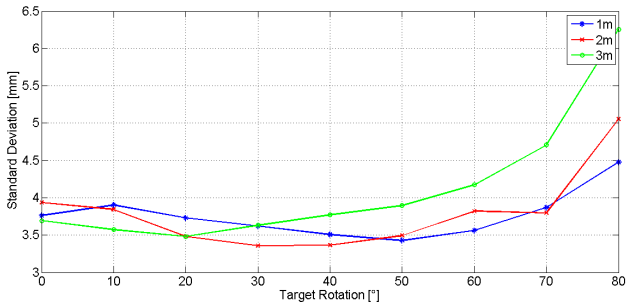


Figure 4.19: Angle Effect: standard deviation trends for 3 different tests. Beyond 70° the standard deviation values rise considerably.

similar for the three tested distances and the values were comparable with those shown in the sections above, up to an incidence angle of approximately 70°. A more oblique beam leads to a considerable increase in the data spread. This test was of particular interest to understand some faulty measurements in the data acquired on the boat. In fact, the sensor placed astern and scanning the mainsail was not able to estimate a distance for some point on top of the sail since the laser beams reached the surface with an angle wider than 70°. Another TOF sensor position might be considered.

Color effect

In this section, the effect of the target color is discussed. Different colored adhesive films have been attached, onto a glass planar surface placed at a fixed distance of 2000 mm far from the sensor. Once more, Setup *A* was used and 1000 scans have been acquired for each film. Data could be approximate by a normal distributions and mean values and standard deviations have been computed. Various colored films were available, but the analysis was limted to the most interesting cases suggested in [22], [24] and [26]. In particular, considered

colors are: green, yellow, black, and two different gray level films. Figure 4.20 shows the results for these tests.

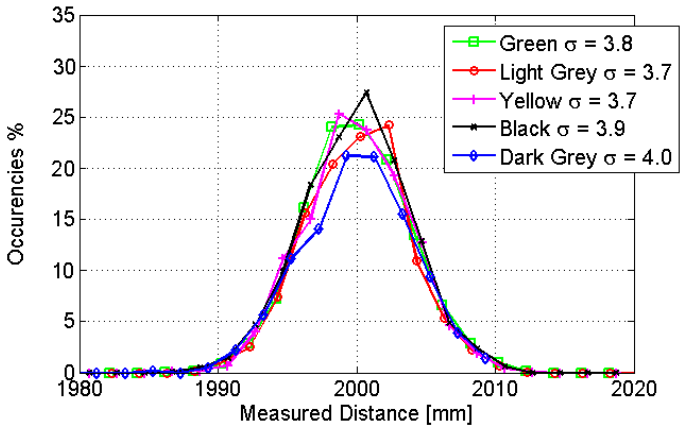


Figure 4.20: Color test: distributions of distance values acquired for target placed at 2000 mm and covered with adhesive films of different colors.

From these tests, it can be said that the effect of the color does not in-

Color	Mean [mm]	Standard Deviation [mm]	Systematic error [mm]
Green	1997.8	3.8	2.2
Yellow	2005.3	3.8	5.3
Black	2005.3	3.9	5.3
Light Grey	1997.7	3.7	2.3
Dark Grey	2014.7	4.0	14.7

Table 4.4: Color effect (target placed at a nominal distance of 2000 mm).

fluence significantly the measurements. However, looking into detail, it is possible to notice that:

- dark gray and the black target present the highest values of spread (standard deviation respectively 4.0 and 3.9 mm);
- dark gray target presents the worst performance (maximum displacement to the nominal distance: 14.7 mm);
- light grey test presents the best performance (minimum displacement to the nominal distance: 2.3 mm).

Thus, in conclusion, brighter targets lead to slightly better performances, probably because of a better reflection of the laser signal, in agreement with the results reported in literature.
Comparing the spread of the measures obtained testing the sail tissue

(4.1 mm) to the one obtained in the light gray test (3.7 mm), coating the sail with this bright film to increase the quality of the field measurements.

Angulation Effect

This test aims to clarify the possible differences in data returning from different angles (i.e. different beams) to the scanner. Recalling the operating principle, these could be caused by imperfection in the rotating prism inside the scanner. For this test the scanner was placed onto the rotating base and the target was fixed at a distance of 2000 mm. The scanner was rotated and 1000 measurements were acquired for the beam perpendicular to the target. The setup is schematized in Figure 4.21. Results for the experiments are shown in Figure 4.22. Note

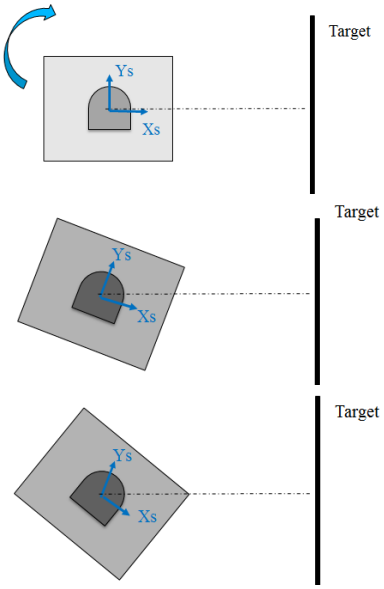
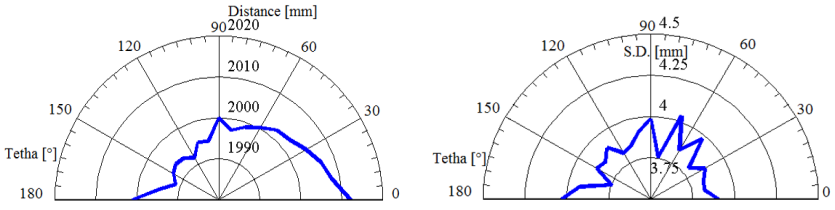


Figure 4.21: Setup angulation test: the scanner is progressively rotated with respect to the target.



(a) (b)

Figure 4.22: Angulation test: standard deviation is not influenced while distance measurement presents a bigger error for the beams related positive *Xs* values.

that the resulting standard deviation values are all included between

3.7 mm and 4.0 mm, consistent with the previous considerations. On the other hand it seems that the sensor slightly overestimated the distance in the first section of the acquisition, from 0° to 60°, and it underestimated it on the opposite side, from 90° to 180°. This effect could be due to a misalignment between the rotational axis used to turn the scanner and the internal mirror rotational axis of the scanner. The manufacturer assured that it is placed exactly in the middle of the physical sensor case, but the impossibility to open it prevented to verify this information.

Light effect

Some faulty cases came out during on field acquisitions because of the sun light that directly hit the scanner. Thus, the effect of an intense light source onto the sensor performances was investigated. To reproduce this situation a lamp was placed beyond the sail target, so that the source of light was completely covered by the sail tissue and then progressively moved close to an edge until it came out completely. The relative distance between sensor and sail tissue did not change. A similar test was carried out outdoor, placing the laser scanner and the sail target in the sun light direction. As shown in Figure 4.23, the light source impaired the measures. Moreover, as soon as the light entered the sensor, the standard deviation increased almost three times. The highest peaks in blue and red curves correspond to

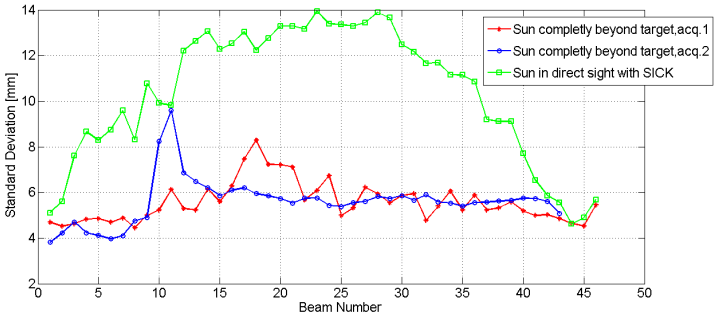


Figure 4.23: Light Effect: uncertainty increases while sun light hits the sensor.

the measure performed by the beams in the same direction of the line joining sun and scanner. For each case the measurement standard deviation worsen with respect to a standard lighting case, but the worst situation occurs when the sun hits directly the scanner (+ 350% Standard Deviation).

Mixed pixel

The mixed pixel problem occurs when the spot mark of the laser beam, that hits the tested surface, falls on the edge of the target. For that point, the sensor averages the contribute of the signal reflected by the target and the one reflected by the background leading to an averaged estimated distance. Two tests were reported to ver-

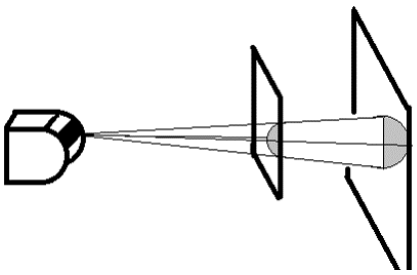


Figure 4.24: Mixed pixel problem.

ify that the mixed pixel problem occurs independently from the gap distance between target and background: test 1 considered a target placed at 1000 mm far from the sensor and a background at 2000 mm; test 2 considered a target placed closer to the background (at 1750 mm far from the sensor). Figure 4.25 shows the first test. Both tests

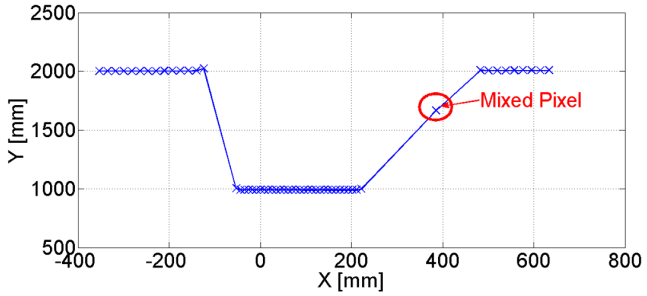


Figure 4.25: Mixed Pixel - Planar target placed at a distance of 1000 mm from the background.

present non physical points in the gap between the planar target and the background due to the averaged distance performed by the scanner. Mixed pixel problem was encountered also scanning sail plan for scaled model in wind tunnel. Interference between the two sails appeared as shown in Figure 4.26. This unwanted points can be deleted using some proper filters as explained in Chapter 5.
Concerning tests on field, the sail surface can be correlate to the planar target and the sky to a background placed at an infinite distance from the sensor. Thus, a test with a planar target oriented towards the sky has been performed. No mixed pixel error was evident. This probably means that whenever the laser spot hits the sail edge, if enough signal intensity is reflected, the measure is retrieved, otherwise the

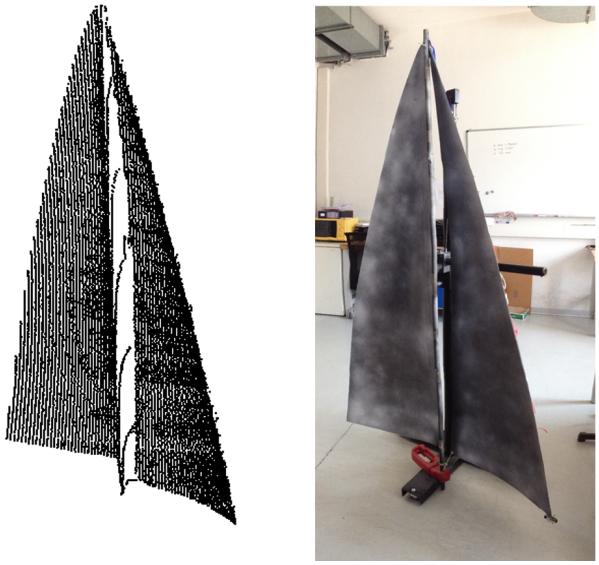


Figure 4.26: Mixed Pixel - Interference between mainsail and jib surface leads to incorrect measurement (see the points departing from the mast).

point is lost.

5 METHODS

Acquired data comes in the form of 3D point clouds, representing in a discrete way the sampled surface. Data are processed to finally provide a continuous representation of the scanned shape, i.e a CAD model to be used for shape visualization and analysis. The steps composing the data elaboration pipeline are presented in details in this chapter. Tests onto artificial clouds have been preliminary done to validate the code, and then the procedure has been tested onto real acquisitions to verify its applicability to the study case.

5.1 DATA ELABORATION PROCEDURE

The TOF acquisition unit provides data in the form of 3D point clouds representing, in a discrete way, the scanned shape. For reverse engineering applications, instead, it is necessary to realize a 3D model, i.e. to numerically, and continuously, describe the object scanned. The model can be used to render images of the object from arbitrary viewpoints, to simulate the appearance of the object in novel synthetic environments (for CFD codes for example), to perform geometrical measurements and shape analysis, and to redesign the object using the model as starting point.
The pipeline of operations for turning the acquired data into a usable model is reported in Figure 5.1, along with a brief explanation about how each step has to be adapted to the application considered in the present work. In fact, standard algorithms are available in literature to face each pipeline operation, but dealing with a custom made device it is necessary to adjust them to treat non-uniform and non-organized point clouds. First of all, the clouds are oriented in a reference coordinate system, usually aligning the yacht center-line to X axis, and the mast to Z axis. Once the clouds are oriented, segmentation is performed to isolate the points in the cloud representing the sail surface. Proximity and point normals are the principal criteria to identify and separate different objects in the scene. Then, denoising, statistical outlier removal and cloud smoothing are performed to enhance the sail cluster, and consequently surface modeling problem is faced. A linear piece-wise approximation of the surfaces can be obtained by connecting sample points to form a triangular mesh, otherwise, for a smoother surface B-Spline based methods are exploited. Finally, the surface reconstructed is analyzed and geometrical param-

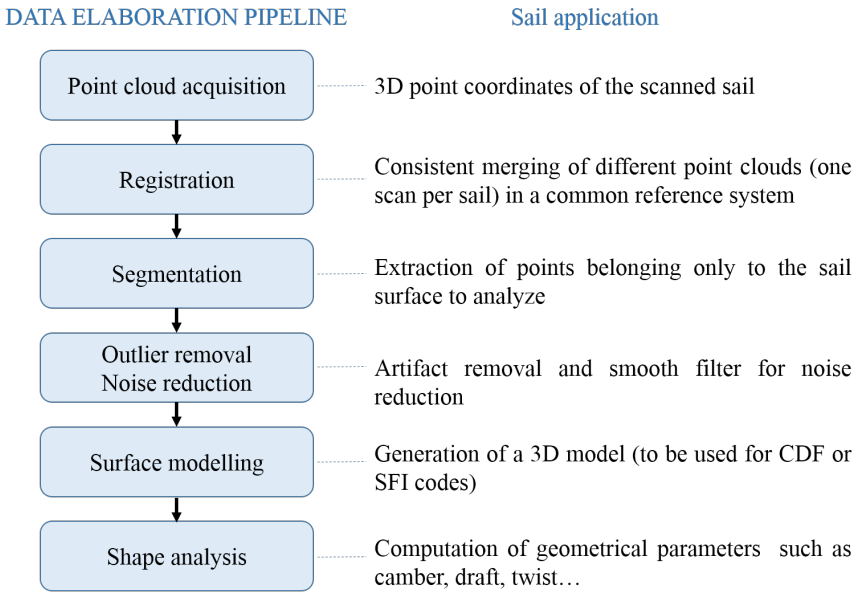
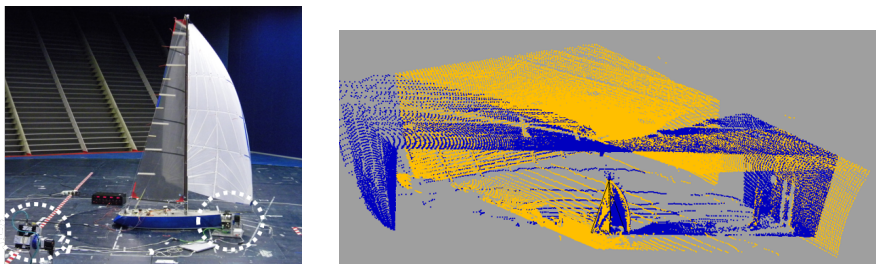


Figure 5.1: Point cloud elaboration pipeline.

eters are extracted to provide synthetic information about how the sail is flying. Each step is described in detail in the next sections.

5.2 POINT CLOUD ACQUISITION

As mentioned before, the acquired data come in the form of point clouds. A point cloud is a data structure used to represent a collection of multi-dimensional points. Commonly, a point cloud is a three-dimensional set that encloses the spatial coordinates of an object sampled surface, as in this context. But using devices able to acquire also images of the scene, information about the color of each point sampled can be added leading to a four-dimensional cloud.
In Figure 5.2 an example of 3D point clouds acquired during wind tunnel tests is presented. The position of the acquisition units are marked in the dashed circles.

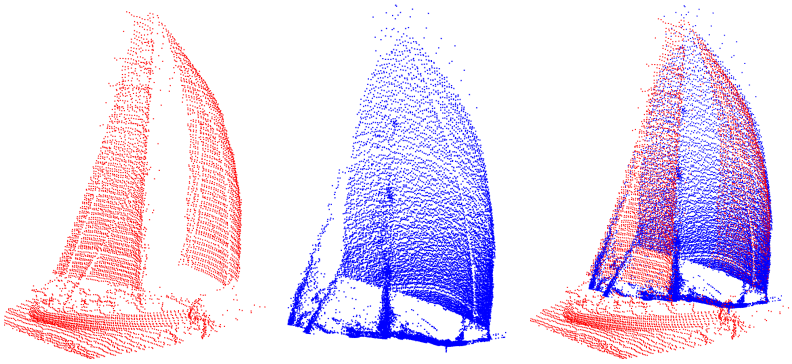


(a) (b)

Figure 5.2: Example of data acquisition: wind tunnel setup photo (a), and 3D point clouds acquired (b).

5.3 POINT CLOUD REGISTRATION

The problem of consistently aligning various 3D point cloud data views into a complete model is known as registration. Its goal is to estimate the relative positions and orientations of the separately acquired views in a global coordinate framework, and to find a way to align and merge them together into a single point cloud so that the intersecting areas overlap perfectly.
The work presented in this section aims at finding point-to-point correspondences in real noisy data views in order to estimate rotation and translation matrices that transform the scanned points into a consistent global coordinate framework.
A motivation example in this sense is given in Figure 5.3 , where two separate scans from two TOF acquisition units, one dedicated to the mainsail and the other to the off-wind sail, are merged to form a unique point cloud representing the whole yacht model.
Different methods can be used for registering clouds, starting from



(a) Mainsail (b) Spinnaker (c) Complete sail plan

Figure 5.3: Point cloud registration: two clouds representing the yacht model from different views (a) and (b) are consistently merged together (c).

a manual approach where the user has to drag, move and rotate one cloud to match the other and ending with automatic algorithms that iteratively estimate a transformation matrix to apply to the moving cloud to make it closer to the reference at each step; as reported in the next sections.

5.3.1 *SDV registration*

If correspondences between point in the reference cloud and points in the cloud to register are known, the registration problem can be easily solved using SVD decomposition (see Appendix A for details) of the matrix containing information about 3D coordinates of points

belonging to both the clouds [27]. In fact, SVD of a matrix M can be written as:

$$M = U\Sigma V^T \tag{5.1}$$

where U is an $m \times m$ real or complex unitary matrix, Σ is an $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal, and V^T is an $n \times n$ real or complex unitary matrix.
When M is an $m \times m$ real square matrix with positive determinant, U, V^T, and Σ are real $m \times m$ matrices as well, Σ is a scaling matrix, and U, V^T are rotation matrices. Imposing a unitary scale factor, U and V^T define the registration matrix [28], which can be expressed as a unique rotation and a translation, obtained as:

$$R = V \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det\left(V \cdot U^T\right) \end{bmatrix} \cdot U^T \tag{5.2}$$

$$T = p_2 - R \cdot p_1 \tag{5.3}$$

where $p_2$and $p_1$are the center of mass, respectively, of the target data-set and the source data-set.
The input matrix M, as anticipated above, is composed of two other matrices A and B as follows:

$$\bar{\mu_x} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \bar{\mu_y} = \frac{1}{N}\sum_{i=1}^{N} y_i \qquad \bar{\mu}_z = \frac{1}{N}\sum_{i=1}^{N} z_i \tag{5.4}$$

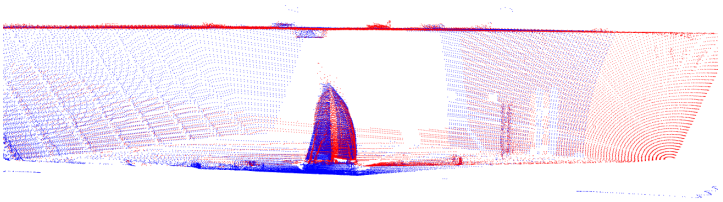
$\bar{\mu_R} = [\bar{\mu_x}, \bar{\mu_y}, \bar{\mu_z}]$ is the center of mass of the reference point cloud, and $\bar{\mu_F}$ the center of mass of the registering cloud.

$$A = \begin{bmatrix} x_1^R - \bar{\mu_{Rx}} & y_1^R - \bar{\mu_{Ry}} & z_1^R - \bar{\mu_{Rz}} \\ .. & .. & .. \\ x_m^R - \bar{\mu_{Rx}} & y_m^R - \bar{\mu_{Ry}} & z_m^R - \bar{\mu_{Rz}} \end{bmatrix} \tag{5.5}$$

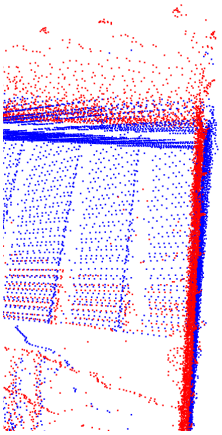
$$B = \begin{bmatrix} x_1^F - \bar{\mu_{Fx}} & y_1^F - \bar{\mu_{Fy}} & z_1^F - \bar{\mu_{Fz}} \\ .. & .. & .. \\ x_m^F - \bar{\mu_{Fx}} & y_m^F - \bar{\mu_{Fy}} & z_m^F - \bar{\mu_{Fz}} \end{bmatrix} \tag{5.6}$$

$$M = B * A^T \tag{5.7}$$

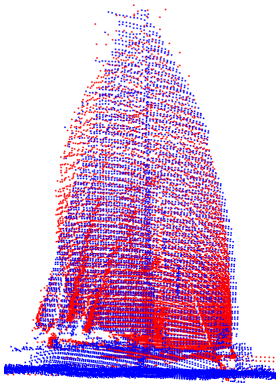
As mentioned above, it is mandatory that point coordinates in Equations 5.5 and 5.6 are properly listed according to a point to point correspondence in the input data-sets. This means that a selected list of points $p_i \in P_1$ have to coincide from a feature representation point of view with another list of points $q_j \in P_2$, where P_1 and P_2 represent two partial view data-sets. Differently said, the sets of p_i and q_j have been sampled on the same real world surfaces, but from different acquisition poses. This means however, that in the complete point cloud model that needs to be created, they could be merged together, especially if their coordinates are equal $p_i = q_j$, thus reducing the number of points overall.
Because the quality of the data-sets is influenced by sensor noise and other perturbing factors, the coordinates of the points will almost never be equal unfortunately, and the above simplification will not hold. The greater the noise level, the worse the registration result.
Tests conducted on real acquisitions confirmed the above: as notable in Figure 5.4a the clouds could not be perfectly aligned due to the noisy data. The faulty alignment is visible especially looking at the wind tunnel walls, as shown in Figure 5.4b, and in the gennaker luff, in Figure 5.4c. Resulting distances between the two registered clouds



(a) Wind tunnel cloud registration



(b) Wall close-up



(c) Gennaker luff close-up

Figure 5.4: Example of SVD registration for real data: acquisition noise impairs the cloud alignment.

are more than ten times greater than the acquisition accuracy. They can not be ignored and lead to discard the SVD method for the real

acquired cloud registration. Another weak point of this methods is the fact that the user has to manually select the points in the two clouds and to set their correspondences. The results depend on the accuracy used in picking these points.

5.3.2 *ICP and Feature Based registration*

Automatic registration of clouds can be done using other algorithms present in literature. They can be divided in two main classes: the Feature Based algorithms, and the Iterative Closest Point (ICP) algorithms.
In the feature based approach, points representing particular features such as edges or corners of objects lying in a scanned environment are identified in the reference cloud (*keypoint detection*) and matched with similar features identified in the registering cloud (*correspondence matching*). Once the correspondence list is set, an SVD registration is performed (*transformation estimation*). Few keypoint detection methods have been recently developed for 3D point clouds such as Harris 3D [29], Scale Invariant Feature Transformation (SIFT 3D) [30], and Intrinsic Shape Signature (ISS 3D) [31]. They are mainly based on the assumption that interesting points are described by evident changes in normal direction and in curvature (refer to Appendix A for a definition of these cloud characteristics).
The ICP approach, instead, has been proposed by Besl [32] and Zhang [33], and consists of an iterative descend method which tries to find the optimal transformation (rotation matrix R and translation vector T) between two data-sets $p_i \in P1$ and $q_j \in P2$ by minimizing the Euclidean distance error metric between their overlapping areas.

$$\min \sum_{i=1}^{n} \|R \cdot p_i + T - q_i\|^2 \tag{5.8}$$

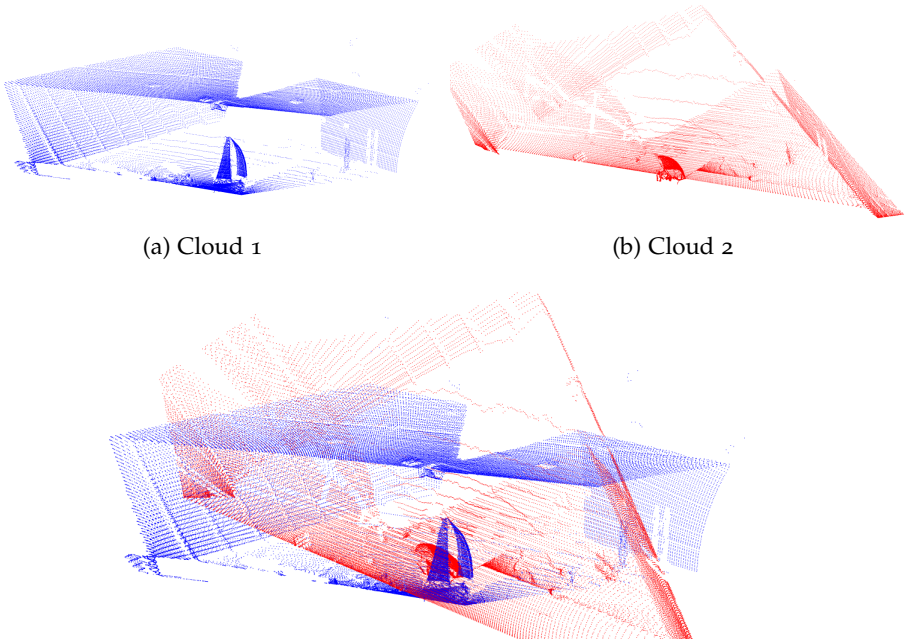
Modified algorithms, considering not only 3D spatial information, have been implemented: Johnson et al.[34] takes into account the colors acquire as well, Eggert et al. [35] performs the simultaneous registration of multiple range views, Declerck et al. [36] carries out 3D-2D projective transformations. Ristic et al. [37] instead proposes to use a triangular mesh model approximation to accelerate the algorithm and suggests to assume the standard uncertainty of the device used to perform the measurement as tolerance to terminate the ICP iterations.
Whichever the approach considered, the ICP performs the steps in Algorithm 1: Besl et al., however, proved that ICP algorithm converges always to a local minimum, it means that the algorithm is able to

Algorithmus 1 : ICP Algorithm

1. Identification of the set Q of closest points between the source data-set S_d and the target data-set T_d
2. Identification of the transformation matrices (R and T), minimizing the chosen cost function over Q
3. Application of the transformation to the measured points set as $S_d^{'} = R * S_d + T$
4. If change in the cost function is greater than a preset value the procedure repeat from Point 1
5. Else exit

provide the optimal solution only if the cloud data-sets are close to each other. Clouds that start the procedure presenting very different spatial positioning are not likely to be correctly registered, while on the other hand, ICP can achieve fine accuracy, when there is only a little displacement between the two data-sets. This is due to the fact that the algorithm estimates the solution considering all the points in the moving cloud. If the clouds are close to each other, a large number of right correspondences can be set, leading to a very accurate registration - measurement uncertainty averaged over thousands of points-. On the contrary, if the clouds are positioned far from each other, a large number of wrong correspondences could be set, leading to a complete failure in the registration process (refer to Appendix A.0.0.2, where different tests onto artificial data confirm the above).
Unfortunately, this is the most likely situation to be encountered for the application considered. Each acquisition unit - with its proper coordinate reference system - scans a single sail, which means that for a complete reconstruction of the sail plan a registration between two clouds is required.
The scanners are placed properly in order to avoid occlusion, but this leads to a poor overlapping scanned area. A simple test has been conducted on raw acquisitions. The registration has been tried with the ICP, since it has proven to be the algorithm with the best trade-off between time elapsed and accuracy. It fails in registering the raw clouds as shown in Figure 5.5: the procedure is not able to converge, even with a high number of iterations (more than 500 iterations).

To overcome the problem a proper algorithm has been developed and presented in the next Section.



(a) Cloud 1

(b) Cloud 2

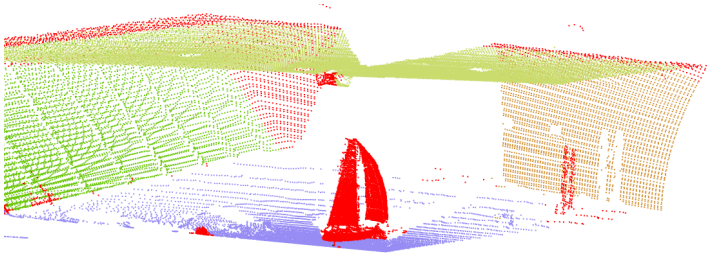
(c) Cloud 2 wrongly registered over Cloud 1

Figure 5.5: ICP Registration of wind tunnel acquisition.

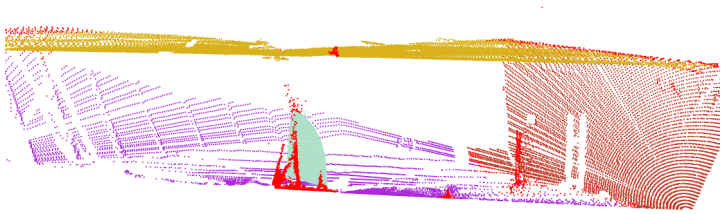
5.3.3 *Proposed registration strategy*

A custom-made procedure has been proposed to achieve a correct registration of acquisitions taken both during the wind tunnel and full scale tests.
The first step is to compute an initial alignment and then refine it by performing few iterations of the ICP algorithm, until reaching a desired level of accuracy (close to the measurement uncertainty). The rough alignment is not done through SVD approach, since acquisition noise and human errors in picking and matching few points of the clouds impair the result as shown in Figure 5.4a, and even adding ICP refinement the clouds would not match correctly.
The idea is to exploit the knowledge of some geometrical features describing the whole scanned environment. In particular, the algorithm extracts planar surfaces from each cloud and computes the spatial pose of each plane. The cloud to be registered is rotated and moved so that its detected surfaces overlap the corresponding reference cloud ones. Three planar surfaces for each cloud are sufficient: two pairs of planes are used for retrieving the rotation matrix, while translation vector is estimated computing the euclidean distance between the points of intersection of each triple of planes. A possible alternative for the translation estimation is to directly provide the coordinates of two matching points and compute the euclidean distance between them. In the first case, the origin of the common reference system corresponds to the origin of the reference cloud system; in the second

case, the origin of the common reference system can be chosen by the user (for example selecting the points representing the yacht mast foot). This strategy reduces the user interaction with respect to the SVD approach since planar surfaces can be automatically detected using standard algorithms for identification of primitive geometries, and limits at most the influence of the measurement uncertainty since it computes the transformation matrix based on the orientation of the planes that best fit thousands of points.
The procedure is described in Algorithm 2. The first step is the detection of planes in the scene, and Region Growing Clustering and RANdom SAmple Consensus (RANSAC) paradigm have been used (see Section 5.4). In particular, Region Growing algorithm splits the scene into clusters of points characterized by having uniform curvature (see Appendix A for definition of point cloud curvature); and RANSAC algorithm evaluate whether a cluster can be modeled as a sampled planar surface. It also provides the equation of the plane that fits the points of the cluster the best along with its associated normal vector. Examples of Region Growing segmentation and plane identification is shown in the Figures 5.6a and 5.6b representing clouds acquired during a scale model campaign: walls of the wind tunnel chamber are used to align the clouds coming from two acquisition units.



(a) Reference cloud - Identification of planar surfaces



(b) Moving cloud - Identification of planar surfaces

Figure 5.6: Registration procedure - Step 1 and 2: identification of planar surfaces (walls and ceiling).

Once correspondent planes are flagged and poses computed, the rotation matrix representing the relative orientation between them is

Algorithmus 2 : Proposed registration procedure

1. Segmentation of the reference point cloud PC_r:
   a) Identification of *Plane* 1_r and *Plane* 2_r
   b) Computation of spatial poses for *Plane* 1_r and *Plane* 2_r
2. Segmentation of the moving point cloud PC_m:
   a) Identification of *Plane* 1_m
   b) Computation of spatial pose for *Plane* 1_m
3. First rotation
   a) Computation of first rotation matrix R1
   b) Application of R1 to the moving cloud $\rightarrow$ *Plane* 1_m aligned to *Plane* 1_r, new cloud PC'_m
4. Segmentation of the new moving point cloud PC'_m
   a) Identification of *Plane* $2'_m$
   b) Computation of spatial pose for *Plane* $2'_m$
5. Second rotation
   a) Computation of second rotation matrix R2
   b) Application of R2 to the new moving cloud $\rightarrow$ *Plane* $2'_m$ aligned to *Plane* 2_r, new cloud PC''_m
6. Translation:
   a) Option 1: Extraction of a pair of matching points from reference P_r and moving point cloud P''_m
   b) Option 2:
      i. Identification of *Plane* 3_r
      ii. Computation of spatial pose for *Plane* 3_r
      iii. Identification of *Plane* $3''_m$
      iv. Computation of spatial pose for *Plane* $3''_m$
      v. Computation of the intersection points P_r and P''_m of the triples of planes
   c) Computation of the euclidean distance P_r - P''_m $\rightarrow$ translation vector T
7. ICP refinement:
   a) Filtering of the clouds removing non-overlapping areas
   b) ICP iterations until reaching the desired accuracy level and storing of resulting transformation matrix RT

calculated. Given two vectors, $v_1$and v_2 , the transformation matrix, R, to rotate one onto the other is:

$$v = v_1 \times v_2 \tag{5.9}$$

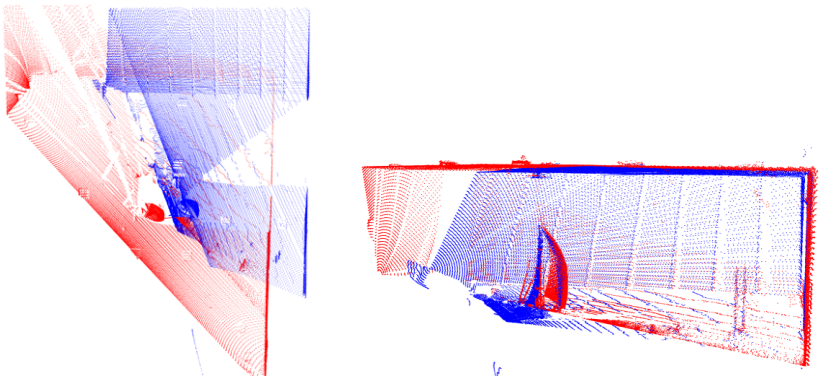
$$c = v_1 \cdot v_2 \tag{5.10}$$

$$R = I + [v]_x + [v]_x \frac{1-c}{\|v\|^2} \tag{5.11}$$

where $[v]_x$ is the skew-symmetric cross-product matrix of v:

$$[v]_x = \begin{bmatrix} 0 & -v_3 & v_2 \\ v_3 & 0 & -v_1 \\ -v_2 & v_1 & 0 \end{bmatrix} \tag{5.12}$$

The cross-product v between $v_1$and v_2 can be interpreted as the rotation axes, while the dot-product is the cosine of the angle that has to be applied. Figure 5.7a shows the result of the first rotation, which aligns the walls, and Figure 5.7b shows the rotation with respect to the ceilings. The order of the transformations is not important, this means that the procedure returns the same output, when aligning first the walls and then the ceilings or viceversa. At this point, the



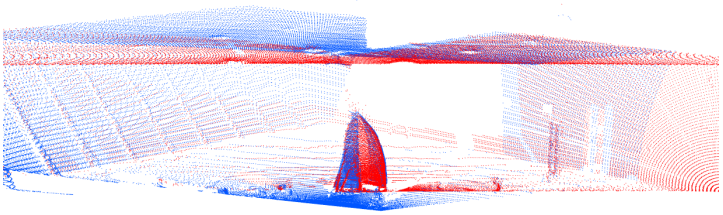
(a) First rotation: walls aligned (top view)

(b) Second rotation: ceilings aligned (lateral view)

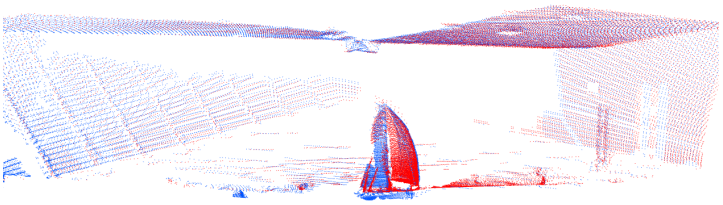
Figure 5.7: Registration procedure - Step 3-5: cloud alignment.

user selects a pair of corresponding points within the two clouds, euclidean distance is computed and translation is performed (see Figure 5.8a). Then, ICP iterations should be performed, to refine the alignment. The goodness of the final transformation is evaluated through a fitness score defined as the sum of squared distances between points in the reference cloud and points in the moving cloud. If its value is

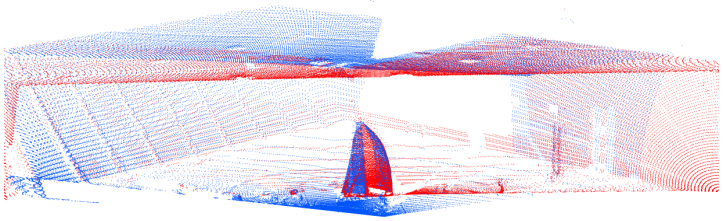
smaller than the measurement uncertainty found in Section 4.2, then the registration is considered well-performed, otherwise another iteration is suggested.
Before computing ICP iterations, the roughly aligned clouds have to be filtered. In fact, as explained in Section 5.3.2, ICP approach might not converge to the correct solution if a large number of correspondences are wrongly set. All the points belonging to non-overlapping scanned areas do not have any correspondent point in the other cloud. ICP, that uses information regarding all the points in the cloud in order to reduce the acquisition noise the most, would assign them a wrong correspondence, worsening the alignment.
To overcome this issue, a crossed filter has been implemented, starting from the classical spherical filter. Normally the spherical filter creates a sphere of fixed radius around each point. If the sphere does not contain other points from the same clouds (or less then a fixed number of points), the point is considered as an outlier and removed from the data-set. In this case the filter constructs the sphere around the point considered and checks whether it contains points belonging to the other data-set. If so, it keeps the point, otherwise it deletes it.
The results of the filtering are shown in Figure 5.8b. Note that, as expected, all the areas where the data do not overlap are removed, especially on the ceiling and in the background, but so is part of the sails. Thus, the radius of the sphere must be large enough to keep the majority of the sail point.
The final registration scene is showed in Figure 5.8c. The effect of the ICP algorithm might not be so evident looking at the whole clouds, but improvements, with respect to the rough alignment of Figure 5.8a, are highlighted in Figure 5.9, where close-ups of the clouds before and after the ICP are compared. The effectiveness of the ICP can be proven also through a statistical analysis of the distribution of minimum point to point distances between moving cloud and reference. Figure 5.10 shows the histogram of the distances before and after ICP alignment refinement. As visible, the post-ICP distribution presents a higher number of samples, more than doubled, in the low-distance zone (0 to 20 mm), while the pre-ICP state presents a steadier distance distribution. More than 50% of point distances are within 50 mm for post-ICP clouds, while only 30 % for the pre-ICP clouds.
These value might seem too high also for the post-ICP case but it must be kept in mind that the two scanners are not sampling the environment exactly in the same physical points, since they are placed in different positions and they present different TOF sensor angular resolution and scanning rate.
To evaluate the overall procedure accuracy, the reference cloud has been rotated by 90° around X axis, moved by 1000 mm in each positive axis direction and registered with the original cloud. The im-



(a) Translation (front view)



(b) Filtered cloud (only overlapping areas remain)



(c) Final registration (ICP refinement)

Figure 5.8: Registration procedure - Step 6 and 7: translation, filtering and ICP refinement.

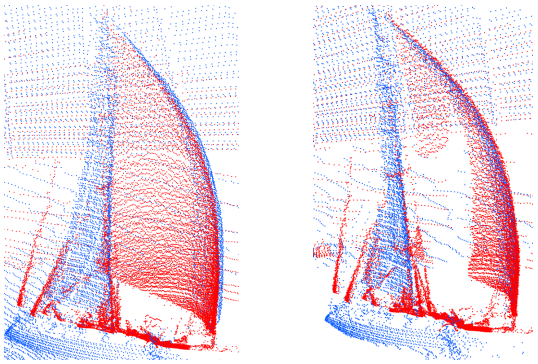
posed roto-translation matrix is compared to the matrix returned by the procedure in Equation 5.13.

$$\begin{bmatrix} 1 & 0 & 0 & 1000 \\ 0 & 0 & -1 & 1000 \\ 0 & 1 & 0 & 1000 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0.999 & 10^{-8} & 10^{-7} & 1000 \\ 10^{-7} & 10^{-7} & -0.999 & 999.97 \\ 10^{-7} & 1 & 10^{-7} & 1000.01 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.13}$$

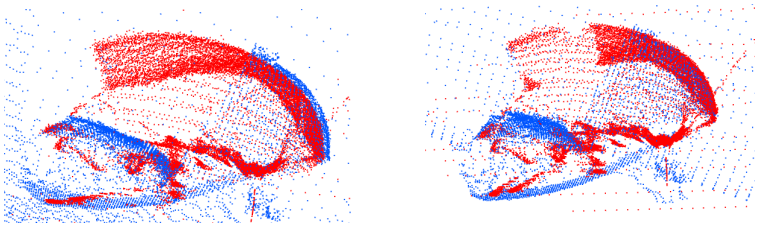
The registration is almost perfectly achieved and small discrepancies can be imputed to numerical approximation in the transformation implemented. Histogram of Figure 5.11 confirms the above, since all the distances are lower than the millimeter.

The procedure has been used also for full scale tests where the deck and the open skylight were used as planar surfaces for the clouds alignment. Figure 5.12 presents the results for the registration of two clouds representing a Comet 35 scanned from the stern (blue cloud) and from the heel of the mast (red cloud).

After achieving a good registration, the final transformation matrix



(a) Gennaker Sail - Side View



(b) Gennaker Sail - Top View

Figure 5.9: Registering point clouds details - pre-ICP (left) and after-ICP (right).

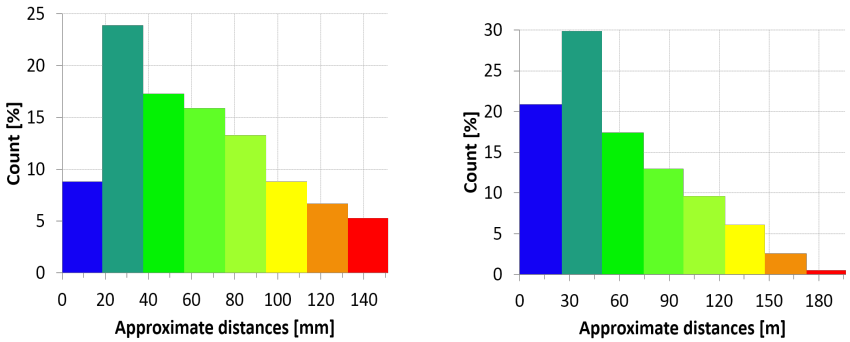


Figure 5.10: Statistical point to point distance [mm] before (left) and after (right) performing ICP.

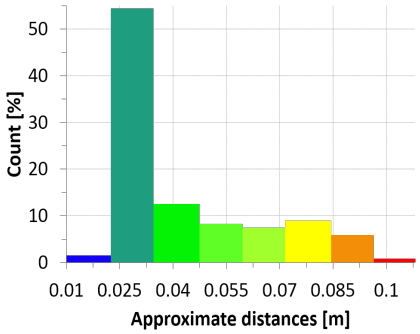
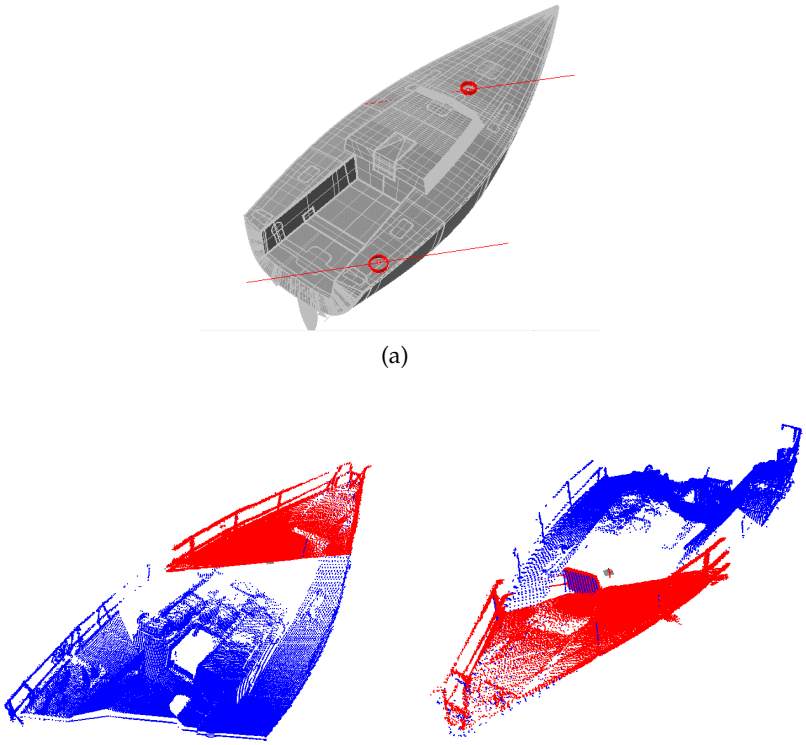


Figure 5.11: Statistical point to point distance [mm] after registration of transformed cloud over the original one.



(a) (b) (c)

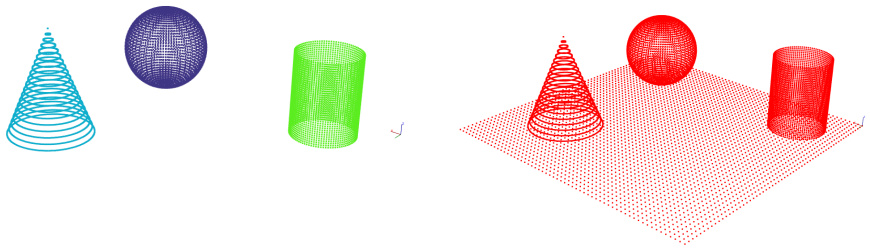
Figure 5.12: Registration of full scale acquisition of a yacht deck - Yacht CAD model and highlighted TOF scanner positions in red (a), results of two clouds (red and blue) registration (b-c).

is saved to a .txt file in order to be applied to all the other acquisitions taken during the wind tunnel campaign saving time in the complete data elaboration pipeline. This can be possible , of course, under the assumption that TOF units were not moved within the first scan and the following ones.

5.4 POINT CLOUD SEGMENTATION

In order to analyze and reconstruct the sail shape, the scanned scene has to be interpreted, and points belonging to the sail surfaces have to be extracted from the whole point cloud.
Segmentation and clustering are described as the operation of dividing large data-sets in smaller, disjointed subsets. Handling chunks of data instead of the whole has many advantages, but the developing of these algorithms was first pushed by the need of reducing the processing time of further operations (triangulation for example). Segmentation and clustering end up with almost the same result, although with a main distinctive difference, as segmentation supply additional information on the subsets returned, while clustering only

creates several possible data-sets. Segmentation returns extra information since it searches in the data set pre-determined models, such as spheres, planes or cones and it can identify the parameters of the model, like the radius of the sphere. On the other hand, clustering divides the original data in subsets starting from its local properties, so it could be run without the knowledge on what shape has to be searched. Since the difference between the output of the two procedures is so subtle that in many situations they can easily be used interchangeably, this work refers to *segmentation* as a general procedure that extracts a set of points from a wider data-set.
Segmentation is a key process in computer vision. Most of the algorithms developed were initially applied on 2D data coming from cameras and later, with the spread of 3D sensors such as laser scanners or depth cameras, adapted to point clouds.
Initial attempts have been carried out considering the euclidean distance as discriminating criterion; later normals and curvature information have been added leading to a more intuitive scene decomposition.
As mentioned above, the Euclidean Cluster Extraction algorithm is the simplest method for scene segmentation. As the name suggests, it relies only on the euclidean distance to populate the clusters: given two points belonging to the data-set, they are assigned to the same cluster only if their distance is smaller then a given threshold. As one may imagine, this kind of clustering works fine whenever an object has to be separate from the background or from other distant objects, but it is not able to split adjacent objects in different clusters: for example in a scene composed by objects lying on the ground, points belonging to the object bases are close to points belonging to the ground and that means they could be associate to the same cluster as shown in Figure 5.13. This happened also while trying to separate



(a) (b)

Figure 5.13: Euclidean segmentation: objects are correctly separated (a), while adjacent objects are assigned to a unique cluster (b).

points belonging to the sail surface to points belonging to the mast, the boom or the deck. Thus, methods that take into account more geometrical information have been considered.

5.4.1 *RANdom SAmple Consensus paradigm*

In Section 5.3.3 planar surfaces such as the walls of the wind tunnel chamber have been easily extracted from the whole scanned scene using RANSAC algorithm. This algorithm, in fact, uses both information about the spatial coordinates of the points and their normals, that are computed implicitly within the algorithm. Looking for the walls meant looking for all the points close to each other and presenting similar normal vectors.
In particular, the algorithm starts from the assumption that in the data considered there is a subset of points that can be explained through a mathematical model. Data can be divided in inliers, i.e. points that well suit the considered model, accepting a certain tolerance, and outliers, i.e. points that do not suit it.
The original version was developed by Fischler et al. in [38], where straight lines had to be detected. Later, other authors improved the procedure including the research of planar geometries such as circles and ellpises; and finally, Schnabel et al. in [39] published the first three-dimensional implementation enabling the identification of planes, spheres, cylinders and cones.
Despite the fact that yacht sails could not be represented perfectly by these regular geometries, RANSAC algorithm was taken into consideration since a level of tolerance could be set by the user and flat sails such as mainsails and jibs could be approximate as partial cones or cylinders.
The procedure steps are reported in detail in Algorithm 3 but can be divided in three main parts. In the first part, a subset containing the smallest number of points required to construct the model sought is extracted, and the model is effectively created to verify its consistency. Descriptions of the subsets required and the procedure to construct different geometric models used in this work can be found in [39] and reported in Appendix A.0.0.2.
Once created the model, in the second part, data points are tested and flagged as inliers or outliers with respect to the parameter ε representing the maximum discrepancy acceptable between a point and the model constrain, and to the parameter α representing the maximum angular discrepancy acceptable between point and model normal directions. Repeating these steps leads to different identified shapes, each formed starting by different minimal subsets. The final part is the evaluation of the best shape extracted, i.e. the shape that provides the maximum number of inliers, and thus, the highest score function. The score function $\sigma(\psi)$ is responsible for measuring the quality of a given shape candidate and is defined as follow:

$$\sigma(\psi) = |P_\psi| \tag{5.14}$$

Algorithmus 3 : RANSAC paradigm

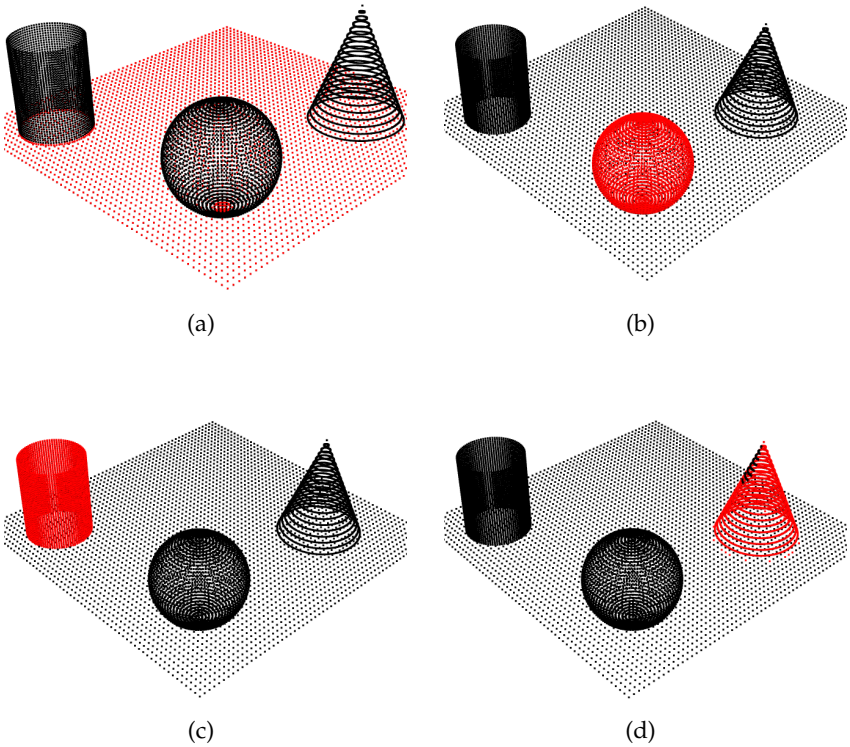
1. Computation of an octree representation for the input point cloud data-set P
2. Extraction of the minimal point set S to create the model
3. Creation of the candidate shape ψ from the points in S and computation of the shape's parameters (i.e. radius, planarity...)
   a) Identification of inliers or outliers points from the point cloud P according to constrains on:
      i. euclidean distance between point and candidate shape surface smaller than threshold ε
      ii. normal variation between adjacent point smaller than threshold α degree
   b) Evaluation of the score function $\sigma(\psi)$
   c) Re-computation of the shape parameters starting from all the inliers identified
4. Repetition of step 2 and 3 for a fixed number of times to extract many candidate shapes and their score functions
5. Rejection of candidate shapes with too few points
6. Extraction of points of P belonging to the shape with the highest score function

i.e. it correspond to the number of points of P_ψ, that represent the cluster of points satisfying the constrained mentioned above and thus defined as:

$$\hat{P}_\psi = \{p | p \in P \wedge d(\psi, p) < \varepsilon \wedge \arccos(|n(p) \cdot n(\psi, p)|) < \alpha\} \quad (5.15)$$

$$P_\psi = maxcomponent(\psi, \hat{P_\psi})$$

where d(ψ, p) is the euclidian distance between the point p and the shape primitive ψ, n(p) is the normal in p and n(ψ, p) is the normal of ψ in p's projection on ψ. *maxcomponent*(ψ,P^ψ) extracts the group of points in P^ψ whose projections onto ψ belong to the largest connected component on ψ.
Several preliminary tests have been conducted onto artificial data to get more confident to the algorithm. Influence of threshold parameters and influence of noise data have been considered. For the sake of brevity, only few of them are reported here. Table 5.1 and Figure 5.14 shows a correct identification of different geometrical objects in an artificial scene. Points highlighted in red have been associated to the geometric model sought.



(a) (b)

(c) (d)

Figure 5.14: Basic elements correctly identified by RANSAC paradigm: plane (a), sphere (b), cylinder (c), cone (d).

In this case the smaller the parameter ε, the more accurate the segmentation is. To confirm that, Figure 5.15 presents the same scene but wrongly segmented since ε was set an order of magnitude greater than before. Thus, the plane is considered by the procedure as a sphere of a radius tending to infinite. However, the ε threshold has to be increased in all the cases in which noise is added to the sampled points in order to take into account the measurement uncertainty (see Figure 5.16). Finally, tests onto scanned sails have been performed. RANSAC algorithm was run first considering a cylinder as the model

Shape	Model points	Identified points	% of identified points
Plane	3600	4140	85
Sphere	8100	8105	99.94
Cylinder	3600	3608	99.78
Cone	3620	3301	91.71

Table 5.1: Segmentation of basics elements with RANSAC paradigm.

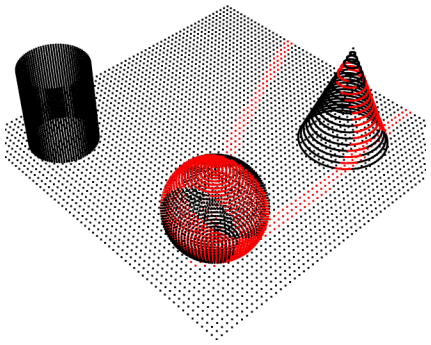
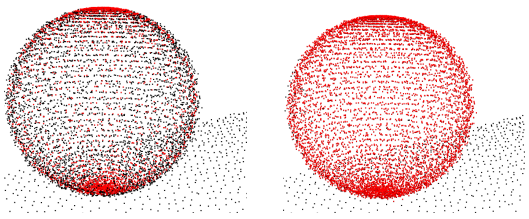


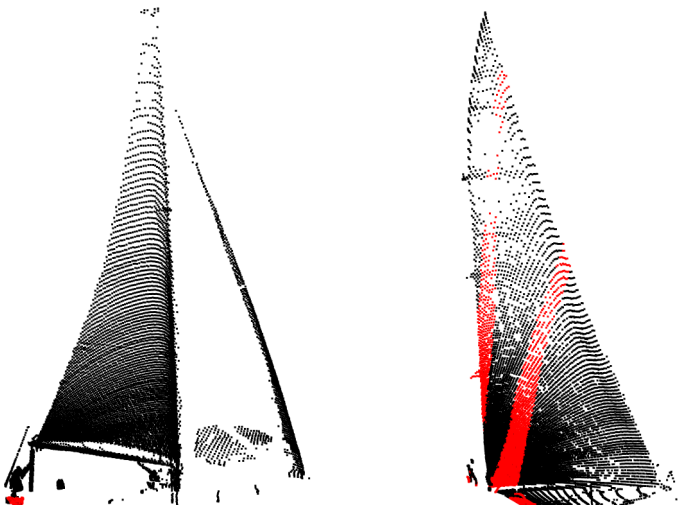
Figure 5.15: Failed segmentation: ε threshold too high.



(a) (b)

Figure 5.16: ε effect onto noisy data: percentage of sphere points identified is 70% for a low distance tolerance (a), while increases up to 95% for higher threshold values (b).

to be sought and addressed it especially to mainsails. Then a cone model was considered and efforts were addressed especially to approximate jib sails. However in both cases the results are completely unsatisfying as visible in Figure 5.17 and in Figure 5.18, and reported in Tables 5.2 and 5.3. RANSAC approach failed in identifying the

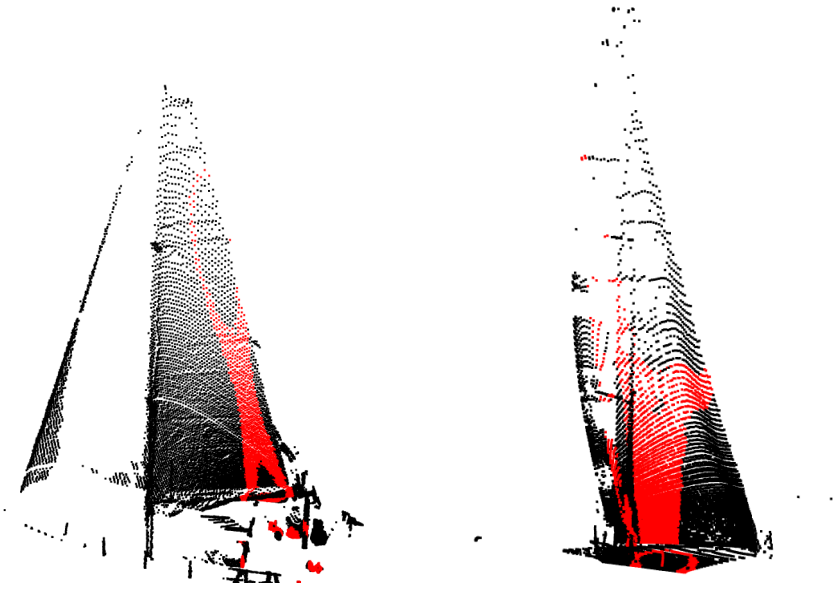


(a) Mainsail (b) Jib

Figure 5.17: RANSAC segmentation of full scale mainsail (left) and jib (right) - Cylinder model, ε= 60 mm.

Cylinder model	Sail points	% of identified points
Mainsail	7054	0
Jib	11799	28.86

Table 5.2: Segmentation of full scale sails through RANSAC algorithm - Cylinder model.



(a) Mainsail (b) Jib

Figure 5.18: RANSAC segmentation of full scale mainsail (left) and jib (right) - Cone model, ε= 60 mm.

Cone model	Sail points	% of identified points
Mainsail	6537	21.65
Jib	8837	43.7

Table 5.3: Segmentation of full scale sails through RANSAC algorithm - Cone model.

sail surfaces for all the cases tested. The main reason could be the fact that the sails present, in addition to the camber curvature, a twist contribution along the vertical direction that is not characteristic of a cylindrical or conical shape; so the model used is not totally suitable for the identification of such surfaces.
The conclusion is that RANSAC is a good choice for segmenting scenes composed of geometrical shapes as the ones of Section 5.3.3, or when a single object has to be decomposed in its principal parts (i.e.

identifying its faces) but another solution has to be found to extract the sail clusters.

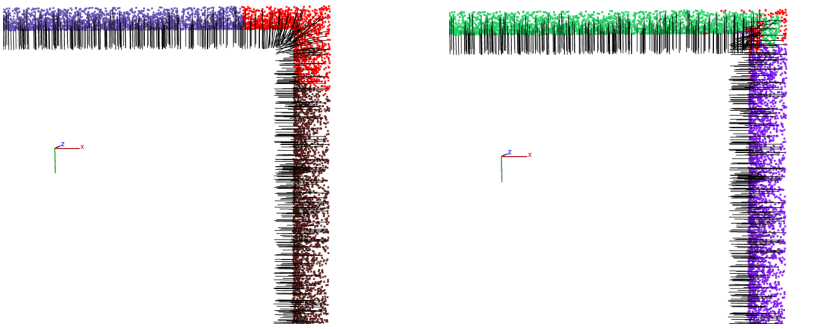
5.4.2 *Region Growing Clustering*

The region growing approach was developed initially for image segmentation, and states that pixels representing the same object must have similar characteristics, for example the same gray intensity level while considering gray-scale images. The algorithm requires an input set of seed pixels and, as deducible from its name, it proceeds growing regions with consistent properties starting from them [40].
The adaptation for the three dimensional space was implemented by Rusu in [41] as a "natural extension" of the Euclidean Cluster Extraction. His first intention was to identify and extract different surfaces colliding with each others, like a table lean on a wall, so he has chosen as the discriminating factor the normal of a point, which replaces the euclidean distance. Moreover, he chose to consider as seeds the points of the original data-set with minimum curvature. This means that the segmentation starts from flat areas and proceeds to aggregate as many points as possible, according to two constrains - explained in Algorithm 4-: a point is added to the cluster represented by a seed only if their normals do not diverge more than a threshold value ϑ_{th}, and if the point has a curvature lower then a curvature limit c_{th}. The Region Growing Clustering is computationally simpler than the RANSAC since no mathematical models have to be constructed. Moreover, the user can somehow control the algorithm performance through a series of parameters that can be set to adjust the cluster creation process according to the characteristics of the object sought. The principal parameters are: minimum and maximum number of points belonging to the cluster (to set the size of the object); and maximum point normal deviation and maximum acceptable surface curvature (to set the shape of the object).
Moreover, for normal computation, also the size of the neighborhood has to be properly set. In fact, as stressed in Appendix A, the normal vector associated to a point correspond to the vector perpendicular to the plane that best fit a certain number of neighboring points. For noisy data a larger number of neighbors is suggested. This would reduce the measurement uncertainty and, at the same time leads to a "smoother" cloud, and thus a smaller curvature threshold can be set to extract different objects in the scene. On the contrary, dealing with accurate scanning, normal computation can be done using a smaller neighboring size leading to sharper object edges, and thus higher value of normal variation can be set. Examples of the interaction between neighborhood size and normal variation threshold are reported in Figure 5.19 where two faces of a 90° corner are correctly separated. Another peculiarity of this algorithm is the possibility for a point to

Algorithmus 4 : Region Growing Clustering

1. create a region list $\mathbf{R}$, initially empty, an available points list $\mathbf{A}$, such that $\{\mathbf{A}\} \leftarrow \{1, ..., |\mathbf{P}|\}$ and a seed list $\mathbf{S}$;
2. while $\mathbf{A}$ is not empty:
   a) create a current region set $\mathbf{R_c}$;
   b) select the seed as the point with minimum curvature in $\mathbf{A}$ and add it in $\mathbf{S}$;
   c) for i=0 to size($\mathbf{S}$) do:
      i. find the nearest neighbors $\mathbf{B_c}$ of the current seed point $\mathbf{S_i}$;
      ii. for j=0 to size($\mathbf{B_c}$) do:
         A. for the current neighbor point $\mathbf{P_j} \leftarrow \mathbf{B_c}(j)$;
         B. if $\mathbf{P_j}$has still not be assigned ($\mathbf{P_j} \in \mathbf{A}$) and
            $$\cos^{-1}(|N\{\mathbf{S_i}\}, N\{\mathbf{P_j}\}|) < \theta_{th}$$
            then add the point to the region and remove it from the available points:
         C. if
            $$c(\mathbf{P_j}) < c_{th}$$
            then add the point to $\mathbf{S}$;
   d) if the current region $\mathbf{R_c}$ has too little points or too much points ($|\mathbf{R_c}| < N_{min} \wedge |\mathbf{R_c}| > N_{max}$) then is rejected, else
   e) return the current region $\mathbf{R_c}$ as cluster;
3. return the set of clusters.

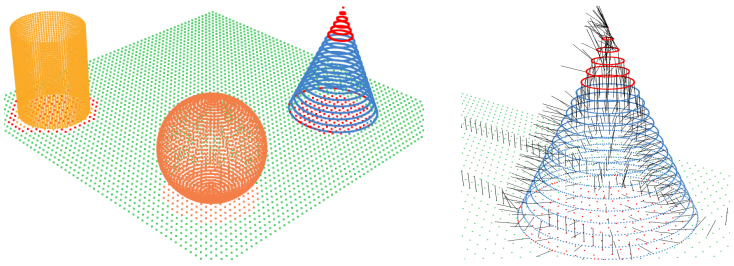
not be part of a cluster at the end of the steps, which happens when it is assigned to a cluster that is eventually rejected due to its cardinality. Also for this algorithm several tests have been conducted onto simplified scenes.
Unlike RANSAC algorithm, Region Growing approach produces many clusters in a single run, so it is possible to identify different objects in the scene. Figure 5.20 provides a visual representation of the output, and Table 5.4a quantitatively analyzes it. The sphere and the cylinder are well identified due to their constant curvature, unlike the cone that presents difficulties in normal estimation at its vertex as highlighted in Figure 5.20b. This approach yields to percentages of identified points lower than RANSAC method but it turned to be more robust to noise effect. Region Growing Clustering has been applied



(a) N_{min}=50, N_{max}=10000, c_{th}=1, ϑ_{th}=0.1, N_{neigh}=100

(b) N_{min}=50, N_{max}=10000, c_{th}=1, ϑ_{th}=4, N_{neigh}=10

Figure 5.19: Corner segmented with Region Growing, using two different approaches.



(a) (b)

Figure 5.20: Region Growing Segmentation for artificial scene - red points cannot be assigned to any cluster (a), normal estimation problem around the cone vertex (b).

also to acquisitions of full scale sails to verify its applicability to this research. Unlike the RANSAC, its results were absolutely positive: all the points belonging to the sail surfaces could be identified (see Figure 5.21 and Table 5.5a). Some false positive are also flagged as sail point such as few mast points or points representing the rigging. These points are considered as outliers and removed in the following elaboration steps or eventually deleted by the user. The same procedure, after another segmentation parameter tuning, worked fine also on acquisitions recorded in the wind tunnel. Figure 5.23 displays the segmentation performed on two wind tunnel acquisitions with the parameters reported in Table 5.6. The first data-set is dedicated to the gennaker acquisition, while the second to the mainsail. In conclusion, Region Growing Clustering has proved to be a successful algorithm for extracting the sail cluster from a whole acquisition. The procedure relies on the local information about point spatial coordinates, normals and curvature, and different threshold can be set to define the size and the geometry of the object sought. Segmentation, however might not be perfectly performed in case of smooth transitions be-

Shape	Model points	Identified points	% of identified points
Plane	4000	3087	77.17
Sphere	8100	8273	97.86
Cylinder	3600	3654	98.5
Cone	3620	2497	68.75
Not Identified		1409	

(a) Results of the segmentation

c_{th}	ϑ_{th}	$\#_{neigh}$	N_{min}	N_{max}
0.1	20	200	1000	1000000

(b) Parameters set for segmentation

Table 5.4: Region Growing Segmentation for artificial scene.

	# of original points	% of correctly identified points
Mainsail	7054	89.58
Jib	8837	97.17

(a) Results

	c_{th}	ϑ_{th}	$\#_{neigh}$	N_{min}	N_{max}
Mainsail	0.025	15	20	5000	100000
Jib	0.01	5	50	6000	100000

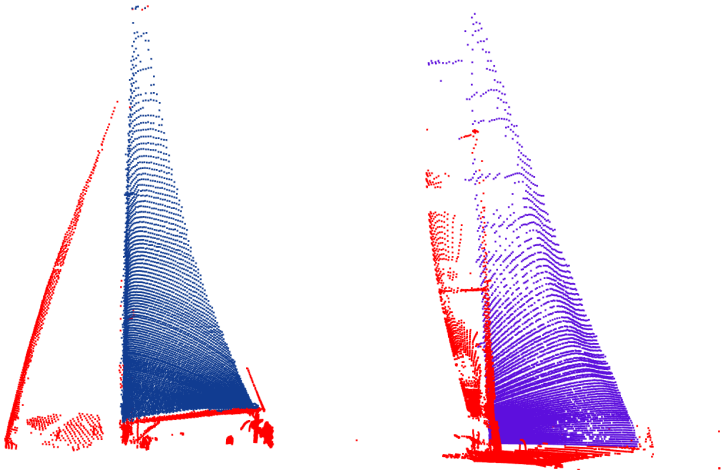
(b) Parameters

Table 5.5: Region Growing Segmentation performed on on-field sails acquisitions.

tween object surfaces, that leads to the inclusion of extra points such as the ones belonging to spreaders and sheets. If outlier removal and surface smoothing steps (see the following section) are not sufficient in deleting these points, human intervention is required.

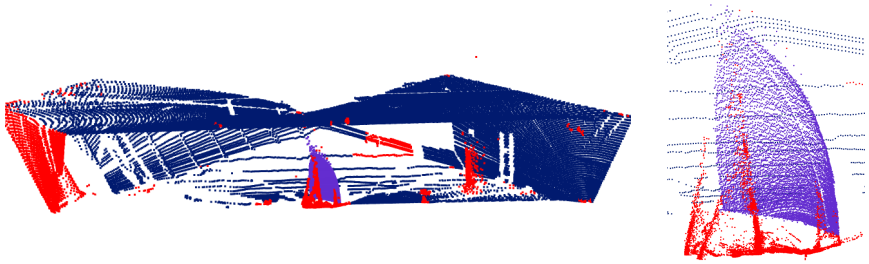
5.5 POINT CLOUD ENHANCEMENT

Data resamplings are procedures that upsample or downsample the general, or local, density of a point cloud, using averaging or interpolation of the original data-set. The need for resampling an already acquired geometry comes directly from the generation of the data:



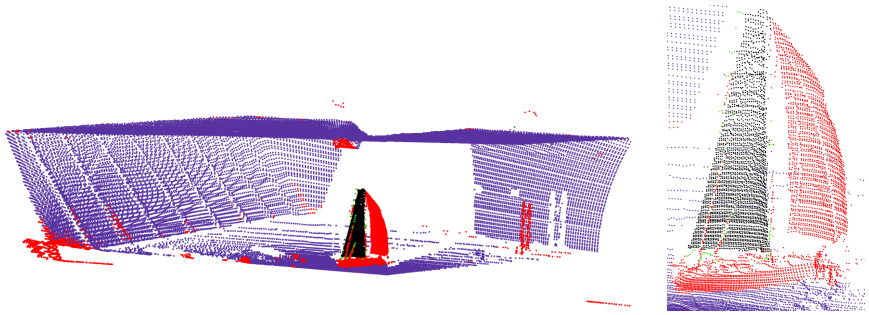
(a) Mainsail (b) Jib

Figure 5.21: Region Growing segmentation for full scale sails acquisitions - mainsail (a) and jib (b).



(a) (b)

Figure 5.22: Wind tunnel gennaker cluster extraction - full scan (a), and sail detail (b).



(a) (b)

Figure 5.23: Wind tunnel mainsail cluster extraction - full scan (a), and sail detail (b).

they are sampled from a real surface and inevitably prone to noise inaccuracy.
This category of techniques is mostly used for its properties of noise

	# of original points	% of correctly identified points
Gennaker	9615	98.8
Mainsail	2585	97.1

(a) Results

	c_{th}	ϑ_{th}	$\#_{neigh}$	N_{min}	N_{max}
Gennaker	0.9	7	50	5000	100000
Mainsail	0.8	7	50	2200	100000

(b) Parameters

Table 5.6: Wind tunnel segmentation

filtering and surface smoothing. In cases where the target presents a smooth surface, this feature can easily be lost in the acquisition process, because the presence of noise could generate spikes or holes. Smoothness has to be recreated through these postprocessing techniques.
Performing a data resampling can also be useful in removing local imperfections. These defects might occur for different reasons such as variable cloud density, overlapping areas due to error in registration of pairs of clouds, surface properties that do not allow for distance measurements causing holes in the cloud and sharp edges that result in mixed pixel problems. Some of these irregularities can not be easily deleted through traditional filtering, such as radius or statistical outlier removal but should be taken into account since they might compromise the data post-processing.
Standard resampling methods made their way into robotics applications from the computer graphics research community [42], and nowadays are commonly formulated as Moving Least Squares (MLS) solutions [43]. This approach is motivated by differential geometry and aims at minimizing the geometric error of the approximation. This is done by locally approximating the surface with polynomials using moving least squares. Given a set of points $P = \{p_i\}$, a smooth surface S_p is defined, representing the input points. The points P are reduced, defining S_p with a reduced set $R = \{r_i\}$ and introducing another MLS surface S_r which approximates S_p. The general paradigm is illustrated for 2D case in Figure 5.24. Points $p_i \in P$ are depicted in purple and define a curve S_p, in purple too (Figure 5.24a). S_p is then re-sampled with red points $r_i \in S_p$(Figure 5.24b). This typically lighter point set is called the "representation points" and defines the red curve S_r which approximates S_p. The key part of the procedure is creating the approximating surface (lines in 2D) for each data-set P or R, and can be divided into two steps: the identification of a reference

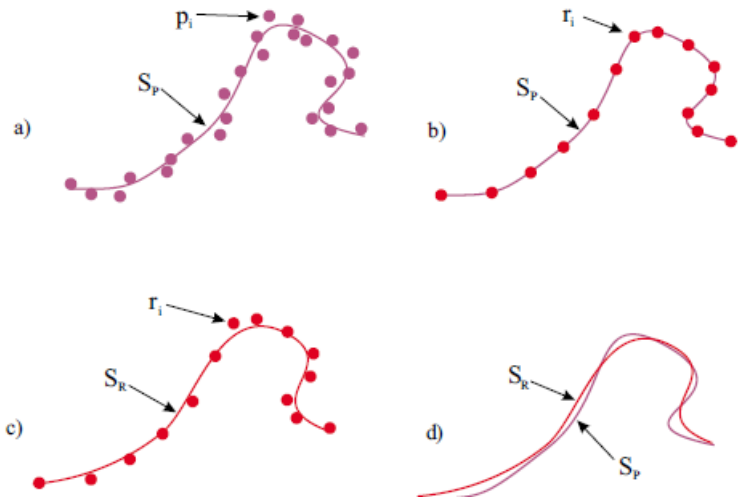


Figure 5.24: Resampling paradigm in 2D space - The original points p_i are approximated through a polynomial curve S_p(both in purple)(a). S_p is then resampled with a fixed point-to-point curvilinear distance, generating r_i(b). The polynomial approximation of r_i is S_r, which approximates also S_p(c-d).

domain and the effective computation of the approximating surfaces surface.
In the first step, the local plane domain H of a point r is constructed in order to minimize a local, weighted sum of square distances of the points p_i to the plane, where p_i are the neighbor points in the volume considered. It is done similarly to the creation of a best-fitting plane of the data-set, although the points are not equally weighted but weights are defined as the function of the distance of p_i to the projection of r on the plane H, rather than the distance to r (see Figure 5.25). Assuming q is the projection of r onto H, then H is found by locally minimizing:

$$\sum_{i=1}^{N} (< n, p_i > -D)^2 \, \theta \, (\|p_i - q\|) \tag{5.16}$$

where θ is a smooth, monotone decreasing function, which is positive on the whole space. The local reference domain is then given by an orthonormal coordinate system on H so that q is the origin of this system. In the second step the reference domain H for r is used to compute a local bivariate polynomial approximation of the surface, in a neighborhood of r. Let q_i be the projection of p_i onto H, and f_i the height of p_i over H (see Figure 5.25), i.e f_i = n (p_i-q). The polynomial approximation g is created, computing the coefficients that minimize the weighted least squares error:

$$\sum_{i=1}^{N} (g \, (x_i, y_i) - f_i)^2 \, \theta \, (\|p_i - q\|) \tag{5.17}$$

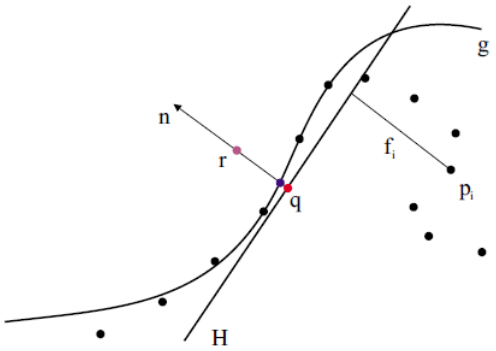
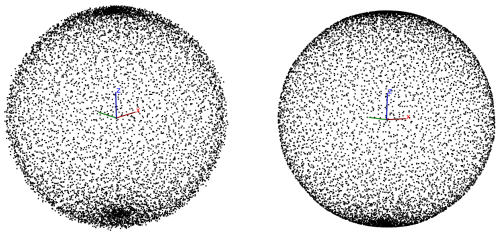


Figure 5.25: MLS Projection Procedure - First, a local reference domain H for the purple point r is generated. The projection of r onto H defines its origin q (the red point). Then, a local polynomial approximation g to the heights f_i of points p_i over H is computed. In both cases, the weight for each of the p_i is a function of the distance to q (the red point). The projection of r onto g (the blue point) is the result of the MLS projection procedure.

The projection P of the point r onto S_p, which is the returned output of the whole procedure, is found evaluating the polynomial value at the origin of the axes, i.e. $P(r) = q + g(0,0)n$. In the procedure, the only user definable term is the radial weight function θ, which is the parameter that influences the density of the output point cloud, because it determines the volume (or area in a two dimensional environment) considered for calculating the neighborhood of each point. The most widely used weight function was first proposed by Levin in [42],as a simple Gaussian function:

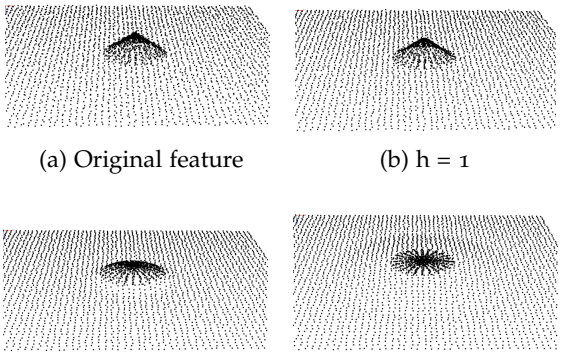
$$\theta(d) = e^{-\frac{d^2}{h^2}} \tag{5.18}$$

where h is the parameter reflecting the neighborhood size. A high value of h is suggested when acquisition noise is relevant and the user wants to reduce it (see the improvement in the noisy cloud test of Figure 5.26). But at the same time, large value of h could smooth out surface details (smaller than h) that might be of interest. More specifically, a small values for h cause the Gaussian weight function to decay faster, thus the approximation is more local. Conversely, large values for h result in a more global approximation, smoothing out sharp elements in the scene, as demonstrated in Figure 5.27. Considering the above, various tests have been carried out on real data acquired during wind tunnel tests and full scale tests in order to properly tune the h value. It must be large enough to reduce the acquisition noise, but, at the same time, it has to be upper limited not to overly smooth the surface altering the shape acquired.
Resampling tests have been evaluated comparing the processed cloud to the original one in terms of point displacement from their initial position. In particular, a point is labeled as "good point" if it is relocated



(a) Original Sphere (b) Smoothing h = 1

Figure 5.26: Effect of smoothing on a sphere of radius r = 4 a.u. and added noise of amplitude = 0.12 (a.u.).

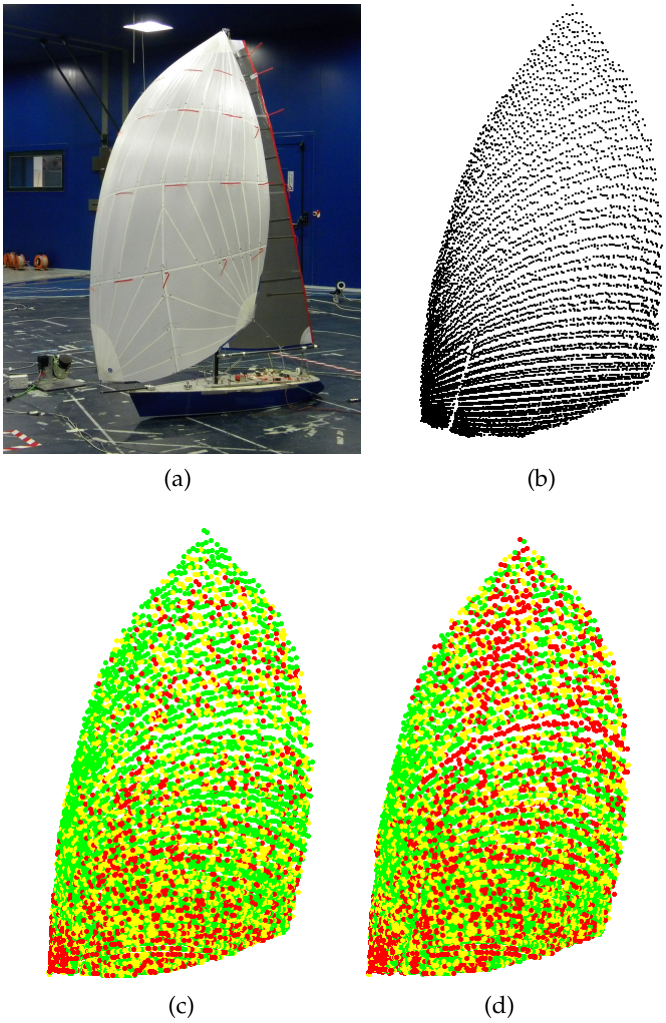


(a) Original feature (b) h = 1

(c) h = 5 (d) h = 10

Figure 5.27: Sharp feature erosion increasing h value - The original cone has radius equal to 3 (a.u.) and height equal to 2 (a.u.)

in a position whose distance from the original one is smaller than the measurement statistic error (it cloud be considered affected by noise). According to what stated in Section 4.2 as results of the metrological qualification of the TOF scanner -(Figure 4.14- a point is marked this way if is moved less than 4 mm. For greater displacements two situations might occur: the point is moved but still belongs to the actual sail surface, or it is moved so far that it could not be part of the sail surface anymore. The trade off value is estimated as the sum of statistical and systematic error that can be sought, once more, in Section 4.2. At a distance of 2 m (maximum height of scale model sail)the systematic error was estimated equal to 5 mm. Thus, the threshold value is set equal to 9 mm. Points moving more than the threshold are labeled as "bad points"; points whose displacement falls between 4 and 9 mm are considered "fair points". A fair point represents a point that is moved considerably but without altering the shape.
An example of results of the resampling process onto data acquired for a gennaker of a scale yacht model is presented in Figure 5.28 and Table 5.7. In Figure 5.28d, red points are more evident in the whole cluster with respect to Figure 5.28c. The edges of the sails seem to be almost untouched, which is good. A good resampling result de-



(a) (b)

(c) (d)

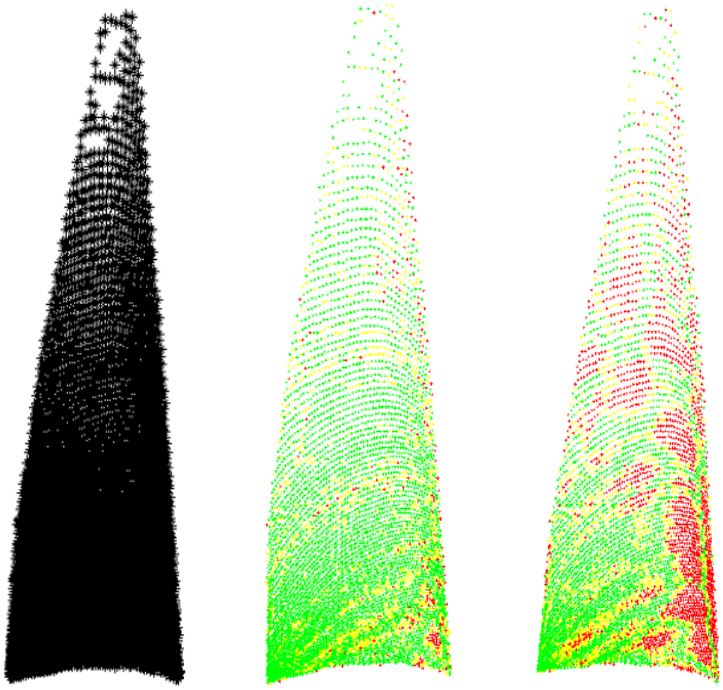
Figure 5.28: Scale model gennaker test - Setup photo (a), original cloud acquired (b), displacement color map for resampled cloud with h = 60 mm (c) and resampled cloud with h = 200 mm (d). Good points in green (small displacement), fair point in yellow and bad points in red (large displacement).

h =	60	80	100	200	400	1000
good points %	46.7	44.2	43.7	42.8	39.3	17.2
fair points %	44.2	44.3	42.5	43	43.4	26.2
bad points %	9.1	11.5	13.8	14.2	17.3	56.6

Table 5.7: Percentages of small (good) or large (bad) displacements in the resampled data-set for a gennaker scale model sail.

pend not only on the parameter set, but also from the input dataset. If the input data-set presents outliers, they might not be removed and moreover, they influence the surface approximation attracting points

towards them.
Similar observations can be made on acquisitions of real sails. Of course, the threshold values for "good" and "bad" points have to be adjusted, according to distance measured, i.e. the sail size. In the following case a mainsail is considered; its maximum height is 7 m. Good points are always defined as samples whose displacement is smaller than the noise threshold (4 mm), while the limit for bad points is moved up to 14 mm. Various h values have been considered, whose results are showed in Figure 5.29 and in Table 5.8. As visible in Fig-



(a) (b) (c)

Figure 5.29: Full scale mainsail point cloud - Displacement colormap for different h values: good points in green (small displacement), fair point in yellow and bad points in red (large displacement).

h =	175	225	275	350	500	750	1000
good points %	62.3	61.3	61.3	59	54.1	42.5	33.1
fair points %	37.7	38.7	38.68	40.9	45.3	50.5	48.5
bad points %	0	0	0.02	0.1	0.6	7	18.4

Table 5.8: Percentages of small (good) or large (bad) displacements in the resampled data-set for a full size mainsail.

ure 5.29, the most critical part for the mainsail is the luff edge. Bad points concentrate around the mast and setting a h value too high might lead to modify the luff shape. After several tries h = 500 mm

has been chosen as a suitable h level, but in most cases it depends also on the quality of the single acquisition.

5.6 SURFACE MODELING

Even if it is possible to implement some analysis algorithms taking the point cloud as input (generally suitably filtered), in many cases this kind of analysis shows itself to be insufficient for a complete shape analysis. In fact, it does not provide enough information about the shape topology and, moreover, it does not allow a clear idea on what the shape actually is. It is obvious that a proper visualization and surface reconstruction are fundamental to perform a correct shape analysis.
The problem can be states as follows: given a set of points P, which are sampled from a surface in R^3, construct a surface S, so that the points of P lie on S or in its proximity. From this definition follows that S is an interpolating surface with respect to P in the first case (explicit method), while S is an approximating surface in the second case (implicit method) [44].

5.6.1 *Piece-wise linear surface modeling*

Among the interpolation methods, the most commonly used is the piece-wise linear surface reconstruction. Piece-wise linear surface modeling algorithms lead to the construction of a polygonal mesh starting from the measured points. A polygonal mesh is a set of connected polygonally bounded planar surfaces [45]. It consists of edges, vertices and polygons connected such that each edge is shared by at most two polygons.
Polygonal meshes are used to model also curved surfaces: the consequent error can be arbitrarily reduced using smaller polygons. Yet this model improvement yields to an increment of the memory requirements and the algorithm execution time.
However, no improvement can be effective while dealing with noisy or non-uniform data.
Among the different polygonal meshes, the triangular ones are the most used (Figure 5.30), since they allow to express the topological properties of the surface with the best trade-off between numerical robustness, algorithmic simplicity and efficient display. The underlying, intuitive, reason is that a surface can be locally approximated as a plane, which, in turn, is identified by a minimum of 3 points. The use of triangular meshes is nowadays a standard in computer graphics applications and there exist several different algorithms to realize it. In [46] Mücke stated that, for a set of points P, a surface or volume

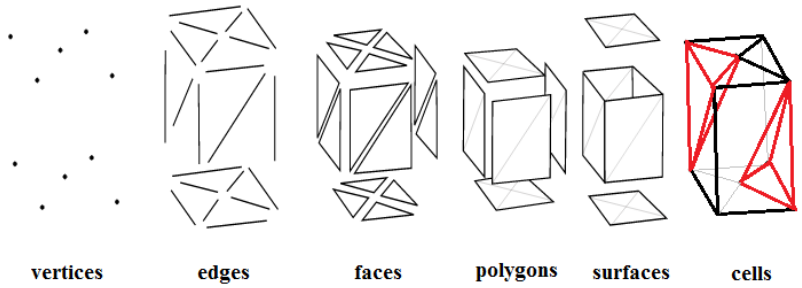


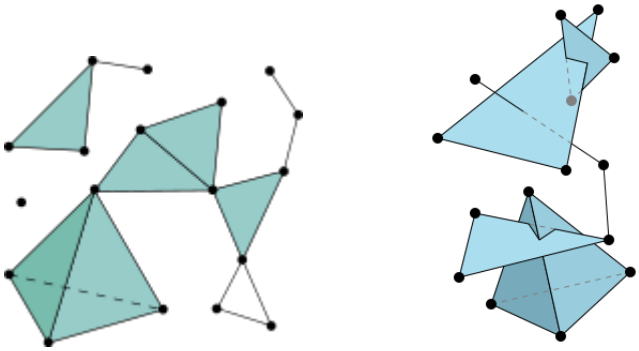
Figure 5.30: Linear piece-wise surface interpolation - Elements of a triangular mesh.

representation is possible through a k-simplex, denoted as σ_P, where k=|*P*|-1. For a set with a single point, the only representation possible is a 0-simplex, which is the point itself, or vertex; a two-point set can be divided in two vertices and a 1-simplex, which is an edge. A three points set forms a 2-simplex, or triangle and a four points set forms forms a 3-simplex, which is a tetrahedron, in addition to the ones said before. The tetrahedron is chosen as the smallest unit to describe a three-dimensional geometry, while k-simplices with k>3, which represent four-dimensional units, are not used, due to the significance lack to human visualization. Mücke also stated that a proper reconstruction is achieved when the set of all the simplicies, **C**, satisfy the following properties:

1. if $\sigma_P \in C$ then $\sigma_{P'} \in C$ for every $P' \subseteq P$. In other words, for every simplex σ_P, *C* contains all the faces of σ_P as well.

2. if σ_T, $\sigma_{P'} \in C$ then either $\sigma_P \bigcap \sigma_{P'} = 0$, or $\sigma_P \bigcap \sigma_{P'} = \sigma_{P \bigcap P'} =$ conv($P \bigcap P'$). In other words, the intersection of any two simplices in *C* is either empty or a face of both.

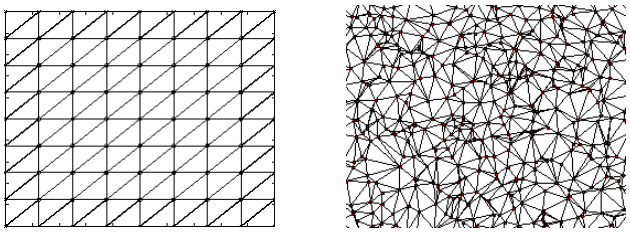
This properties define a set of simplies as a simplicial complex; as visible from Figure 5.31 non-simplicial complexes are unable to represent correctly a surface. A simplicial complex *C* is called a (geometric) triangulation of a of a set of points **S** if all vertices of *C* are points of *S* and |*C*|=conv(*S*). In other words in addition to the condition of the simplicial complex, the set of vertices of the simplices have to coincide with *S*.
The challenge for surface reconstruction algorithms is to find methods which deal with a wide variety of shape, whose output is, before all, a simplicial complex, and eventually is a good representation of the object considered. Once a proper triangulation is performed, evaluating its goodness is not an easy task. Because of its intrinsic visualization nature, it is much easier to judge the quality of a mesh for human beings rather than for a computer. Several check can be made on the geometric composition of the triangles [47]. It is common practice to avoid building "slim" triangles, which are shapes with the presence



(a) (b)

Figure 5.31: Simplicial complexes (a), and non-simplicial complexes (b).

of really small angles (less than 10 degrees). Additionally a mesh can be defined as regular, meaning that is composed of triangles similar one to the other. Refer to Figure 5.32 for explanation. Moreover, in this



(a) (b)

Figure 5.32: Regular Mesh (a) vs Irregular Mesh (b).

work, another criterion for admitting a mesh as acceptable, is that it does not have holes, since the goal is to reconstruct a continuous surface.
Some of the most renowned algorithms have been taken into consideration and results are presented in the next sections.

5.6.1.1 *Greedy projection*

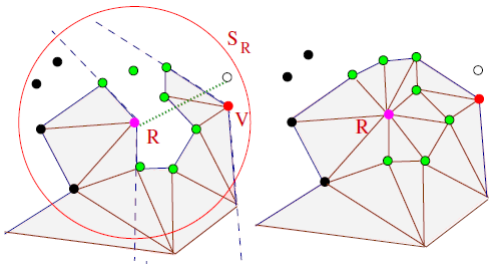
The Greedy Projection Algorithm, presented by Rusu et al. [41], starts from large, noisy and possible unorganized point cloud, recreating the underlying surface as a triangulation of the data-set, using an incremental method. It is based on the surface growing principle, following a greedy type approach. This means that the procedure starts creating a triangle and then keeps on adding new triangles until all points in the cloud have been considered or no more valid triangles can be joined to the existing mesh. The algorithm works with a three dimensional input but its output is composed of two dimensional triangular simplices, which form at least one surface, when possible. The creation of the mesh follows Algorithm 5. Step 1 defines the size

Algorithmus 5 : Greedy Projection Algorithm

For each point $p \in P$:

1. *Nearest neighbors search*: a r-neighborhood is selected by searching for the point nearest k neighbors in a sphere with radius $r = \mu \cdot d_0$ ($d_0$is the distance from the point p to its closest neighbor and μ is a user specified constant).
2. *Neighborhood projection*: the neighborhood is projected on a plane that is approximately tangential to the surface formed by the neighbors, which are then ordered around p.
3. *Pruning*: the neighbors are pruned by distance and visibility criterion.
4. *Triangulation*: creation of triangles. p is connected to two consecutive neighbors to form a triangle. Additional controls are made on the angles between the edges and on the edge length. Triangles are created iteratively until all points in the neighborhood belong to a triangle.

of the neighborhood to triangulate as a function of the cloud density (through parameter d_0) and a user set parameter μ. Step 2 is the reason why the output is a surface: even though the input are three dimensional points, the actual triangulation is performed after a projection on a local plane. Step 3 reports the pruning of the points made with two different methods: the distance criterion, and the visiblity criterion. Visibility criterion, applied as soon as part of the mesh is triangulated, removes all the points that do not have a "direct vision" to the reference point, as shown in Figure 5.33. When a triangle



(a)

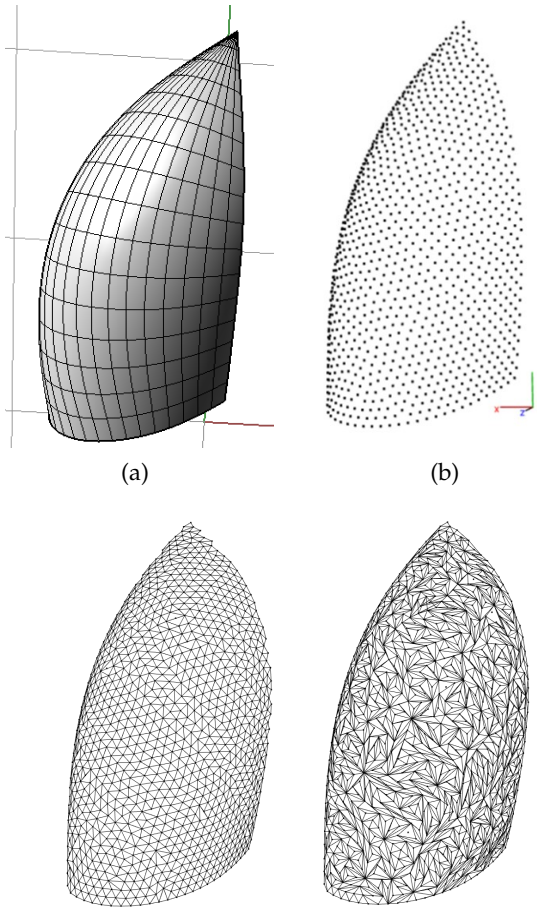
Figure 5.33: Visibility Criterion: before pruning (left) and resulting triangles (right).

is generated (Step 4), four conditions has to be verified to confirm it as part of the final mesh:

- *Minimum angle threshold*: the angles between each edge has to be greater than a user specified value.

- *Maximum angle threshold*: the angles between each edge has to be smaller than a user specified value.
- *Maximum surface angle*: the deviation between the normals of the vertex of the triangles has to be smaller than a user specified value.
- *Maximum edge length*: each edge of the triangle has to be smaller than a user specified value.

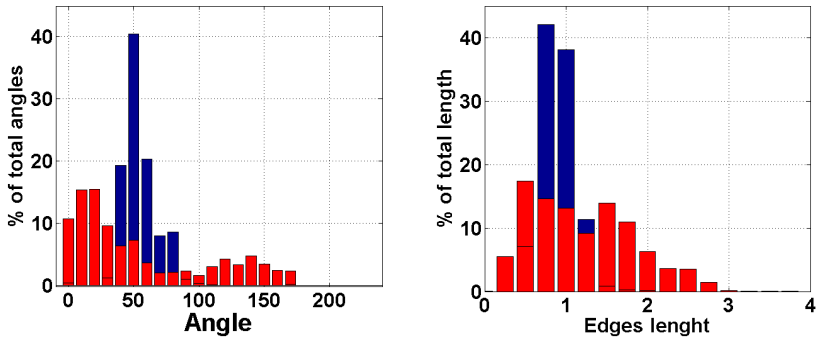
The first two conditions are introduced in order to eliminate small triangles, which could be the result of noisy data, and big triangles, which could be created due to the greedy approach of the procedure. Small triangles rise the numerical complexity of the mesh, without adding significant information, while on the other hand big triangles could lead to a bad approximation of the surface to be reconstructed. Aware of the above, tests have been performed first on synthetic sail shapes, and then on real sail acquisitions. Figure 5.34 presents the greedy triangulation for a cloud extracted sampling a CAD model of a gennaker sail given by the manufacturer. It is the most ideal acquisition case, since it has been uniformly sampled and, of course is not affected by acquisition noise. A first triangulation is performed, with a minimum neighborhood size that avoids holes, i.e setting $\mu = 1.75$. The triangles generated are mostly regular, but the shape edges are not reconstructed perfectly, as seen at the top of the sail. To achieve a better results for the sail edges, the neighborhood is expanded setting $\mu = 5$. The resulting triangulation is more irregular, even if sail edges have been improved. Their differences can be visualized through a comparison of their triangle edge length and angles as shown in Figure 5.35. As expected, in the first mesh (blue bars), triangles have angles mainly between 45° and 90°, while in the second one have more scattered values (red data). In addition, edge length analysis confirms the greater regularity of the first mesh with respect to the second one, in fact its triangles present edge length ratio centered around one, i.e. edges have the same length. Real data-sets are different from the previous case since point cloud density is not constant and acquisition noise influence the point sampled spatial positions. In particular, placing the TOF scanner unit on the ground of the wind tunnel or on the sail deck leads to a high density at the bottom of the sail (part closed to the acquisition unit) and sparser points at the top. Setting a neighborhood size for the entire scan yields to a mesh composed of small triangles at the bottom and bigger triangles on the top. Tests on real acquisitions are reported in Figure 5.36a for a gennaker acquired in the wind tunnel and in Figure 5.36c for a mainsail acquired on field. Even easing all the constrains about angles and triangle edge length, the triangulations created are not acceptable, due to the presence of holes highlighted in the shaded view mode. The parameters used for the creation of the meshes showed are displayed in Table 5.9. Since



(a) (b)

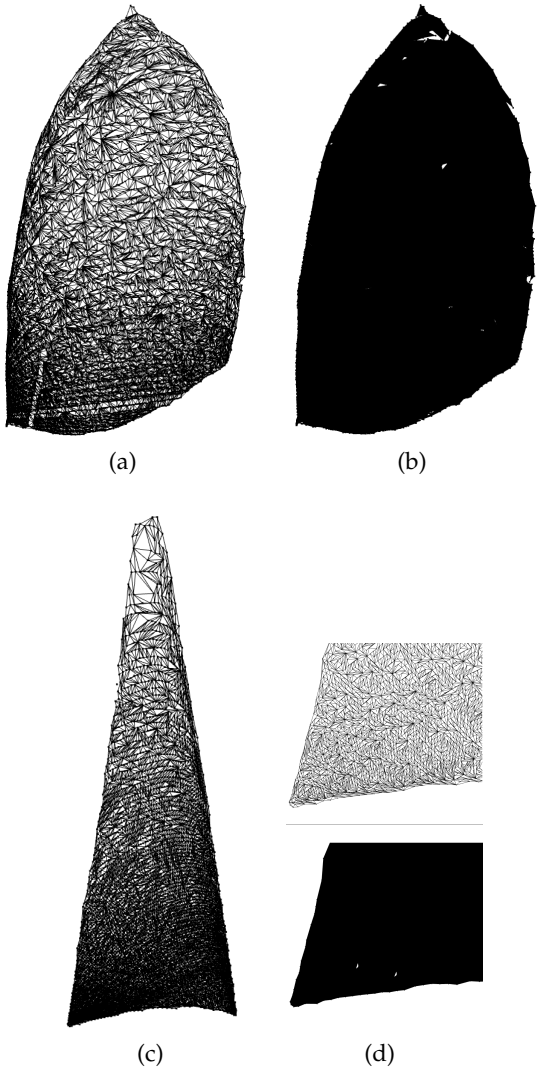
(c) (d)

Figure 5.34: Greedy Projection triangulation for a uniform density artificial sail cloud (a) with different neighborhood size: $\mu = 1.75$ (b), and $\mu = 5$(c).



(a) (b)

Figure 5.35: Sail model triangulation comparison - blue data refers to the regular mesh in Figure 5.34c, red data to irregular mesh in Figure 5.34d



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

Figure 5.36: Greedy Projection: Real acquisition triangulations for a scale gennaker (a-b) and a full scale mainsail(c-d).

	# Neigh.	μ	Max. Edge [a.u]	Min. Angle [°]	Max. Angle [°]	Max. $\Delta\overrightarrow{nor}$ [°]
Gennaker	50	7	300	30	150	20
Mainsail	50	8	400	45	150	20

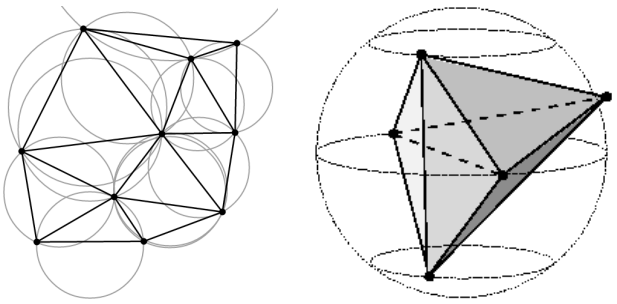
Table 5.9: Greedy Projection: Real acquisition triangulation parameters.

the absence of holes is the first prerequisite that the triangulation has to satisfy, these meshes are not acceptable. Changing the parameters does not lead to any improvement in triangulation and other tests on different acquisitions confirmed it. In conclusion this method is able to reconstruct a surface given a 3D data-set, and several parameters can be set to control the size and the shape of the surface approxi-

mating triangles. It performs the best when the data-set presents a uniform point density, and thus, it can be used for triangulate CAD model point cloud but it has tuned to be ineffective for real sail acquisitions.

5.6.1.2 *Delaunay triangulation*

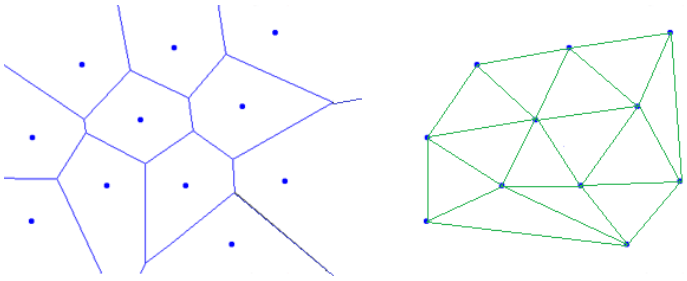
The Delaunay triangulation was invented by Boris Delaunay [48] and, for a set of points in the plane is a triangulation such that no triangle circumcircle contains a point belonging to that set. An interesting property of the Delaunay triangulation is the maximization of the minimum angle of the triangles that make up the grid: it tends to avoid sliver triangles. In three-dimensional spaces it is necessary to consider the circumscribed spheres in place of the circumferences.
From a mathematical point of view the Delaunay triangulation is the



(a) (b)

Figure 5.37: Delaunay Triangulation in 2D domain (a), and in 3D domain (b).

dual graph of the Voronoi tessellation [49], that is a special kind of decomposition of a metric space determined by distances to a specified discrete set of objects in the space, e.g. a discrete set of points. In the simplest and most common case, a set of points S in the plane, the Voronoi diagram for S is the partition of the plane which associates a region $V(p)$ with each point p from S in such a way that all points in $V(p)$ are closer to p than to any other point in S. Lots works in literature cover this topic and a large number of algorithms have been proposed. Refer to [50], [51], [52], [53], and [54] for a complete description. As done for the greedy projection algorithm, the performances of the Delaunay triangulation have been tested for artificial cloud and real acquisition of scale model and full scale sail. The native three dimensional Delaunay is applied to the cloud but the results of the triangulation are solids and not surfaces. In fact, if the algorithm receive as input a 3D point cloud, it creates tetrahedrals retrieving the data-set convex hull. Morever, no parameters can be set to control the triangulation process. Figure 5.39 shows the volumes created. Since for the study considered, a surface is sought, the Delaunay triangula-



(a) (b)

Figure 5.38: Voronoi diagram (a), and correspondent Delaunay triangulation (b).

(a) Sail model triangulation (b) Scale model gennaker triangulation (c) Full scale mainsail triangulation

Figure 5.39: 3D Delaunay triangulation - Creation of volumes and not surfaces.

tion has been tested onto the same data-set but after a projection onto a plane. The choice of the projection plane is crucial, in fact the points distribution must not be altered. This means that, for each sample p, its neighbors in 3D have to correspond to the its neighbors in 2D on the projection plane. Two different planes have been considered: 3D point best fitting plane and the plane described by point polar coordinates (see Section 4.1, Figure 4.2). Once the points are projected onto the plane, Delaunay trinagulation is performed. Connections between points are saved and recreated in the 3D space leading to a meshed surface. The main steps are described in Algorithm 6 and explained in Figure 5.40. Tests onto real acquisitions confirmed the applicability of this strategy for the surface reconstruction. Figures 5.41, 5.42, and 5.43 presents the triangulation results for artificial sail cloud, scale model gennaker acquisition and mainsail full scale acquisition. The long edge triangles, visible in Figure 5.43in the 2D mesh are easily filtered in the last step of the procedure.

Algorithmus 6 : Surface reconstruction strategy based on 2D Delaunay triangulation.

1. Projection of the input 3D data-set on the plane: p(x,y,z) → p(u,v);
2. 2D Delaunay triangulation (2D point connections);
3. Creation of a 3D mesh connecting 3D points whose corresponding 2D points have been connected in the plane;
4. Trimming triangles with edges longer than a threshold value.

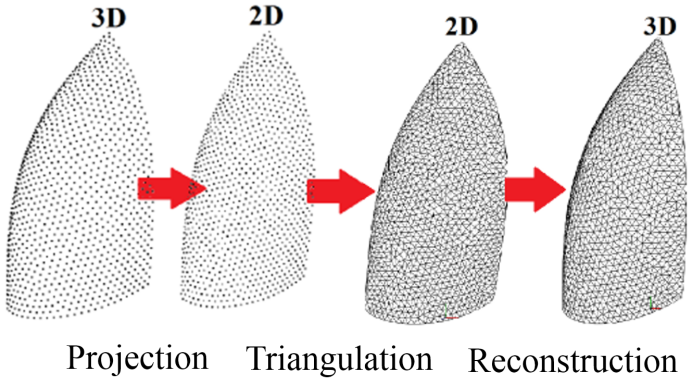


Figure 5.40: Surface reconstruction strategy based on 2D Delaunay triangulation.

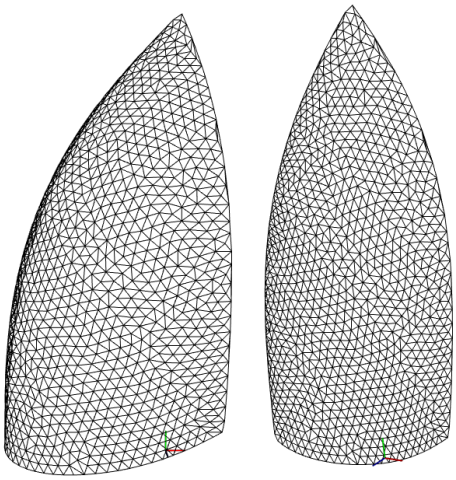
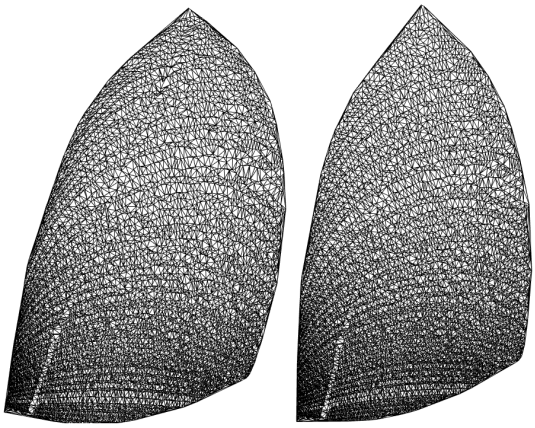
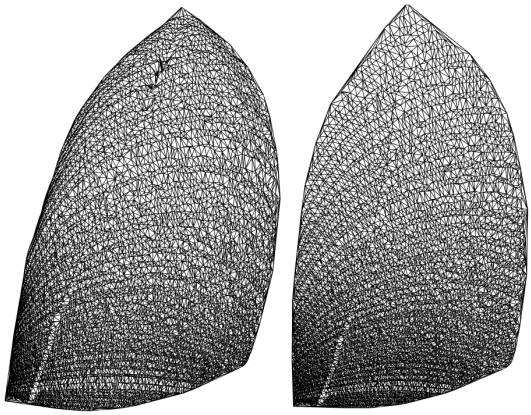


Figure 5.41: Artificial sail shape surface reconstruction - Projection onto point best fitting plane.

The choice between using the best fitting plane or the polar coordinates plane is arbitrary but for off-wind sails or for highly twisted sails a projection onto the best fitting plane might lead to region



(a) 3D and 2D meshes with best fit plane



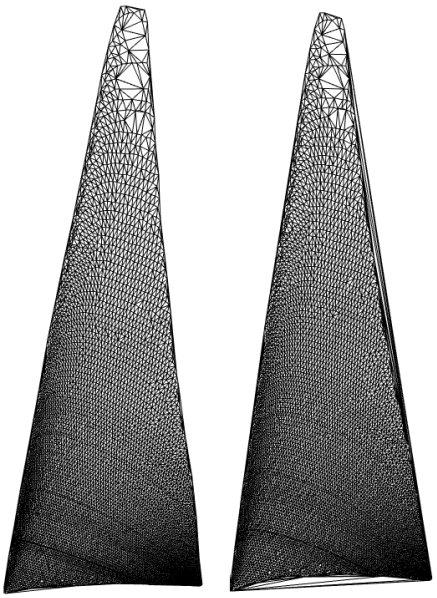
(b) 3D and 2D meshes with (α, γ) plane

Figure 5.42: Gennaker surface reconstruction - Projection onto point best fitting plane (a) and polar coordinates plane (b).

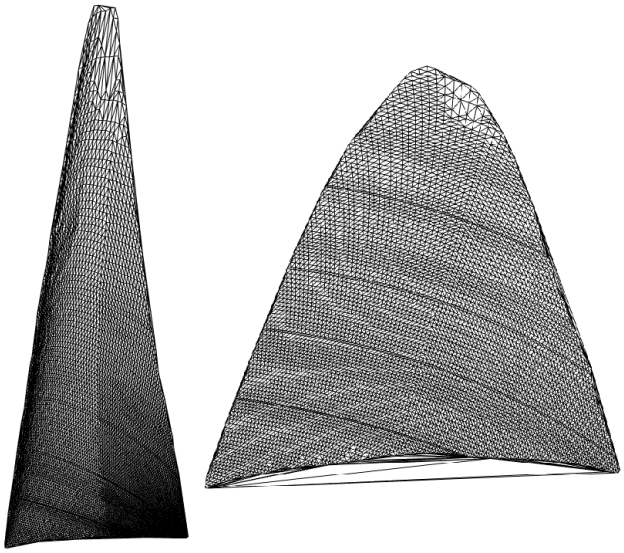
overlapping and then to a incorrect triangulation. As for the three-dimensional case, there is no parameter needed for this triangulation, the creation of the mesh is automatic. Moreover, the non-uniform density of the point cloud does not influence the triangulation process. This method thus could be used for a linear piece-wise surface reconstruction.

5.6.2 *Piece-wise smoothing surface reconstruction*

All the methods described so far realize surfaces that pass exactly through the sampled points. This could be an advantage in terms of reliability of the reconstruction, but sometimes it would not lead to the expected surface since they do not take into account the acquisition noise: originally smooth surfaces might appear rough due to the measurement uncertainty. An approximating surface fits the point with a certain tolerance overcoming this issue and providing a smooth surface.



(a) 3D and 2D meshes with best fit plane



(b) 3D and 2D meshes with (α, γ) plane

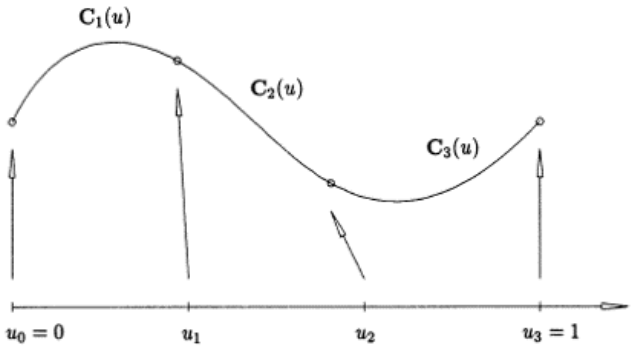
Figure 5.43: Mainsail surface reconstruction - Projection onto point best fitting plane (a) and polar coordinates plane (b).

Many surfaces can be represented analytically, but there are also many surfaces, typically free-form, for which analytical descriptions do not exist. These surfaces are represented in a piece-wise fashion, i.e. similar to a patchwork quilt [55]. The complete surface is obtained combining individual patches together along the edges and assuring the required continuity at the joins: a smooth piece-wise surface reconstruction.
Nowadays a common way to perform this kind of surface approximation is based on B-spline (Basis-spline) curves or surfaces as well as their abstraction to Non-Uniform Rational B-Splines (NURBS). They

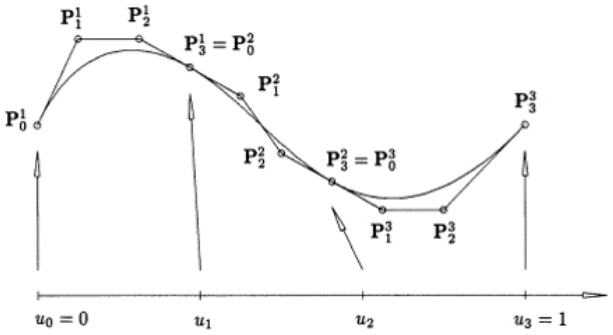
are used in a wide field of applications including entertainment industry, mechanical, electrical and medical engineering, architecture and computer vision. In the NURBS fundamental book [56], the concept of B-spline as well as NURBS curves and surfaces are described in details. Here, the problem of fitting a cluster of points using just a polynomial o rational segment is proved to be inadequate for certain shape. A high degree is required to satisfy a large number of point constrain and this is inefficiently to process and mathematically unstable.
The idea, thus, is to split the curve in various segments:

$$C(u) = \sum_{i=0}^{n} f_i(u) P_i \tag{5.19}$$

where $f_i(u)$ is a piece-wise polynomial function and P_i are control points. Figure 5.44 explains the piece-wise curve approximation and shows the effect of the control points in the curve construction. Differ-



(a) Piece-wise cubic polynomial curve with 3 segments



(b) Polynomial segments represented in Bezier form

Figure 5.44: Piece-wise curve representation. Segments that compose the curve (a), and control points that guide the curve construction (b).

ent piece-wise basis function can be used. For B-Spline basis functions the curve can be express as:

$$C(u) = \sum_{i=0}^{n} N_{i,p}(u) P_i \tag{5.20}$$

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leqslant u \leqslant u_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{5.21}$$

for p=0, while for p=1,2,3,.. :

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - ui} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p-1} - u_{i+1}} N_{i+1,p-1}(u) \tag{5.22}$$

Fitting a B-spline curve to a set of data points p is the task of finding values for the control points that minimize the distance between p and C(u) as shown in Figure 5.45. The definition of a B-spline sur-

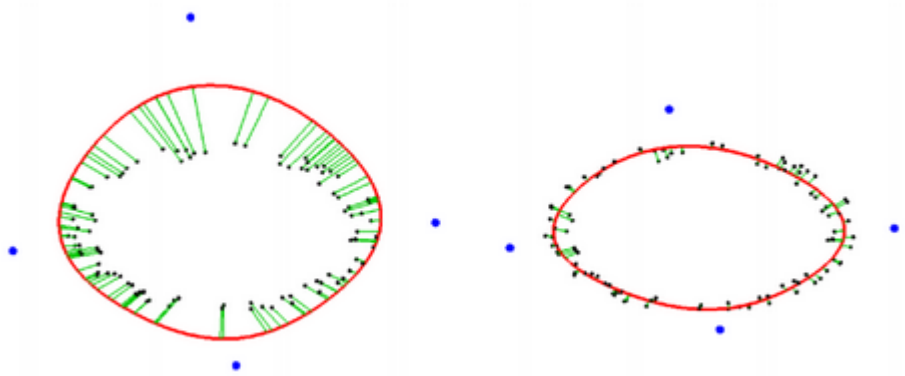


Figure 5.45: Fitting B-spline curves: the distance (green) between the points and the closed B-spline curve (red) is minimized by manipulating the position of the control points

face follows the one of the B-spline curve. A B-spline surface with a parametric domain $(u, v) \in \Omega_S \subset \mathbb{R}^2$ is constructed by linear combinations of the tensor product of B-splines basis functions. Given n, m basis function $f_{i,p}$ and $g_{j,q}$ with i=1,2..,n and j=1,2,..m, the vector-valued coefficients, called control grid, $P_{i,j} \in \mathbb{R}^3$, defines the surface as:

$$S(u, v) = \sum_{i=1}^{n} \sum_{j=1}^{m} f_{i,p}(u) g_{j,q}(v) P_{i,j} \tag{5.23}$$

The same characteristics as for B-spline curves apply for B-spline surfaces. The derivatives in given parametric direction may be determined from the respective one-dimensional basis function. The same

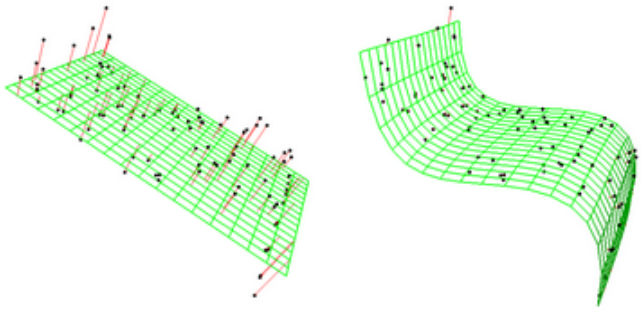


Figure 5.46: Fitting B-spline surfaces: the distance (red) between the points and the B-spline surface (green) is minimized by manipulating the control point.

concept explained in Figure 5.45 in 2D is presented for the three dimensional case in Figure 5.46. Due to the tensor product leading to an orthogonal parametric domain, the boundaries of the reconstructed surface are four-sided. This is not desirable if the searched shape does not have four sides, as for the case studied in this work. One possibility to get rid of the four-sided shape of a B-spline surface is to trim areas that lie outside a certain region. Such a trimming region $\Omega_t \subset \Omega_s$ can be defined on the parametric domain using B-spline curves, the concept idea is displayed in Figure 5.47. B-spline surface fitting has

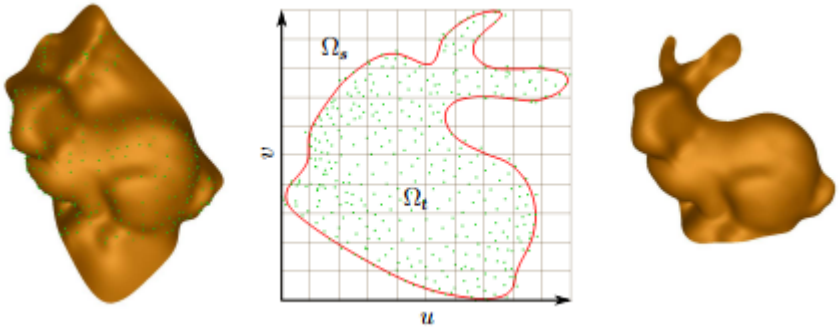
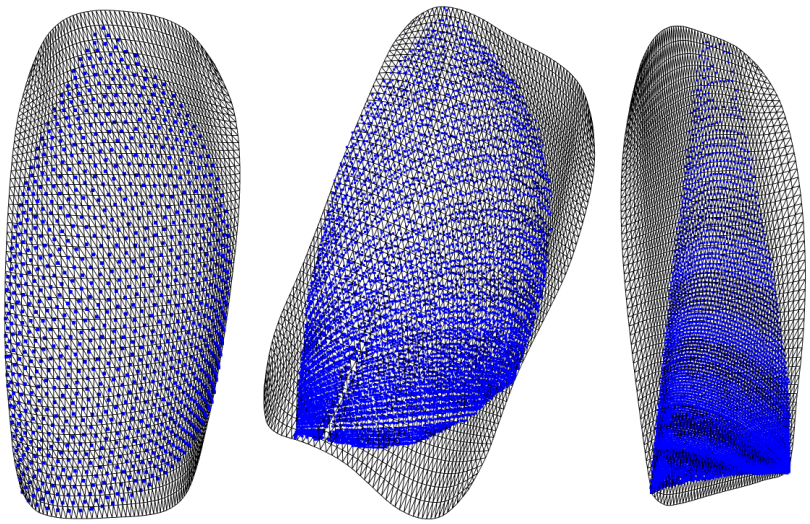


Figure 5.47: B-Spline surface trimming.

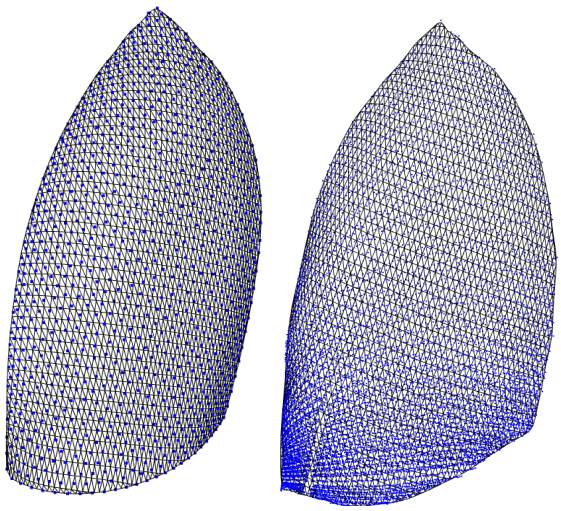
been tested, as for the previously mentioned cases, on artificial data-set and on real point clouds. The process is done iteratively: first a minimum number of control points is considered and the correspondent B-spline is created. The curve/ surface is initialized on/as the plane formed by the two biggest eigenvectors, calculated by principal-component analysis (PCA) of the data-set. A distance metric is associated to the curve/surface, the simplest one is the point-to-line/point-to plane distance, and a threshold is defined; if the evaluation of the metric for a B-spline does not satisfy the limit, then the control points are moved in order to minimize the cost function (as shown in Figure 5.45 and Figure 5.46). Additionally, every k interations, a limited number of control points can be added. Adding control points helps with complex or sharp shapes, but increases the computational costs

of the procedure. Results for these reconstructions are presented in Figure 5.49. The trimmed mainsail reconstruction is not showed be-



(a) Artificial sail (b) Scale model gennaker (c) Full scale mainsail

Figure 5.48: B-spline surface reconstruction without edge trimming.



(a) Artificial sail (b) Scale model gennaker

Figure 5.49: B-spline surface reconstruction with edge trimming.

cause a proper trimming curve has not been retrieved. This is due to the fact the the mainsail acquisition presents in the top part a poor point density. Looking for a trimming curve that follows the point closely would lead to irregular and concave sail edges, easing this constrain would lead to not cut out the four sided surface.
Despite the great results showed by the B-spline surface reconstruction, the necessity to have a robust algorithm, which works with many shapes and facing the non-uniform density problem, brought to use

the strategy explained above and based on 2D Delaunay triangulation. Expedients to take care about this piece-wise linear approximation while analyzing the sail shape are discussed in the next section.

5.7 SAIL SHAPE ANALYSIS

Once the cluster of points belonging to the sampled sail has been extracted, the acquisition noise reduced, and an approximating surface has been created, the geometrical analysis of the shape can be done.
The algorithm is able to automatically detect the shape boundaries. For a triangular mesh for example, the shape perimeter is composed by all the edges that belong only to a single triangle, i.e. not common edges between two triangles. If design parameters given by the sailmaker are available, the length of the total shape perimeter is checked (Figure 5.50c).
Then, vertexes are identified as the intersection between two adjacent boundary segments that form the smallest 3 or 4 (depending whether the sail has a triangular or trapezoidal shape) angles. Adjacent segments representing part of a sail edge for example, form angle close to 180°, while the sail vertexes usually subtend smaller angles (Figure 5.50e and Figure 5.50d).
Once the vertexes are identified, the perimeter can be split in different edges: luff, leech, foot and top are identified and, once again, a metrical check on their length is performed. If displacement between computed length and design length is greater than 10 %, the procedure is stopped and the user can verify the results obtained so far. Luff and leech are, then, split into a number of segments correspondent to the number of sections the user desires. Unlike all the methods mentioned in Section 3.2, there is no limit on the number of sections to analyze and, moreover, sectional planes are not drawn parallel to the yacht deck under the hypothesis that sail sections remain horizontal. In fact, sectional planes are retrieved as follow:

1. Estimation of point normals (representing the local surface normals);
2. Computation of the plane fitting the sail vertexes: $Plane_v$;
3. Computation of the plane $Plane_t$ that passes through the extremities of correspondent luff and leech segment and perpendicular to the $Plane_v$;
4. Computation of the sail section and extraction of the point corresponding to the maximum camber C_{amb} (refer to Figure 5.52);
5. Computation of the angle α between the camber direction and the surface normal in C_{amb} (see Figure 5.51c);

a) if $\alpha \simeq 0$, i.e. the two vectors are parallel, the plane $Plane_t$ can be considered the sectional plane $Plane_s$ and all the other parameters are computed (Figure 5.51b);

b) else the plane $Plane_t$ is rotated iteratively around the section chord by angular steps set by the user until α reaches an acceptable level still set by the user.

Once the correct sectional plane has been retrieved, sail sections are analyzed and synthetic parameters are computed as in Figure 5.52. To verify the output of the sail analysis, the procedure was tested using a data-set coming from a CAD model of a gennaker sail given by the sail-maker for wind tunnel tests. Figures that follow show all the steps described above. Figure 5.53 presents the analysis of 8 curve sections for the gennaker shape and Figure 5.54 provides a comparison between the parameters used to generate the CAD model (called design parameters) and the computed ones.

As shown, the displacement between the design and the computed parameters are on average around 1%, validating the procedure. In Chapter 6 sail acquisitions have been processed and synthetic parameters have been computed as described so far.

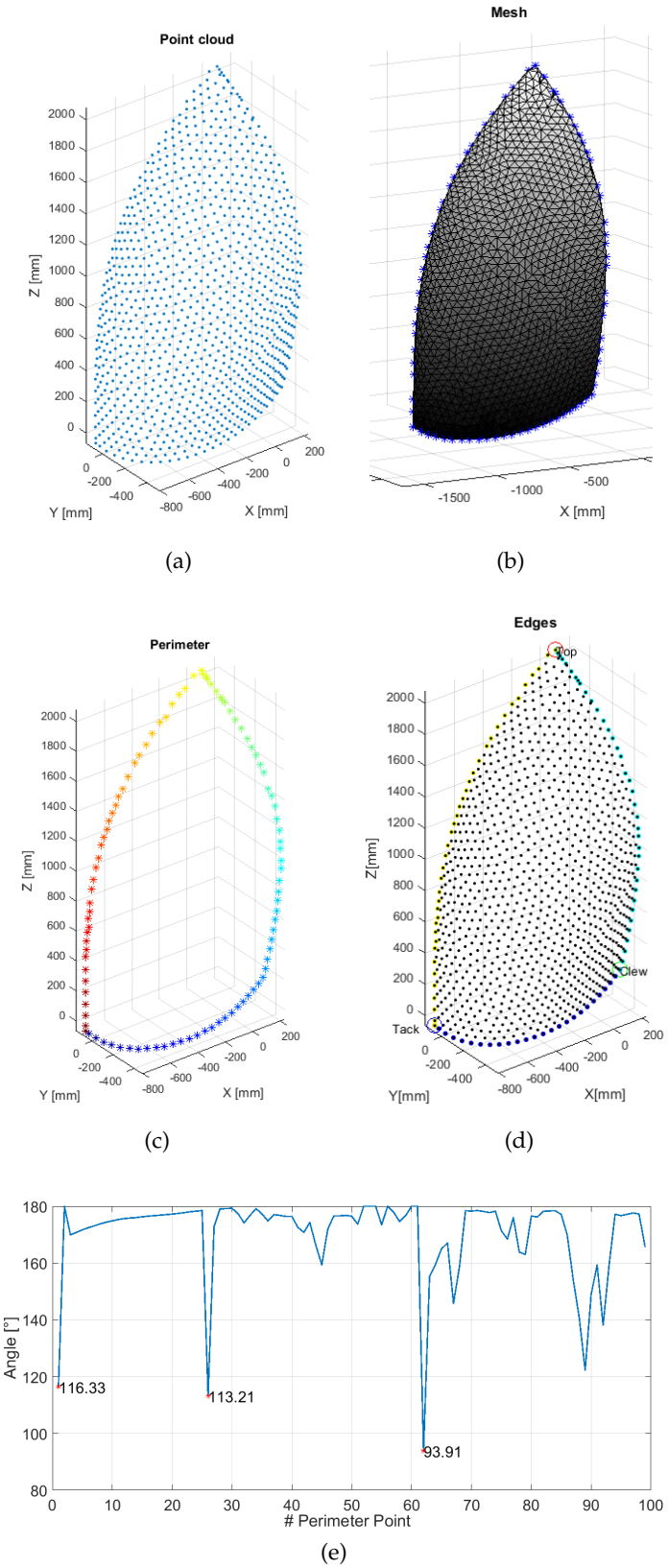
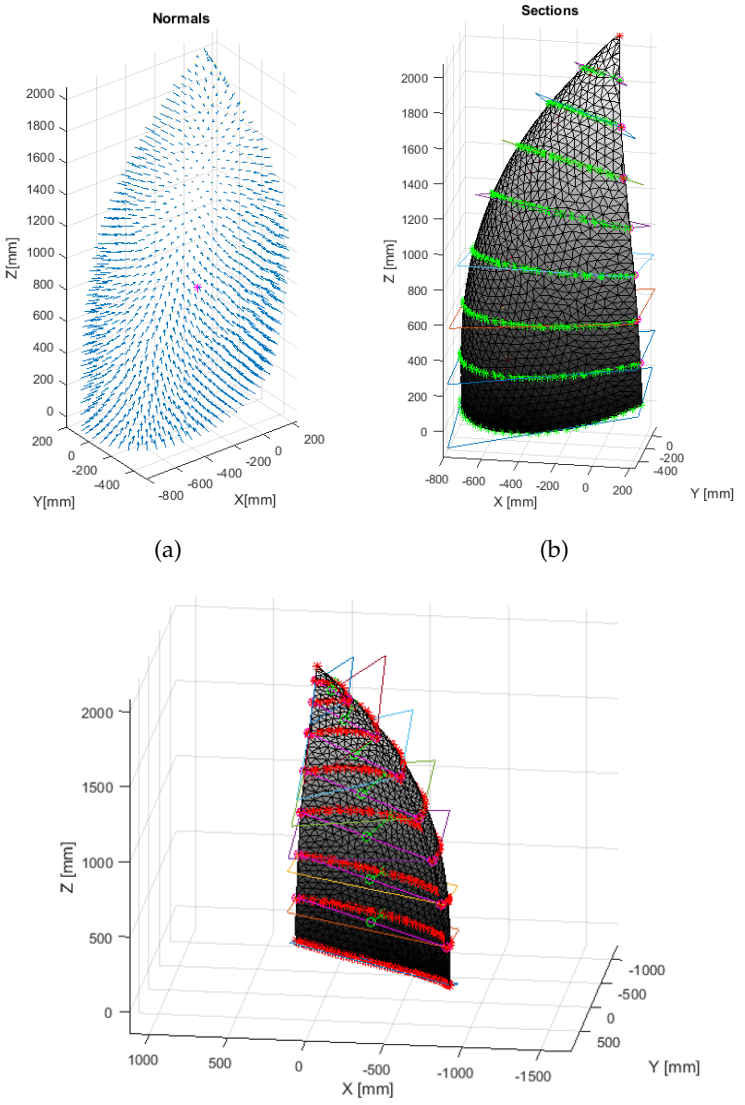


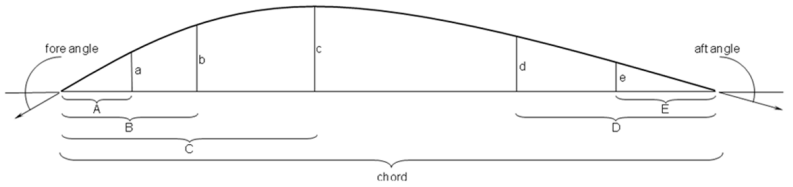
Figure 5.50: Sail shape analysis - Original point cloud (a), triangular mesh approximation (b), shape boundary detection (c), edges detection (d), identification of vertexes as the point corresponding to the smallest angles .



(a) (b)

(c)

Figure 5.51: Sail shape analysis - Point cloud normals (a), sail sections (b), sail section back view (c).



$A = \frac{B}{2}$ $B = \frac{C}{2}$ C *is at max depth* $D = \frac{chord - C}{2}$ $E = \frac{D}{2}$

$front - \% = {}^{b}/_{c}$ $camber = {}^{c}/_{chord}$ $back - \% = {}^{d}/_{c}$

$lead - \% = \frac{a/A}{b/B}$ $draft = {}^{C}/_{chord}$ $trail - \% = \frac{e/E}{d/D}$

Figure 5.52: Sail shape geometrical parameters.

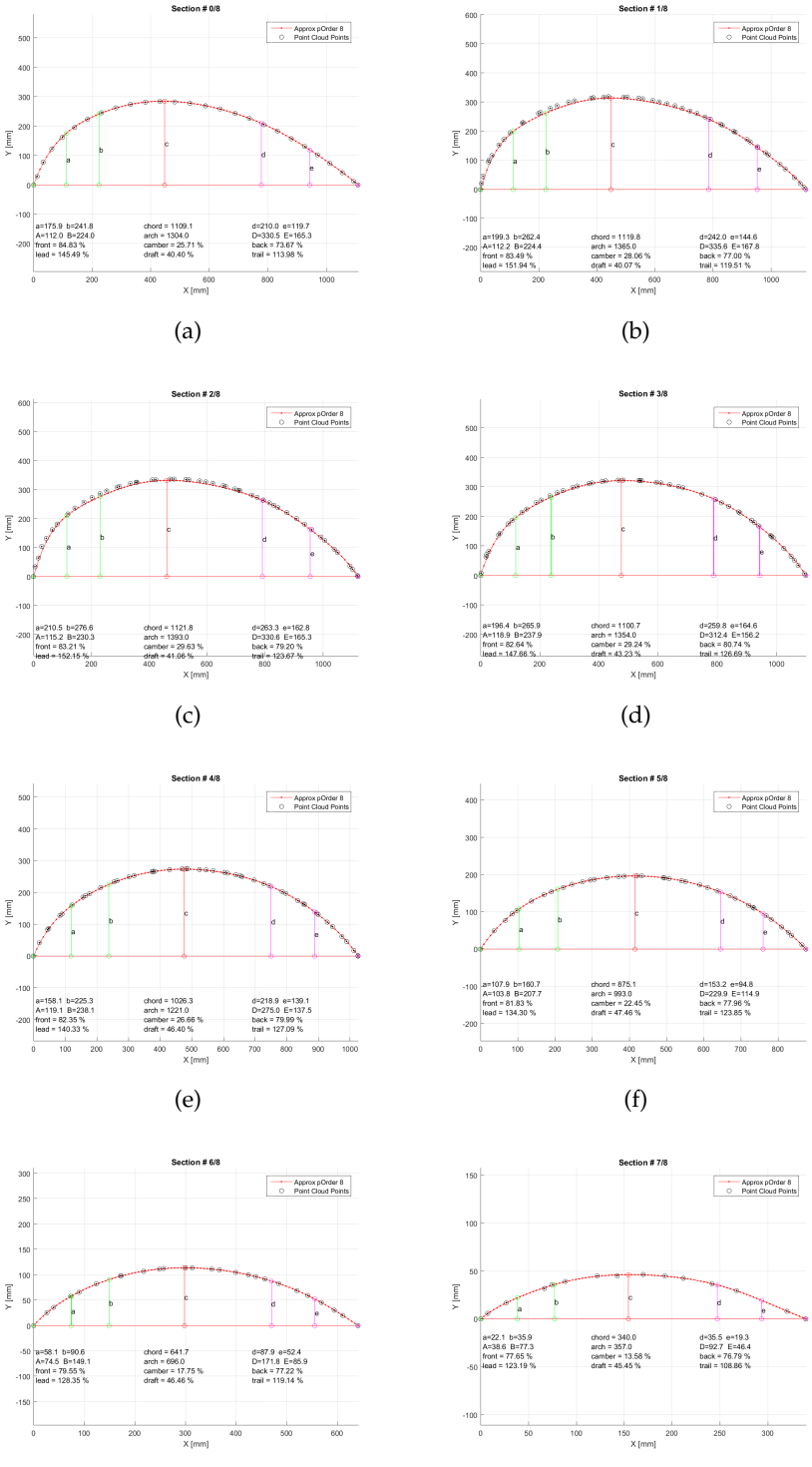
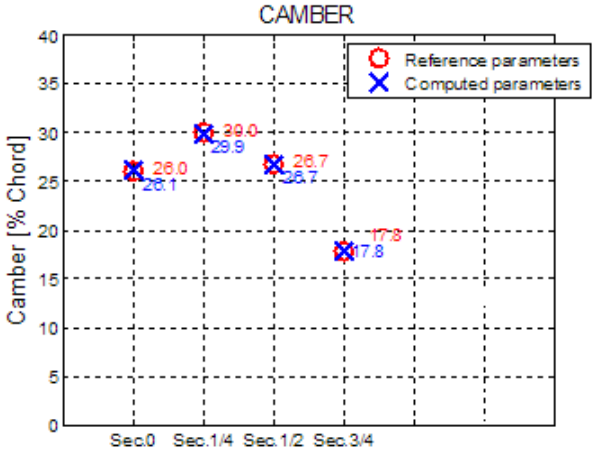
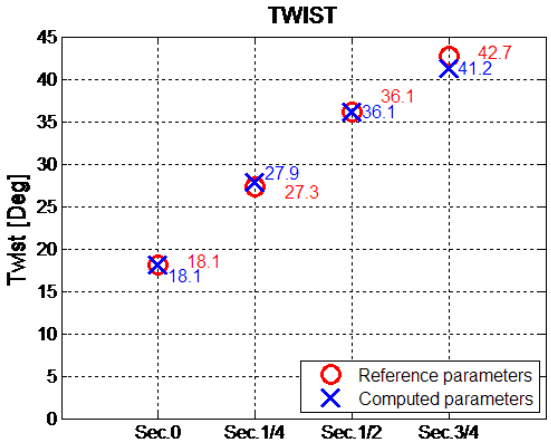


Figure 5.53: Sail section analysis.



(a)



(b)

ERROR [%]	Front	Draft	Camber	Back	Lead	Trail	Twist
Sec. # 0	1.02	1.42	0.37	-2.80	1.67	-1.28	0.00
Sec. # 1/8	-0.49	0.92	-0.41	0.66	-0.53	0.51	1.88
Sec. # 1/4	-0.48	0.69	-0.33	0.62	-0.43	0.51	1.95
Sec. # 3/8	-0.43	0.40	-0.02	0.54	-0.28	0.34	1.25
Sec. # 1/2	-0.32	0.63	0.08	0.09	-0.37	0.16	-0.03
Sec. # 5/8	0.09	1.29	-0.12	-0.06	0.07	0.14	-1.93
Sec. # 3/4	0.00	1.34	0.34	-0.58	-0.44	-0.30	-3.52
Sec. # 7/8	1.03	2.79	0.39	-1.73	0.29	-0.52	-4.83

(c)

Figure 5.54: Comparison between computed parameters and design parameters.

6 CASE STUDY

The methodology described so far for flying sail shape acquisition and analysis is tested and results are reported in this Chapter. Test deals with free form shape representing scale model sails acquired during a wind tunnel campaign aiming at revising ORC aerodynamic coefficients and during tests in support of the validation of an innovative sail pressure measurements system. Asymmetric spinnakers, as well as flat sails such as jibs and mainsails are acquired. Finally two tests conducted on the 10 m Sailing Yacht Laboratory of the Lecco Innovation Hub for a close hauled and a close reach sailing are described, and promising results presented.

6.1 SOFTWARE DESCRIPTION

As mentioned in Section 4.1, data are acquired by means of the TOF scanner unit that can be controlled through a dedicated software developed in LabView environment and reported in Figure 4.5. The user can control the motor motion and scan the angular sector desired with the speed desired.
Once the cloud is acquired, another software, dedicated to the data elaboration, is run. It has been developed in Qt environment through C++ code to be able to exploit some open source libraries such as the Point Cloud Library (PCL) [57], useful for dealing with point cloud data. The graphical user interface for this software is shown in Figure 6.1.

It is composed by a main window mostly occupied by the point cloud viewer in the middle, a tree structure displaying information about the cloud loaded on the right and a toolbar for basic actions such as opening, saving and deleting a file. A second window allows for cloud elaborations. Principal operations are:

1. *Region Of Interest* (ROI): cut out part of the cloud;
2. *Transformation*: rotate or translate a cloud of settable quantities;
3. *Merging*: fuse two or more point cloud into a unique data-set;
4. *Registration*: move the cloud to orient it according to a world reference system;
5. *ICP refinement*: iterative closest point registration refinement;

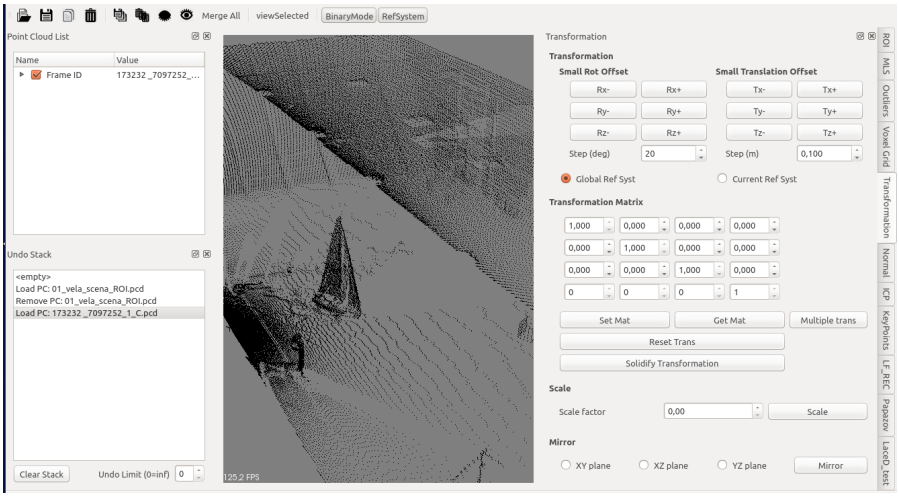


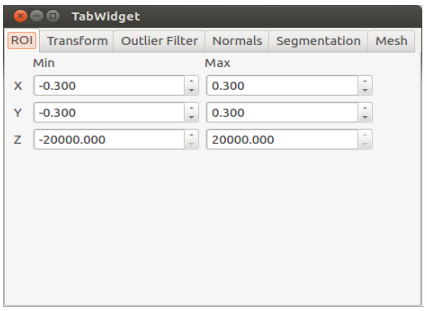
Figure 6.1: Data elaboration software user interface.

6. *Filtering*: outlier removal (statistical filter) or radius filter;
7. *Normal Estimation*: compute the normals;
8. *Segmentation*: separation into different clusters;
9. *Data Resampling*: smoothing a cloud moving a point according to the position of its neighbors;
10. *Surface reconstruction*: meshing or fitting the points.

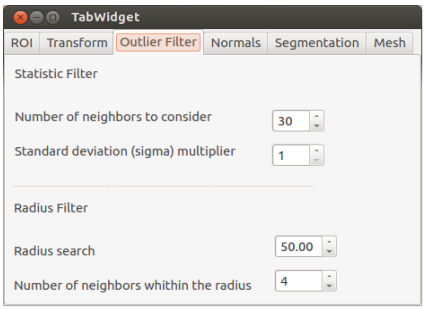
Selecting the tab corresponding to each operation, it is possible to set different parameters to properly tune the results according to the shape one is considering. Figure 6.2 presents an example of the principal parameters to be set for every elaboration step. The analysis of the shape is done running a Matlab code that receives as input the sail cluster, and the design sail dimensions if available, and retrieves synthetic geometric parameters as described in Section 5.7.

6.2 WIND TUNNEL TEST: OFF-WIND SAIL AND FORCE MEASUREMENTS

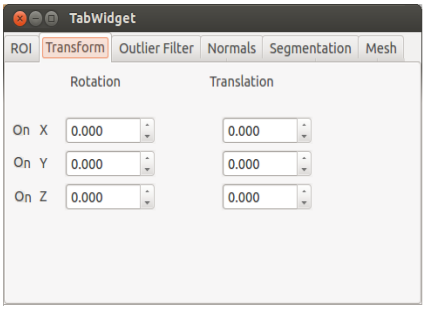
This section provides some results referred to a wind tunnel campaign carried out at Politecnico di Milano during an Offshore Racing Congress project aimed at revising off-wind sails aerodynamic coefficients and ORC VPP aerodynamic model.
The design focus for the required sails to be tested was on asymmetric, masthead, center-line tacked gennakers. Two shapes were produced, an A1 with focus on tight angles for light wind reaching or downwind, and an A2 with focus on broader angles for stronger breeze downwind. For each shape two sizes were produced, a “max”



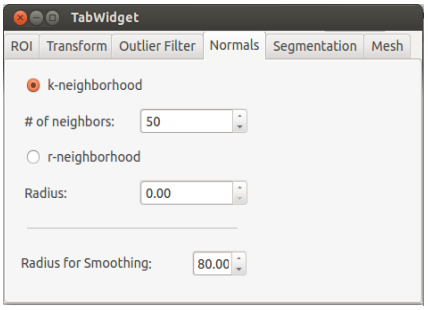
(a) ROI Tab



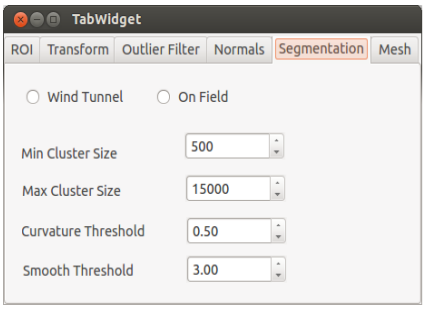
(b) Filters Tab



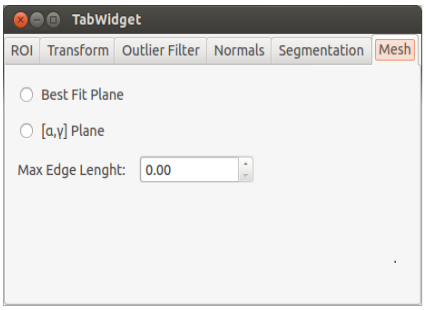
(c) Transform Tab



(d) Normals Tab



(e) Segmentation Tab



(f) Mesh Tab

Figure 6.2: Tab for setting parameters for different elaborations.

version with the maximum reasonable area possible on sail plan, and a "small" version with shortened foot and leech, with area around 87% of the "max" version. During the tests, forces have been measured in a model-fixed coordinate system by a six-component balance and gennaker flying shapes measurements have been taken with the proposed TOF Flying Shape Detection System while the mainsail shapes have been obtained using the currently available Wind Tunnel photogrammetric system described in [12].
Sail shapes acquired with the TOF unit have been reconstructed following the step illustrated in Chapter 5, apart from registration that has been performed manually to merge the mainsail cloud coming from the stereoscopic system and the gennaker point cloud. Segmentation is performed using the parameters reported in Table 6.1 and leading to the cluster shown in Figure 6.3. Outliers filter needed to be

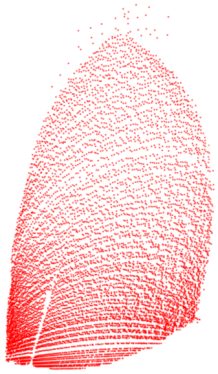


Figure 6.3: Gennaker sail cluster extracted by segmentation - some outliers and mixel pixel points remained in the top part of the sail.

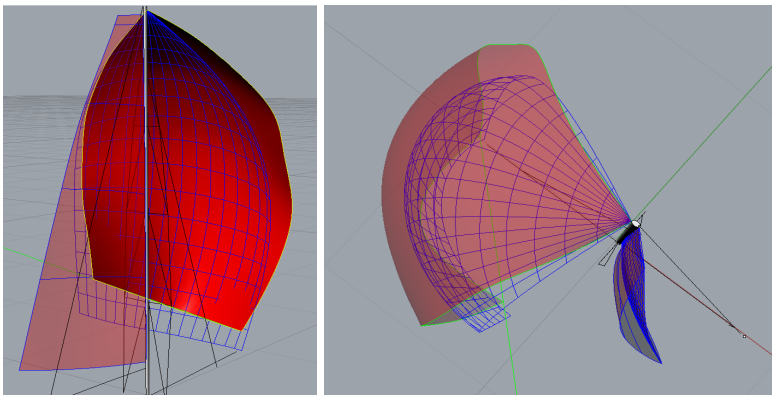
	c_{th}	θ_{th}	$\#_{neigh}$	N_{min}	N_{max}
Gennaker	0.9	7	50	5000	100000
Mainsail	0.8	7	50	5000	10000

Table 6.1: Case study segmentation parameters.

run, then a triangular mesh has been computed to first approximate the sail surface. Sail synthetic parameters have been retrieved for 8 sections, and finally, a loft of the section curves has been performed to render a smooth surface. Then, some consideration about the acquired flying shapes and their correlation to the measured forces have been done and explained in the next sections.

6.2.1 *Design-flying shape comparison*

In this section the results of sail scans at 95° AWA are shown and compared with design shapes in correspondence of the sail trim generating maximum driving force for the given AWA Figure 6.4 shows the comparison between rendered design shape (blue grid) and the rendered flying shape surface (red) obtained by the sail scan measurements. As previously mentioned, using the proposed TOF tool sail surface design parameters can be extracted. For the present case, although several section parameters have been derived for each curve, in Figures 6.5 - 6.8 only camber (profile depth in percentage of chord) and twist (related to boat center-line) are drawn for both sails (mainsail and gennaker), being the most significant for aerodynamic forces. In addition, two significant vertical curves are analyzed with synthetic parameters: LUFA, that is the projection of luff curve on a plane defined by sail corners, and LUSW, that is the projection of luff curve on a plane defined by tack and head and perpendicular to the previous one. Both sails show significant differences from design shape, which can be understood by a comparative analysis involving both



(a) Side view (b) Top view

Figure 6.4: Comparison between rendered design shape (blue grid) and the rendered flying shape surface (red).

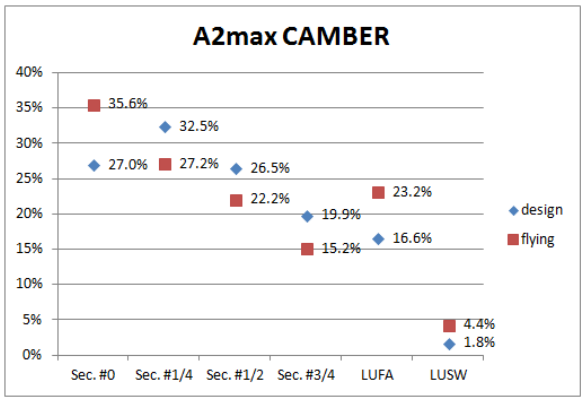


Figure 6.5: A2 max sail. Camber distribution (design vs flying shape).

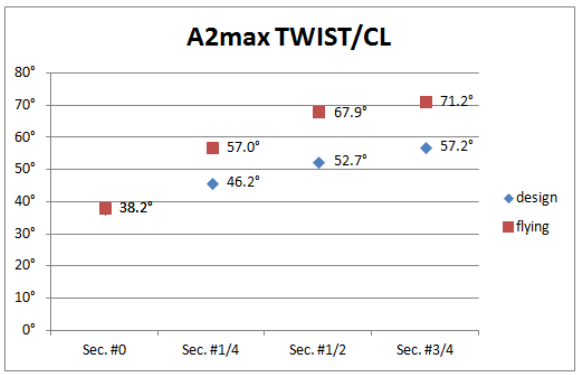


Figure 6.6: A2 max sail. Twist distribution (design vs flying shape).

horizontal and vertical cambers, and section twist. In fact, gennaker A2max is flying flatter than designed, except at lower section, being both trimmed with a higher tack and being more twisted. The reason for this difference in tack height relies on the search for maximum driving force – the flying shape is related to the best trim achieved in

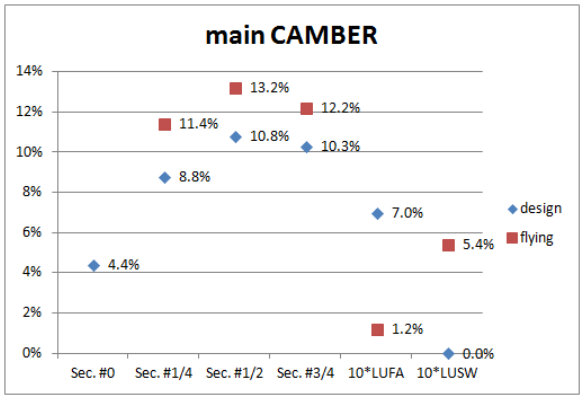


Figure 6.7: Mainsail. Camber distribution (design vs flying shape).

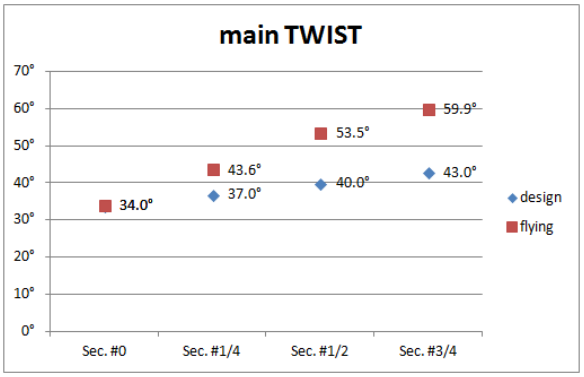


Figure 6.8: Mainsail. Camber distribution (design vs flying shape).

wind tunnel – for the given wind condition. The tack trim adds vertical curvature, which necessarily ends in a reduction of the horizontal one, being the total curvature approximately constant; twist has a similar effect on horizontal curve reduction. Mainsail shows a similar behavior, while opposite as concerns vertical and horizontal curvature. More intuitively, the mast is much straighter than design luff curve - being the latter targeting a mean upwind condition, while in downwind test condition it is allowed to straighten - thus resulting in fuller horizontal sections. The resulting flying shape can be better understood looking at three-dimensional rendering resulting from sail scan (Figure 12-13). In the following, a much different tack trim for gennaker as well as a deeper luff curve and an added twist, as long as an added twist at main. It is appreciable how the mast effect (and possibly low luff tension) results in a deeper main in spite of twist increase. In addition, it is interesting to compare rendered shape with pictures from wind tunnel, illustrating the capability to capture irregular shapes such as on the luff in this case (Figure 6.9). Such a comparison between design and flying shape might be thought of interest in the context where three-dimensional design shape is intended as an approximation of flying shape aimed at generating building (2D) parameters (sail panel and edge curves- the ones influencing the fi-

Figure 6.9: Gennaker flying shape and forces variation with trim.

nal flying shape, together with control lines trim). Therefore, aiming the design shape at realistic (or even optimal) flying shape should be possible, even leaving unchanged the building parameters, thus producing the same design shape. This in turn leads to the possibility of introducing parametric changes to 3D design shape, which may be expected to represent more closely the expected variation in flying shape.

6.2.2 *Gennaker flying shape and forces variation with trim*

In the following, focus is on the relationship of trim maneuvers on gennaker control lines - primarily tack - with sail shape and sail plan forces. Test was carried out at 80° AWA and the trim maneuver consisted in easing the tack line starting from an all-down position and trimming the sheet as needed. Images from 3D rendering, in Figure 6.10, show in detail tack easing, clew movement and overall shape change, in three subsequent trims. Synthetic parameters describe this process in detail, showing how at each step the sail loses depth in the upper part while becoming deeper in the lower. Interestingly, the first trim manoeuvre (form step 0 to step 1) does change a little the luff curve, while the twist is increased due to sheet easing. The second trim, from step 1 to step 2, instead, results in a big increase of luff curve associated with lower twist, apparently due to little sheet trim. Pictures taken in the wind tunnel (Figures 6.14 - 6.16) do confirm in particular the similar luff curve between the two first steps, with a bigger difference in the third. Sail forces evaluation (Figure 6.13) is well correlated with gennaker shape variation. While the driving force is increasing almost linearly with tack ease, confirming the "best practice" in use in sailing regattas, the lateral force at step 1 decreases with twist increase. Step 2, instead, associates more side force with

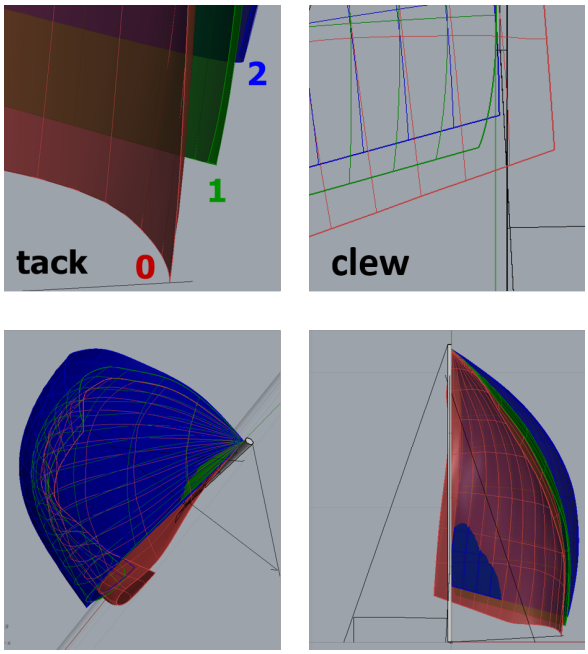


Figure 6.10: Trim maneuvers on gennaker control lines (starting configuration 0 (red), first trim configuration 1 (green) and final configuration 2 (blue)).

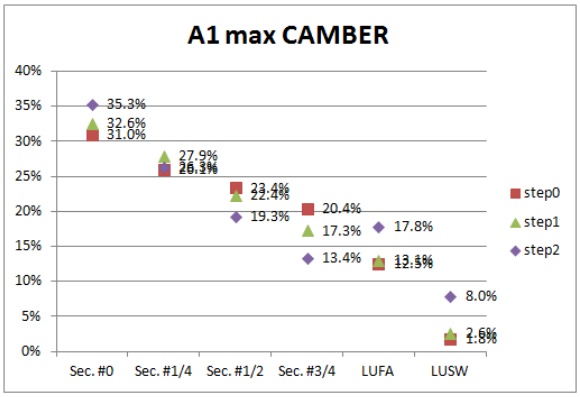


Figure 6.11: A1 max sail. Camber values for different trimming steps

less twist. A deeper analysis can be carried on each curve, being horizontal section or luff curve, for which synthetic parameters have been derived. As an example, in the following diagram all parameters for horizontal middle section are interpolated by a spline allowing an intuitive shape evaluation. A vertical line representing camber and its longitudinal position is drawn, to show how profile depth not only diminishes but also moves forward. The analysis carried out in this paragraph suggests for each design shape a set of possible flying shapes and associated forces can be derived, under parametric control lines variation, identifying reasonable trims. This would make it possible to identify an envelope of possible coefficients to feed a VPP with a complex set of coefficients for each different sail candidate,

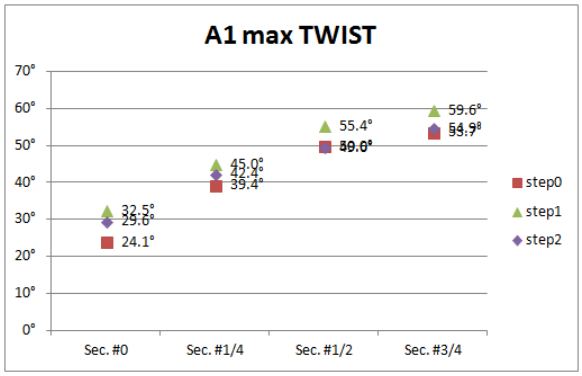


Figure 6.12: A1 max sail. Twist values for different trimming steps.

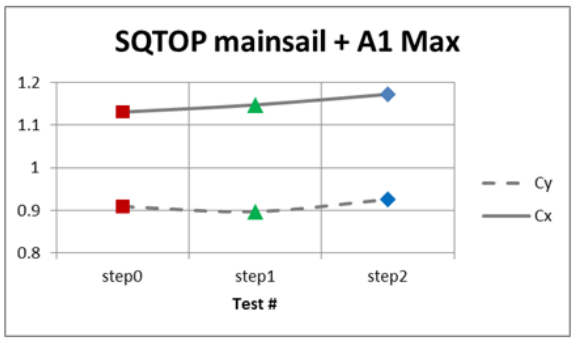


Figure 6.13: Driving and side forces for different trim.

Figure 6.14: Picture from wind tunnel test- Step 0.

better showing behavior under trim variation and relative crossover. The availability of a set of trims and flying shapes for a given design could be also of great help in setting up CFD investigations with the purpose of finding forces and analyzing flows, saving a lot of computational time usually dedicated to this activity, and bypassing the subjective human factor.

Figure 6.15: Picture from wind tunnel test- Step 1.

Figure 6.16: Picture from wind tunnel test- Step 2.

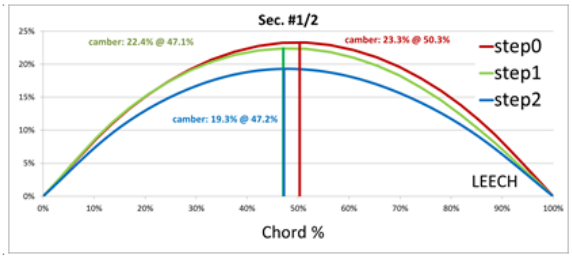
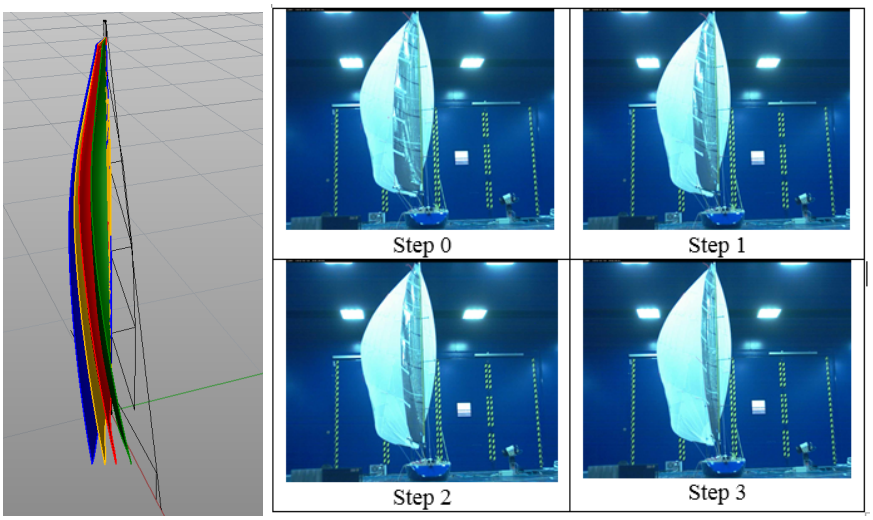


Figure 6.17: A1 max sail. Shape variation for sail Section ♯1/2.

6.2.3 *Mainsail flying shape and force variation with trim*

A similar analysis has been conducted on mainsail trim and its association with sail shape and forces. With reference to tests carried out with the small A2 at 90° AWA, starting from an optimum drive trim, mainsail has been progressively trimmed with less twist by a combined use of sheet and traveler. Starting from step 0, sheet has been

trimmed to step 1, than traveler to step 2 then again sheet to step 3. Figure 6.18a shows the rendered mainsail flying shape surfaces obtained by the corresponding sail measurements, which can be compared with the pictures taken in the wind tunnel (Figure 6.18b). The effect on twist (Figure 6.19) is clear and well correlated with maneuvers: step 0 to 1 the high sheet angle contributes in reducing twist more at foot than at head; step 1 to 2 traveler is brought to windward contributing to an even twist reduction; 2 to 3 sheet is tightened ending close to vertical therefore tensioning leech and reducing twist much more at head than at foot. Forces diagram (Figure 6.20) do show a slight reduction in drive associated to an increase in lateral force, probably due to an increased portion of mainsail being stalled. In conclusion, results for these tests highlighted the differ-



(a) Rendering (b) Pictures

Figure 6.18: Different mainsail trim.

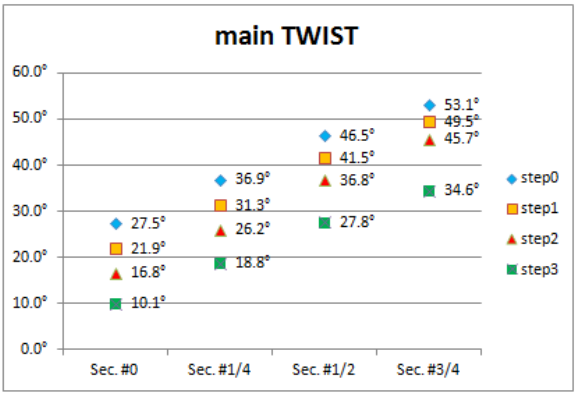


Figure 6.19: Mainsail. Twist values for different trim steps.

ences between flying and design shapes and the sail shape variation as consequence of different trims along with the sail-plan aerody-

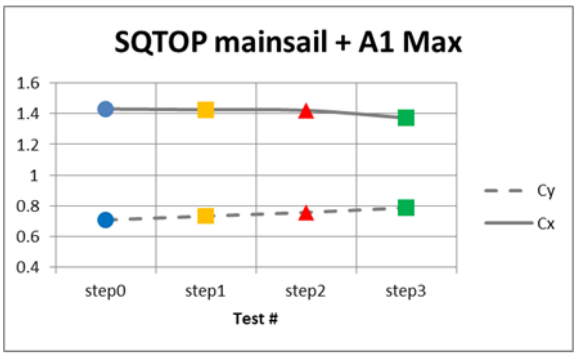


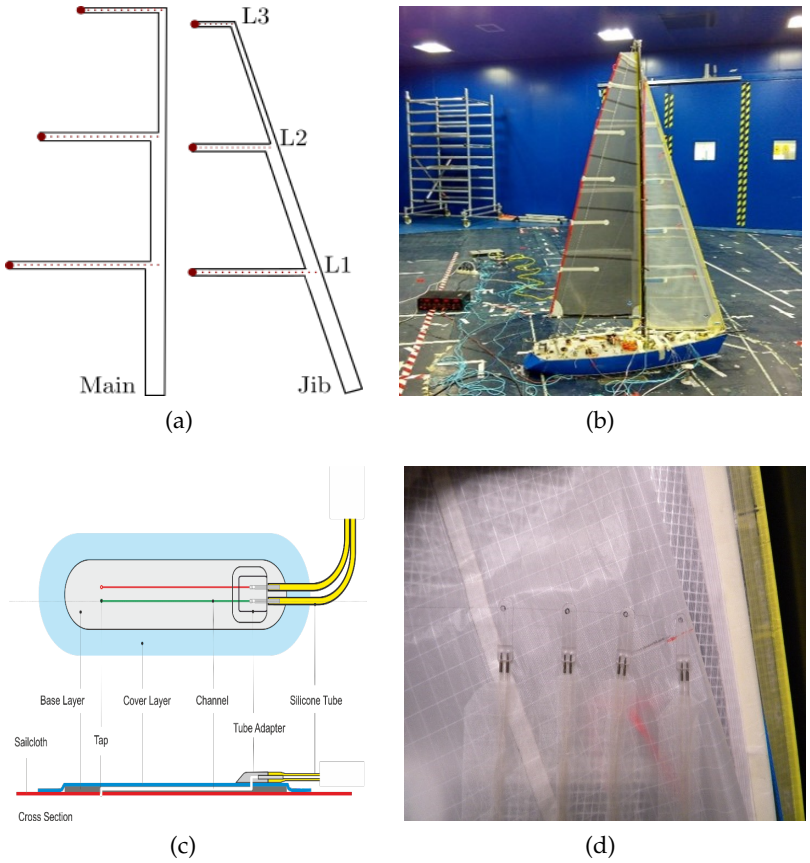
Figure 6.20: Driving and side forces for different trim.

namic performances. Potential improvements on current sail design process and techniques have been discussed, regarding both design shape intended as 3D shape generating building curves, and the opportunity to associate each design shape to a set of possible flying shapes and their related aerodynamic coefficients. Such an approach is intended to both directly interacting with VPP by evaluation of a selected list of design candidates, and possibly inversely letting the VPP identify the best sail design among a database, by match with one or more optimum criteria. Further work is currently in progress in particular with reference to numerical tools bench-marking purposes.

6.3 WIND TUNNEL TEST: SAIL SHAPE AND PRESSURE MEASUREMENTS

Other tests on scale models have been done in the wind tunnel of Politecnico di Milano. This time, tests came in support of a joint project among Politecnico di Milano, CSEM and North Sails, aiming at developing a new sail pressure measurement system based on MEMS sensors (an excellent compromise between size, performance, costs and operational conditions) and pressure strips and pads technology. These devices were designed and produced to give differential measurement between the leeward and windward side of the sails. This wind tunnel campaign was intended to provide a benchmark of the whole measuring system for future full scale applications.
The pressure devices are shown in Figure 6.21. CSEM C16 scanner is a miniaturized electronic pressure scanner with 16 differential pressure sensors and a CAN bus interface for the communication. High attention was given to dimensions and shape of the scanner box (only 6 mm in height), so to have minimal impact on the airflow. Each of the 16 sensors has its own reference input which makes the scanner especially suited for measuring the pressure difference between lee-

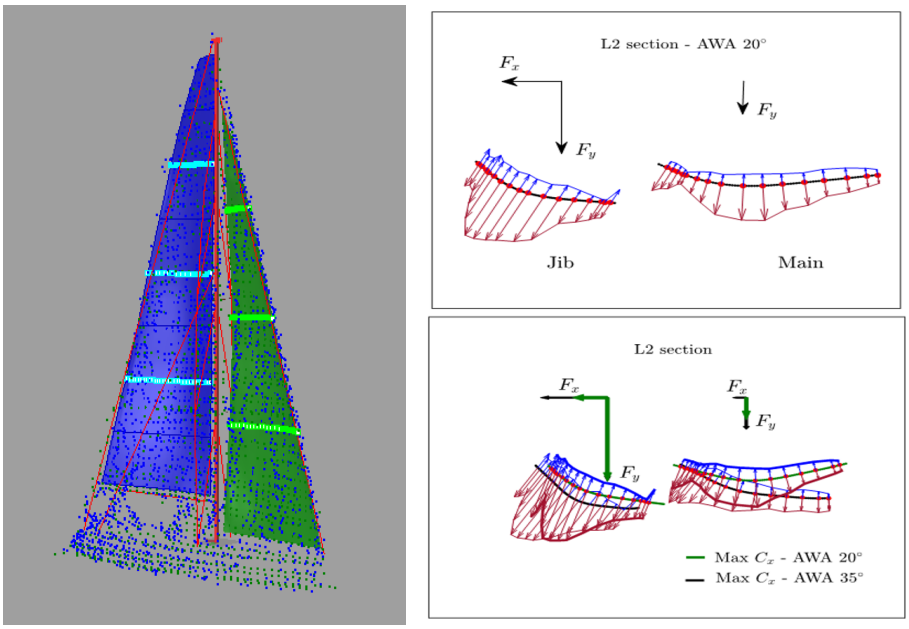
ward and windward side on dedicated spots on the sails. Tiny micro-channels in the pressure strip, made of thin polymer films, propagate the pressure from the tap to the connected pressure scanner. Manufacturing processes have been developed using laser and micro-milling to produce strips with comparatively deep channels. In total 40 taps have been measured per sail. For further technical details refer to CSEM website. A complete 1:10 scale model of a 48' cruiser-racer, con-



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

Figure 6.21: Pressure devices - strips for mainsail and jib (a-b), and pads for off-wind sails (c-d).

sisting of yacht hull body (above the waterline) with deck, mast, rigging and sails was mounted on a six component balance, which has been fitted on the turntable of the wind tunnel (Figure 6.21b). Both sides of the main sail and the jib have been equipped with tailor-made pressure strips. Flying shape detection systems, have been adopted to perform shape measurements along with pressure and force data. A scanner was dedicated to the jib and placed on the ground close to the bow, the other was placed frontally to the mainsail a couple of meters far away not to alter the wind flow. A master software has been programmed to trigger synchronously the acquisition of all devices. Different upwind angles have been tested, from 20° AWA to 35° AWA. Example of the results are provided in Figure 6.22. Cloud



(a) (b)

Figure 6.22: Wind tunnel results - From point cloud to sail surface reconstruction - sail sections in correspondence of the pressure stripes highlighted - (a), and pressure values coupled with the sail shape for section 1/2 (b).

registration has been done following the procedure described in Section 5.3.3, that consist in aligning the wind tunnel chamber walls. Sail clusters extraction and resampling followed. Once a surface had been obtained, sections were realized in correspondence of the pressure stripes, i.e. at 1/4, 1/2 and 3/4 of the luff length for both sails. Section curves have been coupled with the pressure measurements as displayed in Figure 6.22b; where flow separation at the luff edges can be appreciated (top picture), as well as the difference in twist for sail trimmed in two wind sailing conditions (bottom picture). A detailed analysis of the results of this campaign has been reported in a paper submitted for the next Chesapeake Sailing Yacht Symposium of Annapolis (march 2016).

6.4 FULL SCALE TEST: FLYING SAIL SHAPE MEASUREMENTS

Full scale tests have been carried out onto the Sailing Yacht Lab (Figure 6.23), a 10 m length sailing boat developed within the Lecco Innovation Hub project of Politecnico di Milano. This yacht was designed to operate as a dynamometric balance and was strongly inspired and encouraged by the previous experiences developed at MIT [58], at Kanazawa Institute of Technology [17] and Berlin TU [59]. The heart of the system is a 5083 aluminum alloy frame inside the hull that al-

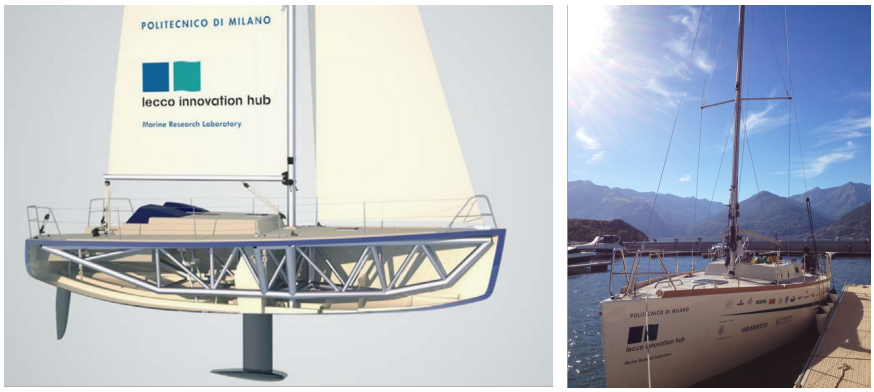
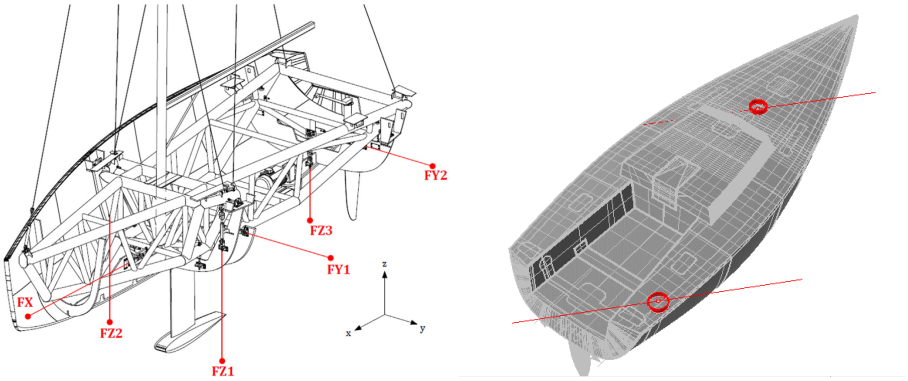


Figure 6.23: Sailing Yacht Laboratory concept (left) and photo (right).

lows the entire rig and sail plan to be connected to a set of load cells to measure the overall forces and moments transmitted by sails and rig to the hull. Figure 6.24 shows the general arrangement of the six load cells which keep the frame in isostatic constrained configuration with respect to the hull. With respect to the shown reference system,



(a) (b)

Figure 6.24: Load cell arrangement (a), and TOF acquisition unit positions (b).

the 6 load cells are set as follows: one load cell along X direction (FX), two along Y direction (FY1, FY2)and three along z direction (FZ1, FZ2, FZ3). The idea behind the choice of their position is to be as close as possible to the highest loads along the respective directions, so that e.g. FX cell is placed near the mast-step connecting the hull to the central part of the frame. In addition, an important criterion of the positioning of the cells was their accessibility during maintenance operations. Maximum loads of the adopted cells are respectively 20 kN for the FX, FY1, FY2, FZ3 and 50 kN for the FZ1 and FZ2.
Moreover, the yacht is equipped with a classical navigation equipment providing wind speed and direction, boat speed, depth as well

as the yacht course by means of a differential GPS receiver; an ultrasonic 3D anemometer mounted on the top of an additional mast; pressure measurement devices composed by MEMS sensors and dedicated pressure pads which have been designed and produced aiming at providing the differential measurement between the sail leeward and windward side; and finally, two TOF acquisition units to measure the sail shapes. For an exaustive desription of the project refer to [60].
Figure 6.24b shows the position of the two scanners on the deck, one at mast foot and the other at the stern next to the winch, while Figure 6.25 shows two photos of the set up. Elaboration of data for two tests

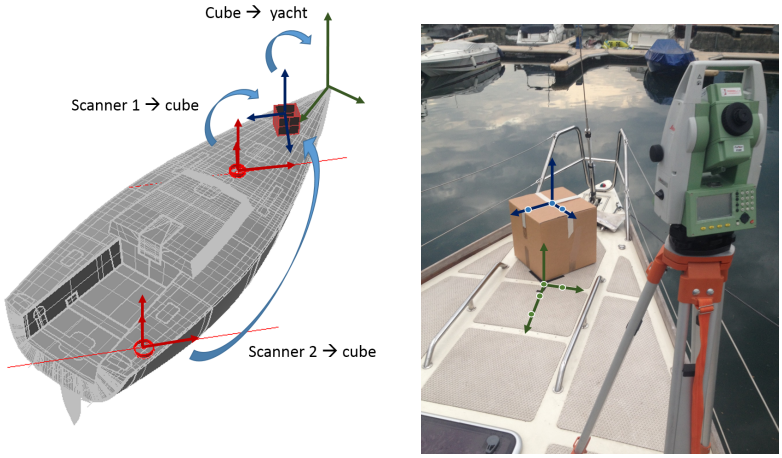
(a)

(b)

Figure 6.25: Full scale setup - acquisition unit dedicated to the mainsail (a), and acquisition unit dedicated to the jib (b).

are reported below: sail plan reconstructed for a close-hauled sailing

and a close reach sailing. For each test two different point clouds have been synchronously acquired, and, then, registered. For cloud registration, once more, the procedure explained in Section 5.3.3 has been used. As planar surfaces this time, three faces of a cube placed at the yacht bow have been exploited. The cube could be scanned by both TOF units and was positioned at the beginning and at the end of the sailing tests to verify the scanners remained in their original position. Computing the pose of the cube surfaces, it is possible to calculate the rototranslation matrix to orient the clouds from the scanner reference systems (in red in Figure 6.26a) to the cube reference system (in blue). Moreover, the position of the cube with respect to the yacht reference system (in green) has been retrieved sampling points on the deck and on the cube faces with a Total Station. This process allows for registering the sail clouds onto the CAD model of the hull. Figure 6.26 explains the above and Figure 6.27 presents the results for the registration.



(a) (b)

Figure 6.26: Cloud registration procedure - Planar surfaces retrieved by scanning the cube faces.

Once the cloud had been registered, the data elaboration started. Clouds have been segmented as shown in Figure 6.28, and using the parameters reported in Table 6.2. Finally, surface have been reconstructed and Figure 6.29 presents the final sail plan rendering. Figure

	c_{th}	θ_{th}	$\#_{neigh}$	N_{min}	N_{max}
Jib	1	10	20	5000	100000
Mainsail	1.5	4	30	6000	100000

Table 6.2: Segmentation parameters for full scale tests.

6.32 presents a detail of the jib that bends due to the handrail as visible in the photo on the right. Similar to the close reach sailing case,

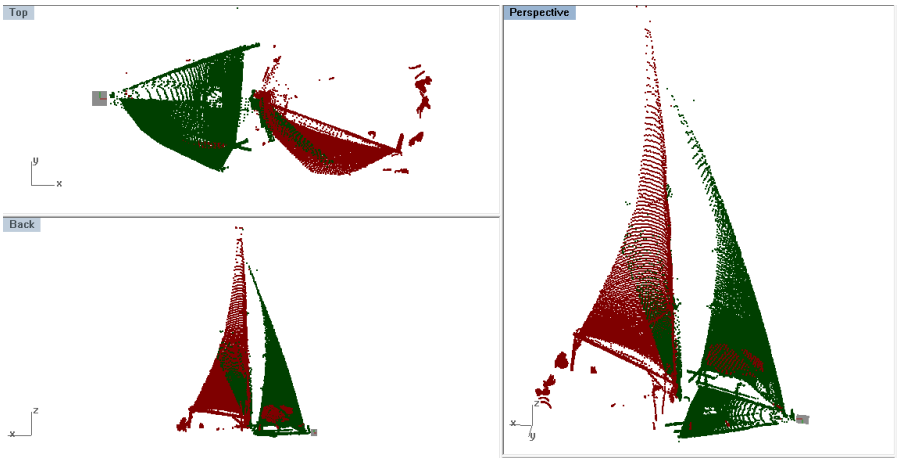
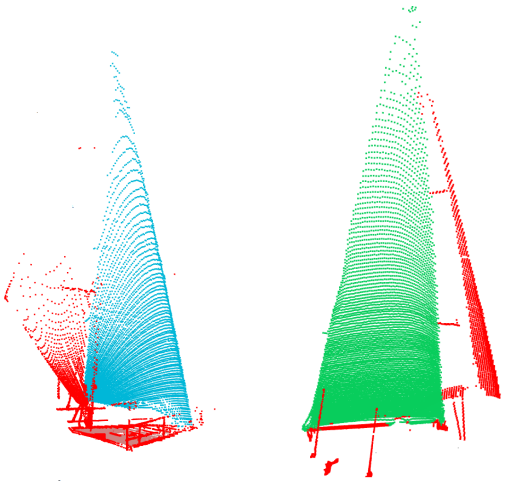


Figure 6.27: Cloud registration results for close reach sailing.



(a) (b)

Figure 6.28: Cloud segmentation for jib cloud (a), and mainsail cloud (b).

Figure 6.30 presents the sail plan reconstruction for the close hauled sailing test. Note the sail hauled with respect to the close reach case (Figure 6.31). Finally, synthetic parameters can be extracted. Figure 6.34 reports the curve sections for jib and main of the close hauled case. Geometrical parameters are also summarized in Figure 6.33.

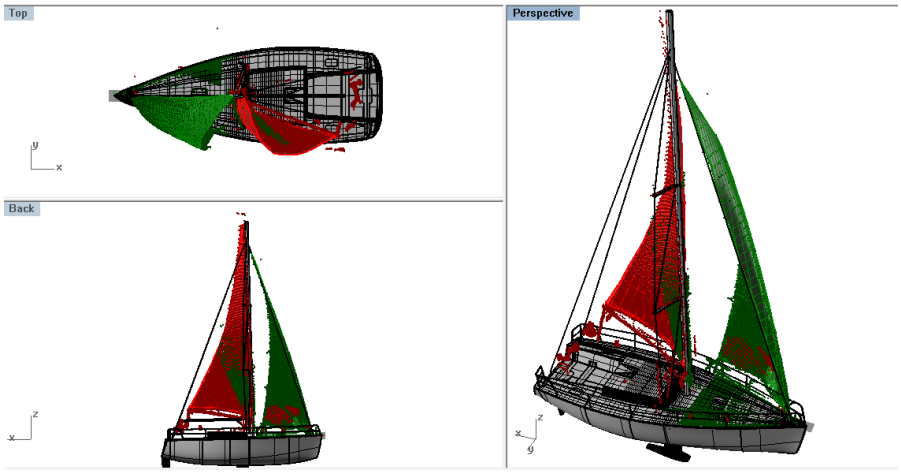


Figure 6.29: Sail reconstruction for close reach sailing.

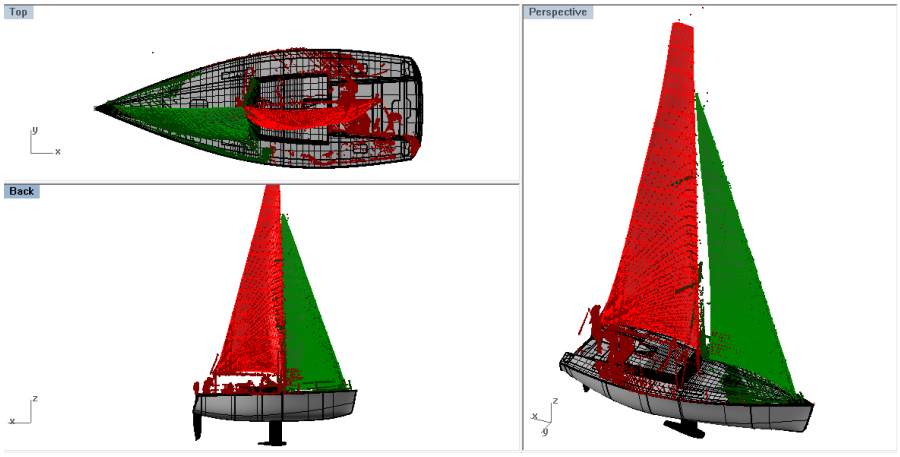
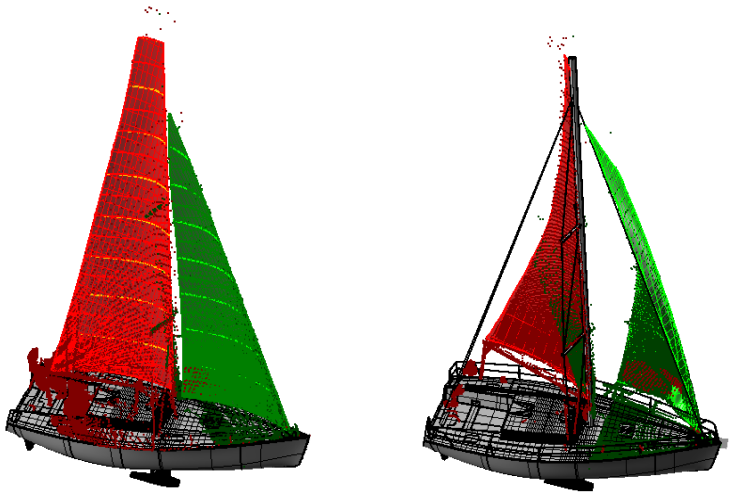


Figure 6.30: Sail reconstruction for close hauled sailing.



(a) (b)

Figure 6.31: Sail plan reconstructed for close hauled test (a), and close reach test (b).

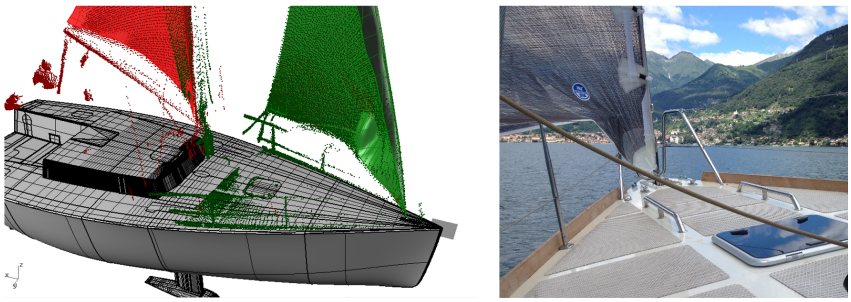


Figure 6.32: Detail of the reconstructed jib bending onto the handrail.

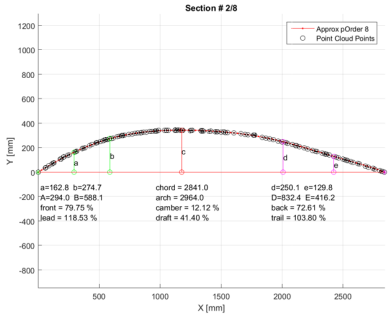
Section #	front	draft	camber	back	lead	trail	twist
0	0.8	50.5	8.7	0.7	1.2	1.1	11.7
1	0.8	42.5	10.5	0.7	1.2	1.1	15.0
2	0.8	41.4	12.1	0.7	1.2	1.0	18.0
3	0.8	43.4	12.9	0.7	1.2	1.1	21.0
4	0.8	42.4	12.7	0.7	1.2	1.0	23.5
5	0.7	38.4	12.0	0.8	1.2	1.1	26.2
6	0.7	43.5	9.8	0.7	1.1	1.0	27.5
7	0.8	42.4	3.6	0.5	1.2	0.2	31.5

(a) Jib - 8 section parameters

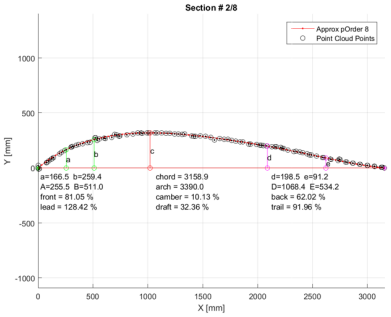
Section #	front	draft	camber	back	lead	trail	twist
0	0.8	47.5	6.3	0.7	1.1	1.1	2.2
1	0.9	24.3	6.3	0.7	1.3	1.0	3.2
2	0.8	32.4	10.1	0.6	1.3	0.9	6.7
3	0.8	32.3	12.5	0.6	1.1	0.8	8.7
4	0.7	38.4	14.5	0.6	1.1	0.9	10.5
5	0.7	39.4	14.3	0.6	1.2	1.0	12.8
6	0.8	37.4	14.0	0.7	1.1	1.0	15.2
7	0.8	37.4	8.6	0.6	1.1	0.9	16.8
8	0.7	73.7	4.2	0.9	1.2	1.3	10.5

(b) Mainsail - 8 section parameters

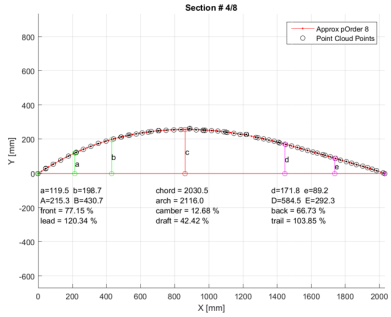
Figure 6.33: Geometrical parameters computed for 8 sections on the jib (a) and on the mainsail reconstructed surface (also the squared top considered as 9th section) (b).



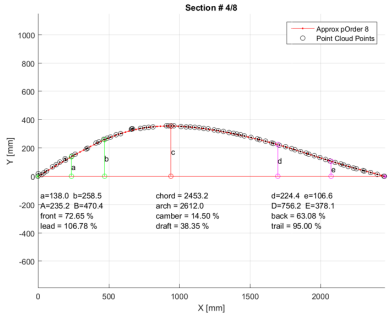
(a)



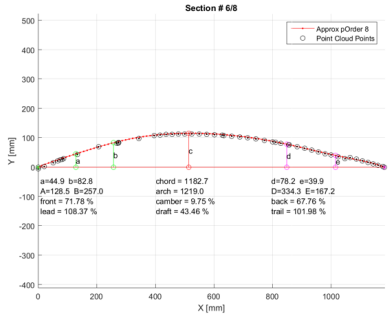
(b)



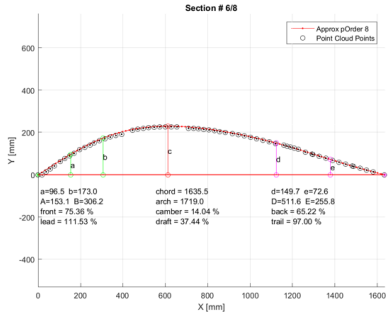
(c)



(d)



(e)



(f)

Figure 6.34: Geometrical parameters computed for sections 1/4, 1/2 and 3/4 for the jib (right) and for the mainsail (left).

7

CONCLUSIONS

RESEARCH CONCLUSION

The main aim of this research was to propose some novel methods for the acquisition and the analysis of free-form shapes.
In particular, for this study, the flying sail shape has been consider as free-form object to be properly measured and geometrically analyzed. The aerodynamic pressure distribution, the forces resulting from rig design and trim, the elasticity of the material and the varying wind conditions leads to a virtually infinite number of shapes that a sail might assume during navigation. These shapes can not be easily described by means of regular geometries - and this is especially true for off-wind sails -, and has to be properly handled. The shape assessment pipeline can be divided in the following steps:

1. *measurement*;
2. *point cloud data elaboration* (which encompasses registration, segmentation, filtering, resampling,and surface modeling) ;
3. *shape analysis*;
4. *performance evaluation* (correlation between object shape and its functionality).

Most commercial applications are only able to handle primitive geometries and require the interaction with the user. Moreover, the solutions provided by the research community are often not satisfactory at all, since the pipeline steps are faced separately. This research, instead, intended to face the whole process, starting from the data acquisition, following with the elaboration and ending with the shape analysis. The algorithms implemented in this work take as input a point cloud, no matter the type or how it is acquired; thus, they can be used in different contexts or with different acquisition devices. The goal was to present the possibility of this method to the nautical field, so that measuring, reconstructing and acquiring the flying sail shape could be useful for a better understanding of the what is happening above the water line. In fact, nowadays, to assess the yacht aerodynamic performances Computational Fluid Dynamics (CFD) codes, or even Fluid Structure Interaction codes can be exploited. However, they start from a design sail shape, that can be by far different from the real flying shape, and still need a massive validation work. This study is intended to provide a contribution to this issue, as detailed in the following.

Measurement

In this work, a customized measurement system, based on Time Of Flight technology, has been realized to acquire the free-form sail shape. The objective was to try to overcome most of the weaknesses of sail acquisition systems tested in the past. Most of them are based on photogrammetry technique that allows for retrieving the position of markers placed on the sail surface, and requires at least a pair of camera for each sail. However, more devices are usually exploited to avoid occlusions. Other techniques use only a camera placed on the deck or at the top of the must and acquire the shape of few colored stripes painted onto the sails. They run under the assumption that the stripes remain in a plane parallel to the deck which can be questionable especially for downwind sails.
Thus, after a brief analysis of different non-contact techniques, reported in Chapter 3, Time Of Flight methodology has been selected. The acquisition device encloses a TOF sensor by Sick, a motor reduction unit, a web-cam and other auxiliary sensors. It provides 3D measurements of the entire sail surface, no matter the sail type neither the flying shape assumed.
This kind of TOF sensors were originally design for safety application, and a detailed metrological qualification was necessary. The systematic error rises increasing the distance between target and sensor, but remains an acceptable value for our application. In fact, the maximum percent error retrieved is 0.3% of the acquired distance. Moreover, the statistical error remains almost constant, around the value of 4 mm, within the analyzed range (1 to 15 meters). The presence of a critical incident angle between the laser ray direction and the normal to the target surface has been worked out: incident angles greater than 70° lead to a significant drop in the measurement accuracy. Outdoor tests have been performed highlighting the influence of the ambient light: measurement standard deviation can increase by the 350% while the sun hits directly the sensor - the sensor is dazzled -. In addition, different target materials have been tested, proving that the sail tissue presents the widest uncertainty due to its transparent nature; and other tests, on the influence of the target color, suggested light gray colored object reflect the signal the best. Other details about the device and its metrological qualification work are reported in [61].

Point cloud data elaboration

Once data have been acquired, they are stored as point cloud structure and consequently elaborated as described in Chapter 5. A dedicated software has been realized using open PCL source libraries, programming in C++ and Matlab language, in Qt environment. Each step of the post processing procedure has been discussed and exam-

ples of the elaboration on synthetic data are reported to verify the algorithm developed. The first step is the cloud registration. In fact, each sail is scanned by a different acquisition unit. This leads to the need for registering different scans in a common coordinate system, in order to compose an unique sail plan of the yacht. To perform this operation a specific algorithm has been developed, based on standard algorithm as SVD or ICP, but relying also on information taken from the acquired overall scene. This provide a better results and turned to be a more robust method with respect to acquisition noise. Then, registered scenes have to be interpreted, that means recognizing different objects in the point cloud and organizing them in distinct clusters. The final goal is the extraction of the point cluster that represents the sail surface. Proximity to other points and normal trends are exploited as discriminant criteria. Unwanted points might remain after the segmentation step due to objects close to the sail surface, such as the boom, shrouds and spreaders or due to errors in the acquisition process, such as the mixed pixels - explained in Chapter 4. The sail cluster is then filtered to face this problem. Sometimes user intervention is required.
To reduce the influence of the acquisition noise a cloud resampling is applied before reconstructing the sail surface. Resampling is substantially a local interpolation of the data that lead to a more smooth surface. A heavy smoothing would alter the overall sail shape, which is obviously unfair. Evaluating this resampling effect is not an easy task, as the threshold for smoothing should be defined by displacements from a reference surface, which is not known as it is the final object searched in this thesis. For the current work, the smoothness operation is fixed as for avoiding loss of samples and point displacements from their original position greater than the value of measurement uncertainty at that distance estimated in the metrological qualification of the device.
Lats step is the 3D surface reconstruction. A custom algorithm has been developed. It is based on Delaunay triangulation but not used in a 3D domain since it would create tetrahedrons instead of surfaces. 3D points are projected in a 2D domain, where a mesh is computed. The two dimensional mesh is then re-projected in the three-dimensional space, providing a point approximating surface.

Shape analysis and performance evaluation

As last step of the process, the reconstructed sail is analyzed and several geometric parameters are computed to give at a glance an idea of the way the sail is flying. The 3D sail surface is cut by means of sectional planes computed in order to be directed as the surface normal evaluated in correspondence of the maximum camber point. The pose

of these planes is sought iteratively and the process might takes few seconds. This way it is possible to avoid the assumption that the sail sections have to belong to planes parallel to the yacht deck. Finally, these parameters and these shapes can be correlated to measurements of forces or pressures to complete the yacht performances analysis as shown in the Chapter 6 reporting few case studies for acquisition on scale models and on field on the Sailing Laboratory Boat. Good results have been obtained. However, various improvements can be considered.

FUTURE WORKS

Concerning the hardware setup, a redesign of the TOF design could be considered to achieve a better result in terms of measurement accuracy, especially trying to avoid mixed pixels and loss of data for direct sun light, especially in the higher part of the sails. In fact, these lack in retrieving point coordinates impairs the algorithm results and user intervention is required to check each pipeline step.
Improvements can be also addressed to a redesign of the acquisition unit with particular attention to the overall dimension. It was not so comfortable sailing with bulky devices on the deck.
Concerning the software, instead, a faster code might provide results almost in real time - the code takes few minutes at the moment -. Dealing with point clouds means dealing with thousands of points, i.e. thousands of information to manage. The ideal improvement would be programming using GPU to parallel different processes.
Moreover, pattern matching algorithm could be taken into consideration. They stat from a CAD model of an object and search for points in the cloud that would match it. It is widely used in the industrial field for rigid objects, but too few has been done for deformable shapes. The idea is to use the design sail shape and to look for it in the cloud ease some constrain on the rigidity of the model. Few tests have been done but not successfully results have been obtained so far.

APPENDIX

A

POINT CLOUD BASIC CONCEPTS

POINT CLOUD DEFINITION

A point cloud is a data structure used to represent a collection of multi-dimensional points. Commonly, a point cloud is a three-dimensional set that encloses the spatial coordinates of an object sampled surface; but adding information about the color, the point cloud becomes four-dimensional. In Figure A.1 an example of 3D point cloud representing a spinnaker sail is presented.

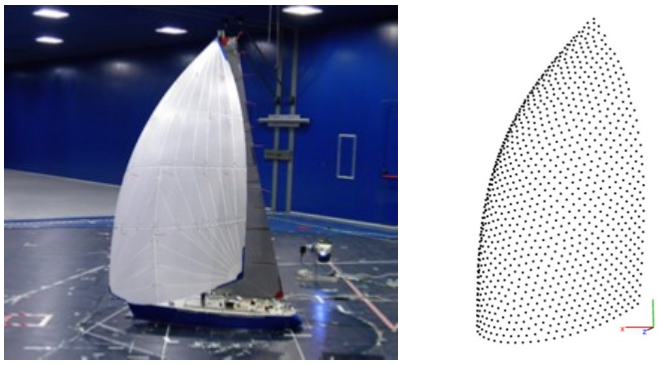


Figure A.1: Example of 3D point cloud representing a spinnaker sail

NEIGHBORHOOD CONCEPT

The neighborhood of a point p is the set of its closest points, according to the Euclidean distance, and with a limit criterion. Points belonging to the neighborhood set are called neighbors. There are two types of neighborhood:

1. *r-neighborhood* : the limit criterion is a distance threshold, i.e. all the points inside a sphere centered in the considered point and of radius r are considered neighbors ;
2. *k-neighborhood*: the limit criterion is the number of neighbors to consider, i.e. only the k-th closest points to p are its neighbors;

The neighborhood is not an object which is an end to itself, but it is used as initial step for retrieving further object information.

POINT CLOUD NORMALS

A surface normal, or simply normal, associated to a point p, is a vector perpendicular to the tangent plane that approximates the surface

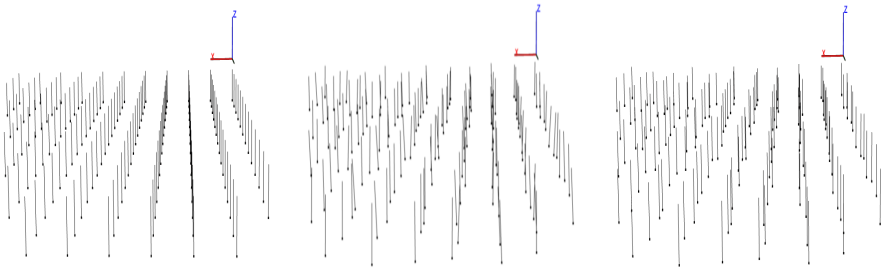
around p. The problem of determining the normal to a point on the surface is approximated by the problem of estimating the normal of a plane locally tangent to the surface, which in turn becomes a least-square plane fitting estimation problem. The solution for estimating the surface normal is therefore reduced to an analysis of the eigenvectors and eigenvalues (through the Principal Component Analysis -PCA-) of a co-variance matrix created from the neighborhood of the query point. For each point *pi*, the co-variance matrix C is constructed as:

$$C = \frac{1}{k}\sum_{i=1}^{k}(p_i - \overline{p})(p_i - \overline{p})^T, \quad C \cdot \overrightarrow{v_j} = \lambda_j \overrightarrow{v_j}, \quad j\epsilon[0,1,2] \qquad (A.1)$$

where: k is the number of point neighbors considered in the neighborhood of *pi*, $\overline{p}$ represents the 3D barycenter of the nearest neighbors, λ_j is the j-th eigenvalue of the co-variance matrix, and $\overrightarrow{v_j}$ the j-th eigenvector. The normal $\overrightarrow{n_i}$ is the eigenvector corresponding to the smallest eigenvalue. $\overrightarrow{n_i}$ gives the direction of the normal, however there is no mathematical way to solve for the sign of the normal, so the orientation results ambiguous. This means that, for a same surface, some normals are oriented inwards and other outwards. A simple solution, proposed in [41], is to align all the normals orientations towards the origin of the axis. This is especially reasonable under the assumption that the point cloud comes from an acquisition device, which is placed in the origin of the axis. The choice of the neighborhood have a great influence in the output normals. The minimum number of neighbors of a point p needed to retrieve its normal is three, because a plane can be retrieved from a minimum of three samples. Using small neighborhood lead to the creation of normals which explain the local behavior of the surface, while a big neighborhood creates normals that represent the general trend of the surface. The size of the neighborhood has to be balanced in function of the point cloud considered. The presence of noise on the samples lead to imprecision in the normals value if the neighborhood is too small, as visible from Figure A.2.

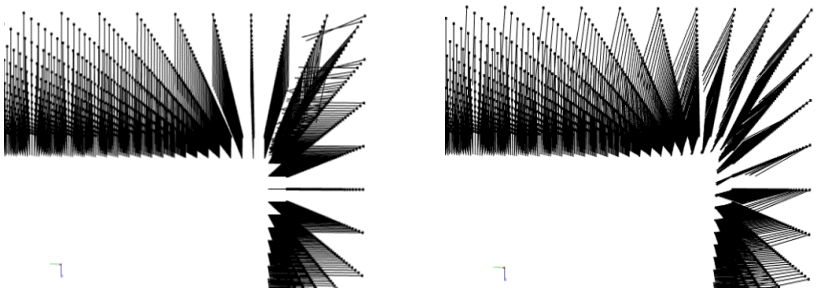
While a large neighborhood may help in reducing noise, on the other hand, it may filter some important information about object local features, such as sharp edges or corners. They might be smoothed as shown in Figure A.3 where normal directions change more gradually around the corner. More in detail, a synthetic point cloud representing a 90° angle between two planar surfaces has been generated and the normals were computed using two different neighborhood sizes (small k=5 and large k=100). Comparing the results, one can notice that in the second case normals do not follow the original surface, but tend to anticipate the 90° change.

The same results displayed in the figures above can be obtained using a r-neighborhood. However the radius has to be adapted to the



(a) (b) (c)

Figure A.2: Normals estimation for a planar surface with or without noise using k-neighborhood of two different sizes: k-neighborhood of bigger size filters out the noise.



(a) (b)

Figure A.3: Normals estimation for a 90° corner with k-neighborhood of two different sizes: k-neighborhood of bigger size reduce the sharpness of the corner edges.

density of the point cloud, which has to be known in advance and has to be constant in the whole data set. These conditions are not always verified, so the use of the k-neighborhood is usually more robust. Normals are used in most of the processing methods as basic information, along with the spatial coordinates of the points. Whenever their use is requested, it should be reminded that they are not unique values, but depend on the neighborhood chosen, which then acts as an additional parameter to take into account.

SURFACE CURVATURE

Intuitively, curvature is the amount by which a geometric object deviates from being flat. Rigorously, the curvature of a point p is a scalar number, found as:

$$\sigma = \frac{\lambda_0}{\lambda_0 + \lambda_1 \lambda_2} \tag{A.2}$$

where: λ_i are is the i-th eigenvalue of the co-variance matrix created from the neighborhood of p, as explained in the previous section. The eigenvalues are ordered in ascending order, so λ_0 is always the smallest eigenvector.

K-D TREE

The K-d tree (k-dimensional tree) is a data structure used in computer science for organizing points in a space with k dimensions. It is a binary search tree with additional constraints imposed. The algorithm iteratively splits the space into two parts and builds a tree structure (Figure A.4). Each point of the cloud is associated to a bounding box of the last level that encloses it, but thanks to the tree structure it is linked to all other larger boxes of the higher levels. K-d trees are, thus, very useful for range and nearest neighbor searches.

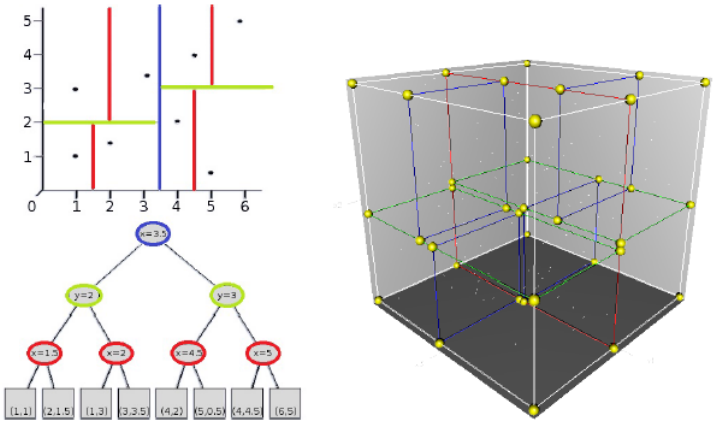


Figure A.4: 2-d tree visualization and corresponding tree structure (left) and 3-d tree visualization - the first division is in red, the second in green and then blue

An octree is a tree-based data structure for managing sparse 3-D data, with the same purpose of the K-d tree. The difference is that each node of the octree has exactly eight children, instead of two. Dividing the space in eight children means that the cubic bounding box of the data is split in eight equal cubes iteratively for each level (Figure A.5).

SINGULAR VALUE DECOMPOSITION

The singular value decomposition (SDV) of a real or complex matrix $m \times n$ M is a factorization in the form:

$$M = U\Sigma V^T \tag{A.3}$$

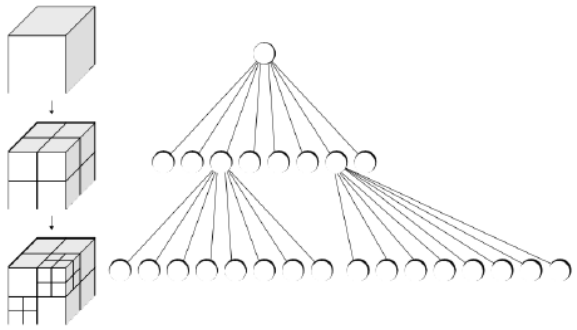


Figure A.5: Octree Visualization and corresponding tree structure.

where U is an $m \times m$ real or complex unitary matrix, Σ is an $m \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal, and V^T is an $n \times n$ real or complex unitary matrix. The diagonal entries σ_i of Σ are known as the singular values of M, from which the name of the method.
When M is an $m \times m$ real square matrix with positive determinant, U, V^T, and Σ are real $m \times m$ matrices as well, Σ is a scaling matrix, and U, V^T are rotation matrices. Imposing a unitary scale factor, U and V^T define the registration matrix [28], which can be expressed as a unique rotation and a translation, obtained as:

$$R = V \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det\left(V \cdot U^T\right) \end{bmatrix} \cdot U^T \tag{A.4}$$

$$T = p_2 - R \cdot p_1 \tag{A.5}$$

where $p_2$and $p_1$are the center of mass, respectively, of the target data-set and the source data-set.
For the above, it is easy to understand that one of the most popular point cloud registration methods is based on the SVD of a matrix containing information about 3D coordinates of corresponding points in two different clouds [27].
The input matrix M, in fact, is composed of other two matrices A and B as follows:

$$\bar{\mu_x} = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad \bar{\mu_y} = \frac{1}{N}\sum_{i=1}^{N} y_i \qquad \bar{\mu}_z = \frac{1}{N}\sum_{i=1}^{N} z_i \tag{A.6}$$

$\bar{\mu_R} = [\bar{\mu_x}, \bar{\mu_y}, \bar{\mu_z}]$ is the baricenter of the reference point cloud, and $\bar{\mu_F}$ the baricenter of the registering cloud.

$$A = \begin{bmatrix} x_1^R - \bar{\mu_{Rx}} & y_1^R - \bar{\mu_{Ry}} & z_1^R - \bar{\mu_{Rz}} \\ .. & .. & .. \\ x_m^R - \bar{\mu_{Rx}} & y_m^R - \bar{\mu_{Ry}} & z_m^R - \bar{\mu_{Rz}} \end{bmatrix} \tag{A.7}$$

$$B = \begin{bmatrix} x_1^F - \bar{\mu_{Fx}} & y_1^F - \bar{\mu_{Fy}} & z_1^F - \bar{\mu_{Fz}} \\ .. & .. & .. \\ x_m^F - \bar{\mu_{Fx}} & y_m^F - \bar{\mu_{Fy}} & z_m^F - \bar{\mu_{Fz}} \end{bmatrix} \tag{A.8}$$

$$M = B * A^T \tag{A.9}$$

The SVD procedure requires 3D coordinates of at least three ordered pairs of corresponding points, and it is not influenced by the number of points in each cloud, neither by their initial relative position.
The choice of the points to match between the two clouds is crucial: if not corresponding points are selected, a wrong registration occurs. Most of the times, the user manually selects the corresponding points in the clouds. The presence of noise in the data-sets influences the algorithm results, as well as the relative position of the point selected: choosing points too close to each others might lead to the calculation of singular matrices, which is obviously unwanted. It is usually a good practice to choose the samples as spaced out as possible.

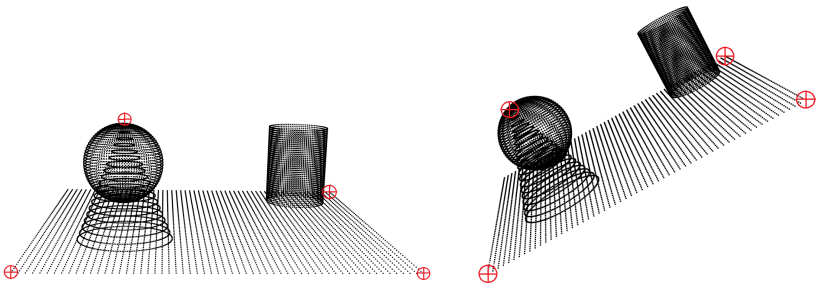


Figure A.6: Cloud registration: reference cloud (left) and rotated cloud (right); points selected for the registration are highlighted in red.

SVD tests on artificial point clouds

The SVD algorithm has been tested for the artificial scene described in Figure A.6, where a rotation of 45° around one axis was imposed to the original cloud. Four matching pairs of points have been manually

selected from the input data-sets (highlighted in red in the figures), and the rigid registration has been computed. The resulting transformation matrix is reported in A.10.

$$\begin{bmatrix} 0.707 & 0 & 0.707 & 0 \\ 0 & 1 & 0 & 0 \\ -0.707 & 0 & 0.707 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0.707 & 0 & 0.707 & -0.001 \\ 0 & 1 & 0 & 0.001 \\ -0.707 & 0 & 0.707 & -0.003 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.10}$$

Table A.1: Comparison between transformations matrices for noise free data-sets: imposed transformation (left) vs SVD estimated transformation (right).

The transformation matrices correspond almost perfectly, aside from minimal errors, due to numerical operations performed by the machine.

A.0.0.1 *Noise effect*

The noise effect is evaluated in the following test: the same scene presented in Figure A.6 is considered, but noise is added to each point. The level of noise is adequate to what found in the metrological qualification of the acquisition system (see Section 4.2). The noise values follow a Gaussian distribution with mean equal to zero and standard deviation $\sigma = 0.05$ (a.u.). So that, the maximum level of noise (3σ) is equal to 0.5 % of the distance between the origin of the reference system and the furthest point in the synthetic data set.

$$\begin{bmatrix} 0.707 & 0 & 0.707 & 0 \\ 0 & 1 & 0 & 0 \\ -0.707 & 0 & 0.707 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0.700 & 0.006 & 0.714 & 0.036 \\ -0.002 & 1 & 0.007 & 0.281 \\ -0.714 & -0.004 & 0.700 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.11}$$

Table A.2: Comparison between transformations matrices noisy data-sets: imposed transformation (left) vs SVD estimated transformation (right).

As expected, the SVD based procedure performs worse than the previous case since there is not perfect correspondence between the sampled points in the two clouds (three plane corners and cone vertex) due to the noise. The relative pose is correctly estimated, as the rotation part of the matrix is right. However the error on the trans-

lation is notable, 56 % of the spatial resolution of the reconstructed cloud.

A.0.0.2 *Number of point effect*

Another test was done in order to verify that the procedure is not influenced by the total number of points in the clouds to be registered. In this case the point cloud has been cut and rotated 45° counterclockwise, as visible in Figure A.7.

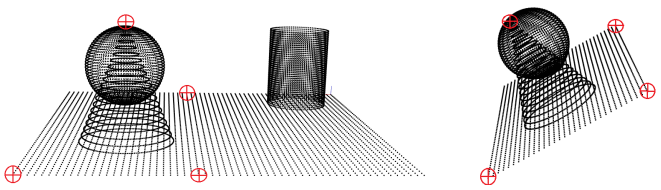


Figure A.7: Partial cloud registration: reference cloud (left) and partial rotated cloud (right); points selected for the registration are highlighted in red.

$$\begin{bmatrix} 0.707 & 0 & 0.707 & 0 \\ 0 & 1 & 0 & 0 \\ -0.707 & 0 & 0.707 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0.707 & 0 & 0.707 & 0.005 \\ 0 & 1 & 0 & -0.001 \\ -0.707 & 0 & 0.707 & 0.006 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{A.12}$$

Table A.3: Comparison between transformations matrices for poor overlapping data-sets: imposed transformation (left) vs SVD estimated transformation (right).

As expected, the presence of few points does not impair the registration, whenever correspondences are perfectly known.

ITERATIVE CLOSEST POINT

The ICP approach, instead, has been proposed by Besl [32] and Zhang [33], and consists of an iterative descend method which tries to find the optimal transformation (rotation matrix R and translation vector T) between two data sets $p_i \in P1$ and $q_j \in P2$ by minimizing the Euclidean distance error metric between their overlapping areas.

$$\min \sum_{i=1}^{n} \|R \cdot p_i + T - q_i\|^2 \tag{A.13}$$

Modified algorithms, considering not only 3D spatial information, have been implemented: Johnson et al. [34] takes into account the colors acquisition as well, Eggert et al. [35] performs the simultaneous registration of multiple range views, Declerck et al. [36] carries out 3D-2D projective transformations. Ristic et al. [37] instead proposes to use a triangular mesh model approximation to accelerate the algorithm and suggests to assume the standard uncertainty of the device used to perform the measurement as tolerance to terminate the ICP iterations.
Whichever the approach considered, the ICP performs the following steps:

1. The set Q of closest points between the source data-set S_d and the target data-set T_d is identified
2. The transformation matrices (R and T), minimizing the chosen cost function over Q, are determined.
3. The transformation is applied to the measured points set as $S_d^{'} = R * S_d + T$.
4. If change in the cost function is greater than a preset value the procedure is repeated from the point 1.
5. Else the procedure ends.

It has been proved that ICP algorithm converges always to a local minimum. This means that the algorithm is able to provide the optimal solution only if the relative position of the cloud data set is close. In order to better understand the reason beyond this, step 2 has to be further investigated. For each iteration, the procedure estimates the best rotation matrix R, performing a singular value decomposition of the matrix M = B*A' (see Appendix section A for details) containing data belonging to a cloud A and the relative closest points in cloud B. Moreover, the score function can be expressed as function of the cloud centers of mass, μ_{corr} and μ_{reg} and that the optimal translation vector applied to the registering cloud at every iteration can be written as:

$$\overrightarrow{q_t} = \mu_{corr} - R \cdot \mu_{reg} \tag{A.14}$$

The construction of the correspondence set B is the critical step: for each point of the cloud to register the closest point (in terms of euclidean distance) of the model cloud is assigned as correspondent. If the clouds are poorly aligned, a model point could be matched to several points of the registering data-set. This leads to the creation of a correspondence set composed of several identical points, which, in turn, leads to incorrect transformation matrices. For every iteration

step the registering cloud is first rotated and then its center of mass is superimposed on the correspondence set center of mass, as A.14. If the correspondence set is not correctly built (i.e. bad initial position) the translation moves the registering cloud not toward the model center of mass. This behavior is not bad in itself, as it still moves the cloud closer the model, but the subsequent ICP iteration might reach a local minimum ending the procedure prematurely. In other words, this means that the algorithm won't move anymore the data, because any movement would lead to an increasing of the cost function, although globally another minimum (i.e. better relative position) exists.

This issue is well known and discussed in many works, as [62], [63], and [64]. In the latest Pottmann et al. analyzed the problem and proposed a procedure that takes into account also information about the point normals and obtains the transformation matrices through an instantaneous kinematics approach rather than SVD. The use of instantaneous kinematics, however, only simplified the processing operation of the procedure proposed by Chen et al. in [63]. They apply a motion to the cloud such that the sum of the squared distances from the reference cloud is minimal. For each point x, its velocity vector is defined as:

$$\mathbf{v}(x) = \bar{\mathbf{c}} + \mathbf{c} \times \mathbf{x} \tag{A.15}$$

where $\bar{\mathbf{c}}$ is the velocity vector of the origin of the reference coordinate system and $\mathbf{c}$ represents the vector of angular velocity (or Darboux vector). Up to the first differentiation order, every three-dimensional movement can be expressed locally as translation with constant velocity ($\bar{\mathbf{c}} = 0$), a uniform rotation around an axis ($\mathbf{c} \cdot \bar{\mathbf{c}} = 0$) or an uniform helical motion ($\mathbf{c} \cdot \bar{\mathbf{c}} \neq 0$). A generic velocity vector field is hypothesized, such that the corresponding velocity vectors $v_i(x)$ for each point minimize the quadratic cost function F:

$$\min \sum_{i=1}^{N} F_i(\mathbf{x_i} - \mathbf{v}(\mathbf{x_i})) \tag{A.16}$$

From the velocity vector field a transformation that displaces the points $\mathbf{x}_i$in the same way as the velocity vector would do is calculated, through a linearization of the motion. The set of correspondences is required as before, but also the normals $\mathbf{n_i}$of the corresponding points are calculated and stored. The motion is considered as general as possible, so it would be a helical motion in the form A.15. The generic distance from the point $\mathbf{x}_i$ to its matching point $\mathbf{y}_i$ can be expressed as function of their initial distance d_i and the velocity vector as:

$$d_i + \mathbf{n}_i \cdot (\bar{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i) \tag{A.17}$$

the cost function (A.16) can be written as:

$$\min \sum_{i=1}^{N} (d_i + \mathbf{n}_i \cdot (\bar{\mathbf{c}} + \mathbf{c} \times \mathbf{x}_i))^2 \tag{A.18}$$

The minimization can be solved using a system of linear equations, rewriting A.17 as:

$$d_i + \mathbf{n}_i \cdot \bar{\mathbf{c}} + (\mathbf{x}_i \times \mathbf{n}_i) \cdot \mathbf{c} = d_i + (\mathbf{x}_i \times \mathbf{n}_i, \mathbf{n}_i) \begin{pmatrix} \mathbf{c} \\ \bar{\mathbf{c}} \end{pmatrix} = d_i + A_i C \tag{A.19}$$

and A.18 as:

$$F(C) = \min \sum_{i=1}^{N} (d_i + A_i C)^2 = \sum_i d_i^2 + 2 \sum_i d_i A_i C + \sum_i C^T A_i^T A_i C \tag{A.20}$$

$$= D + 2B \cdot C + C^T A C$$

The unique minimum of the quadratic cost function is a column vector C, with six entries (since $A_i$is, by construction, a general positive six-by-six matrix) which solves the linear system :

$$AC + B = 0 \tag{A.21}$$

where $A = A_i^T A_i$ and $B = 2 \sum d_i A_i$ can be obtained from the registering points and their corresponding point normals. The rotation and translation matrices can be recovered from the elements of C, with the relations A.0.0.2 and A.0.0.2, since at the start of the procedure, an helical motion was supposed. For more details on helical motion refer to [65].

$$R = \begin{bmatrix} c2 * c3 & c3 * s1 * s2 - c1 * s3 & c1 * c3 * s2 + s1 * s3 \\ c2 * s3 & c1 * c3 + s1 * s2 * s3 & c1 * s2 * s3 - c3 * s1 \\ -s1 & c1 * s1 & c1 * c2 \end{bmatrix} \tag{A.22}$$

where $\begin{matrix} c1 = \cos(C(1)) & c2 = \cos(C(2)) & c3 = \cos(C(3)) \\ s1 = \sin(C(1)) & s2 = \sin(C(2)) & s3 = \sin(C(3)) \end{matrix}$

$$T = \begin{bmatrix} C(4) \\ C(5) \\ C(6) \end{bmatrix} \tag{A.23}$$

Visually speaking, in the first ICP algorithm presented [32] the cost function minimizes the distance from points in the cloud to register and their correspondences, while the second one minimizes the distance between the point and the target surface (i.e. the distance from the plane tangent to the target surface in the correspondence point). This is why the first is usually referred as *point-to-point* ICP, while the other as *point-to-plane* ICP. Both of them rely on the construction of the correspondence set, which is a computational expensive step, since it has to find a correspondence for each point of the cloud to register, comparing it to every point of the reference cloud.
In [41], Rusu proposed an alternative construction of the correspondences set, through the use of the Point Feature Histograms. Briefly, PFH are histograms of the distribution of the normals in a point neighborhood; they can identify peculiar points, like corner points, edge points, points lying on a plane or on a curved surface. The assumption is that corresponding points from different data-sets should have at least similar PFH (feature persistence hypothesis) and that it is not necessary to compare the whole data-sets, but only the points whose PFH are less common.
For example, for registering two cones, it should be sufficient to match the vertexes and the lowest circumference edge points. In this way, the computational cost of matching points significantly drops, since number of the points compared are usually at least cut in half with respect to a classical ICP method. For this implementation, a registering point p_i is matched with another one q_i from the target cloud, selecting it randomly from the list of points Q whose PFH are similar to p_i PFH. Once the points has been matched, a point-to-plane registration is performed, using the cost function:

$$\min \sum_{i=1}^{N} \left\| (R \cdot \mathbf{p}_i + T - \mathbf{q}_i) \cdot \mathbf{n}_{q_i} \right\| \tag{A.24}$$

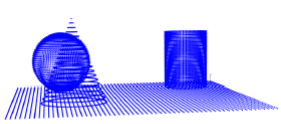
which is comparable to A.18. The random selection of the correspondences might seem suboptimal, but it is a decision made to speed up the computational time. This algorithm then proceeds to complete the registration with the same steps explained for the point-to-plane ICP. Both [32] and [33] found that the algorithm has fast convergence (i.e. large transformation) in the first iterations but it is strongly dependent from the initial position. Thus, it is often referred to as a *finely registration algorithm*, which comes after a first rough alignment. All the algorithms perform an iteration until an end condition is verified; most of the time this condition is represented by a maximum number of iterations allowed or a minimum cost function value to reach that estimates the convergence in the procedure.

ICP tests on artificial point clouds

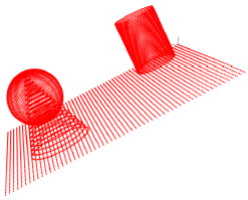
Tests have been conducted on point cloud representing the same scene with regular geometry objects as in the Section A, Figure A.6. Then, a rotation of 30° around the Y axis has been imposed to the original point cloud, as visible in Figure A.8b, and the registration via different ICP algorithms has been performed, retrieving the final transformation matrix. This matrix can be seen as the composition of a translation (represented by the contribute of the fourth column, row one to three), and a rotation, given by the 3×3 up left sub-matrix.

Comparison of different type of ICP algorithms

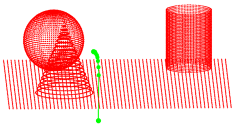
The tests reported below compare the performances of different ICP algorithms, both in terms of alignment result and time consumption. The values shown are retrieved from tests performed on the same computer, which have Intel Core 2 CPU @ 2.40 GHz on a 32-bit Linux operative system.



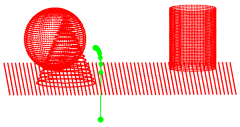
(a) Cloud 1



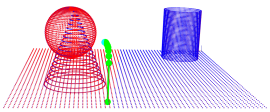
(b) Cloud 2



(c) ICP point-to-point registration: perfect alignment.



(d) ICP point-to-plane registration: perfect alignment.



(e) PFH ICP registration - Clouds very close to each other

Figure A.8: Artificial point clouds before (up) and after (down) registration

As shown in Figure A.8c, all the ICP procedures provide the correct alignment; the green lines and dots highlight the path and the position covered by the center of the cloud to register during the registration procedure at each iteration.
Moreover, all the algorithm were tested setting as end iterative condition a fixed number of iteration (25 in this case) or a relative mean error between two consecutive iterations (set at 10^{-3}). The mean error is the squared sum of the distance for each point from its correspondence divided by the number of points considered. When the difference between the mean error in two consecutive iterations is smaller than a threshold, it means that the algorithm is near a local minimum

and convergence is reached. Further iterations would not improve the resulting output transformation.

	Point	Plane	PFH
Iterations	25	25	25
Time [s]	150	226	7
Relative Error	10^{-16}	10^{-18}	10^{-9}

Table A.4: ICP algorithms comparison for scenes described in Figure A.8 - Iteration Number Ending Condition

	Point	Plane	PFH
Iterations	15	7	13
Time [s]	90	66	4.2
Relative Error	10^{-5}	10^{-5}	10^{-4}

Table A.5: ICP algorithms comparison for scenes described in Figure A.8 - Relative Error Ending Condition

The performances are compared with two different tests, in the first, whose results are showed in Table A.5, all the algorithms are ended after 25 iterations. In the second test, the ending condition is chosen as the relative mean error between two consecutive iterations.

RANDOM SAMPLE CONSENSUS PARADIGM

The RANSAC algorithm starts from the assumption that in the data-set considered there is a subset of points that can be explained through a mathematical model. Data can then be divided in inliers, i.e. points that well suit the considered model (accepting a certain tolerance), and outliers, i.e. points that do not suit it. The algorithm could be divided in three main parts: extraction of the minimal point set to generate a model for the shape sought, shape estimation, and assignment of a score for the shape identified. First of all, the algorithm extracts a minimal subset composed of three points, with their corresponding normals, $<(p_1,n_1),(p_2,n_2),(p_3,n_3)>$. The first point $p_1$of the minimal subset is picked randomly in the whole data set. Then, after computing an octree spatial subdivision - extremely useful in the research of neighbor points, (see Section A) - , a box C (or a level of the tree) containing p1 is selected. $p_2$and $p_3$are chosen within the same box C. The algorithm, in fact, works under the hypotheses that the shape sought is a local phenomenon in a wider set of data and so a shape created from close points has more probability to be the right one instead of one built from distant points. Moreover, choosing C from a proper level of the octree is an important aspect. The smaller

the C dimension is, the greater the probability that the points in the minimal subset correspond to the same shape. For sure, the density of the cloud must be taken into account and the algorithm can automatically adapt to point cloud with non-uniform density, which is especially convenient in this study.
Once the subset is extracted, the model is computed. Below the description of the procedure to create different geometric models starting from the points of the subset [39]:

1. *Planes*: for a plane <p_1,p_2,p_3> constitutes a minimal set when not taking into account the normals in the points. To confirm the plausibility of the generated plane, the deviation of the plane's normal from n_1, n_2, n_3 is determined and the candidate plane is accepted only if all deviations are less than the predefined angle α.

2. *Cylinders*: To generate a cylinder two points with normals are selected. First the direction of the axis is established with a = n1 x n2. Then the two parametric lines p1 +tn1 and p2 +tn2 are projected along the axis onto the ax = 0 plane and take their intersection as the center c. The radius is set to the distance between c and p1 in that plane. The cylinder is accepted as a shape candidate only if the points are within a distance of ε of the cylinder and their normals do not deviate by more than α degrees.

3. *Cones*: Although the cone, too, is fully defined by two points with corresponding normals, for simplicity three points and normals are considered in its generation. The position of the apex c, is derived as the intersection of three planes defined by the point and normal pairs. Then, the normal of the plane defined by the three points gives the direction of the axis a. Now the opening angle ω is given by

Once created the model, data points are tested and flagged as inliers or outliers with respect to the parameter ε representing the maximum discrepancy acceptable between a point and the model constrain, and to the parameter α representing the maximum angular discrepancy acceptable between point and model normal directions. Repeating these steps leads to different identified shapes, each formed starting by different minimal subsets. The final step is the evaluation of the best shape extracted, i.e. the shape that provides the maximum number of inliers, and thus, the highest score function.

BIBLIOGRAPHY

[1] David G Kendall. The diffusion of shape. *Advances in applied probability*, pages 428–430, 1977. (Cited on page 7.)

[2] Richard J Campbell and Patrick J Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81(2):166–210, 2001. (Cited on page 7.)

[3] P Besl. Machine vision for three-dimensional scenes. *The Free-form Surface Matching Problem*, pages 25–71, 1990. (Cited on page 8.)

[4] Wenzhen Huang, Zhenyu Kong, Dariusz Ceglarek, and Emilio Brahmst. The analysis of feature-based measurement error in coordinate metrology. *IIE Transactions*, 36(3):237–251, 2004. (Cited on page 9.)

[5] François Blais. Review of 20 years of range sensor development. *Journal of Electronic Imaging*, 13(1), 2004. (Cited on page 9.)

[6] Yi Ma, Stefano Soatto, Jana Kosecka, and S Shankar Sastry. *An invitation to 3-d vision: from images to geometric models*, volume 26. Springer Science & Business Media, 2012. (Cited on page 9.)

[7] Miles Hansard, Seungkyu Lee, Ouk Choi, and Radu Patrice Horaud. *Time-of-flight cameras: principles, methods and applications*. Springer Science & Business Media, 2012. (Cited on page 12.)

[8] Edoardo Charbon, Matt Fishburn, Richard Walker, Robert K Henderson, and Cristiano Niclass. Spad-based sensors. In *TOF Range-Imaging Cameras*, pages 11–38. Springer, 2013. (Cited on page 12.)

[9] Fabio Remondino and David Stoppa. *TOF range-imaging cameras*, volume 68121. Springer, 2013. (Cited on page 14.)

[10] Les Piegl and Wayne Tiller. *The NURBS book*. Springer Science & Business Media, 2012. (Cited on page 17.)

[11] R Ranzenbach and J Kleene. Utility of flying shapes in the development of offwind sail design databases. In *2nd High performance Yacht Design Conference, Auckland*, 2002. (Cited on page 20.)

[12] F Fossati, F Martina, and S Muggiasca. Experimental database of sail performance and flying shapes in upwind conditions. In *Proc. International Conference on Innovation in High Performance*

Sailing Yachts, Lorient, France, May, pages 29–30, 2008. (Cited on pages 20 and 103.)

[13] Kai Graf and Olaf Müller. Photogrammetric investigation of the flying shape of spinnakers in a twisted flow wind tunnel. In *The 19th Chesapeake Sailing Yacht Symposium, Annapolis, MD*, 2009. (Cited on page 21.)

[14] Hannes Renzsch and Kai Graf. An experimental validation case for fluid-structure interaction simulations of downwind sails. In *21st Chesapeake Sailing Yacht Symp*, 2013. (Cited on page 21.)

[15] Y Tahara, Y Masuyama, T Fukasawa, and M Katori. Cfd calculation of downwind sail performance using flying shape measured by wind tunnel test (summaries of papers published by staff of national maritime research institute at outside organizations). 12(1):49, 2012. (Cited on page 21.)

[16] GF Clauss and W Heisen. 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*, page 87, 2006. (Cited on page 21.)

[17] Y Masuyama and Toichi Fukasawa. Full-scale measurements of sail force and validation of numerical calculation method. In *Proc. 13th Chesapeake Sailing Yacht Symposium, Annapolis, MD, Jan*, volume 25, pages 23–36, 1997. (Cited on pages 22 and 114.)

[18] DJ Le Pelley and O Modral. 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*, pages 2–4, 2008. (Cited on page 22.)

[19] J Deparday, P Bot, F Hauville, D Motta, DJ Le Pelley, and RGJ Flay. Dynamic measurements of pressures, sail shape and forces on a full-scale spinnaker. In *23rd International HISWA Symposium on Yacht Design and Yacht Construction, November 17th-18th, Amsterdam*, 2014. (Cited on page 23.)

[20] Mathieu Salzmann, Julien Pilet, Slobodan Ilic, and Pascal Fua. Surface deformation models for nonrigid 3d shape recovery. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(8):1481–1487, 2007. (Cited on page 24.)

[21] Sick intelligence. Accessed: 2016-01-10. `http://www.sick.com/group/EN/home/Pages/homepage1.aspx`. (Cited on page 27.)

[22] Cang Ye and Johann Borenstein. Characterization of a 2-d laser scanner for mobile robot obstacle negotiation. In *ICRA*, pages 2512–2518, 2002. (Cited on pages 32 and 40.)

[23] Chan-Soo Park, Doik Kim, Bum-Jae You, and Sang-Rok Oh. Characterization of the hokuyo ubg-04lx-f01 2d laser rangefinder. In *RO-MAN, 2010 IEEE*, pages 385–390. IEEE, 2010. (Cited on page 32.)

[24] Majd Alwan, Matthew B Wagner, Glenn Wasson, and Pradip Sheth. Characterization of infrared range-finder pbs-03jn for 2-d mapping. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 3936–3941. IEEE, 2005. (Cited on pages 32 and 40.)

[25] Kyeong-Hwan Lee and Reza Ehsani. Comparison of two 2d laser scanners for sensing object distances, shapes, and surface patterns. *Computers and electronics in agriculture*, 60(2):250–262, 2008. (Cited on page 35.)

[26] Albert Diosi and Lindsay Kleeman. Uncertainty of line segments extracted from static sick pls laser scans. In *SICK PLS laser. In Australiasian Conference on Robotics and Automation*, 2003. (Cited on page 40.)

[27] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3-d point sets. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (5):698–700, 1987. (Cited on pages 50 and 133.)

[28] Inge Söderkvist. Using svd for some fitting problems. *University Lecture*, 2009. (Cited on pages 50 and 133.)

[29] Ivan Sipiran and Benjamin Bustos. Harris 3d: a robust extension of the harris operator for interest point detection on 3d meshes. *The Visual Computer*, 27(11):963–976, 2011. (Cited on page 52.)

[30] Alex Flint, Anthony Dick, and Anton Van Den Hengel. Thrift: Local 3d structure recognition. In *dicta*, pages 182–188. IEEE, 2007. (Cited on page 52.)

[31] Yu Zhong. Intrinsic shape signatures: A shape descriptor for 3d object recognition. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 689–696. IEEE, 2009. (Cited on page 52.)

[32] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992. (Cited on pages 52, 136, and 140.)

[33] Zhengyou Zhang. Iterative point matching for registration of free-form curves. 1992. (Cited on pages 52, 136, and 140.)

[34] Andrew Edie Johnson and Sing Bing Kang. Registration and integration of textured 3d data. *Image and vision computing*, 17(2):135–147, 1999. (Cited on pages 52 and 137.)

[35] David W Eggert, Andrew W Fitzgibbon, and Robert B Fisher. Simultaneous registration of multiple range views for use in reverse engineering of cad models. *Computer Vision and Image Understanding*, 69(3):253–272, 1998. (Cited on pages 52 and 137.)

[36] Jérôme Declerck, Jacques Feldmar, Michael L Goris, and Fabienne Betting. Automatic registration and alignment on a template of cardiac stress and rest reoriented spect images. *Medical Imaging, IEEE Transactions on*, 16(6):727–737, 1997. (Cited on pages 52 and 137.)

[37] Mihailo Ristic and Djordje Brujic. Efficient registration of nurbs geometry. *Image and Vision Computing*, 15(12):925–935, 1997. (Cited on pages 52 and 137.)

[38] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. (Cited on page 63.)

[39] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Shape detection in point clouds. *Computer Graphics Technical Reports*, 2:2, 2006. (Cited on pages 63 and 143.)

[40] Rolf Adams and Leanne Bischof. Seeded region growing. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(6):641–647, 1994. (Cited on page 68.)

[41] Radu Bogdan Rusu. Semantic 3d object maps for everyday manipulation in human living environments. *KI-Künstliche Intelligenz*, 24(4):345–348, 2010. (Cited on pages 68, 81, 130, and 140.)

[42] David Levin. Mesh-independent surface interpolation. In *Geometric modeling for scientific visualization*, pages 37–49. Springer, 2004. (Cited on pages 73 and 75.)

[43] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin, and Claudio T Silva. Computing and rendering point set surfaces. *Visualization and Computer Graphics, IEEE Transactions on*, 9(1):3–15, 2003. (Cited on page 73.)

[44] M Gopi and Shankar Krishnan. A fast and efficient projection-based approach for surface reconstruction. In *Computer Graphics and Image Processing, 2002. Proceedings. XV Brazilian Symposium on*, pages 179–186. IEEE, 2002. (Cited on page 79.)

[45] Leif P Kobbelt, Stephan Bischoff, Mario Botsch, Kilja Kähler, Christian Rössl, Robert Schneider, and Jens Vorsatz. *Geometric modeling based on polygonal meshes*. Max-Planck-Institut für Informatik, 2000. (Cited on page 79.)

[46] Ernst Peter Mücke. Shapes and implementations in three-dimensional geometry. 1993. (Cited on page 79.)

[47] Pascal J Frey and Houman Borouchaki. Surface mesh quality evaluation. *International journal for numerical methods in engineering*, 45(1):101–118, 1999. (Cited on page 80.)

[48] Boris Delaunay. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2, 1934. (Cited on page 86.)

[49] Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991. (Cited on page 86.)

[50] Leonidas Guibas and Jorge Stolfi. Primitives for the manipulation of general subdivisions and the computation of voronoi. *ACM Transactions on Graphics (TOG)*, 4(2):74–123, 1985. (Cited on page 86.)

[51] David F Watson. Computing the n-dimensional delaunay tessellation with application to voronoi polytopes. *The computer journal*, 24(2):167–172, 1981. (Cited on page 86.)

[52] Adrian Bowyer. Computing dirichlet tessellations. *The Computer Journal*, 24(2):162–166, 1981. (Cited on page 86.)

[53] Steven Fortune. A sweepline algorithm for voronoi diagrams. *Algorithmica*, 2(1-4):153–174, 1987. (Cited on page 86.)

[54] Raimund Seidel. Constrained delaunay triangulations and voronoi diagrams with obstacles. *Rep*, 260:178–191, 1988. (Cited on page 86.)

[55] D Rogers. An introduction to nurbs: With historical perspective. morgan kaufmann. *San Francisco, CA*, 2001. (Cited on page 90.)

[56] Les A Piegl and Wayne Tiller. The nurbs book (monographs in visual communication). 1996. (Cited on page 91.)

[57] Point cloud libraries (pcl). Accessed: 2016-01-10. `http://pointclouds.org/`. (Cited on page 101.)

[58] JH Milgram, DB Peters, and DN Eckhouse. Modeling iacc sail forces by combining measurements with cfd. In *Proc. 11th Chesapeake Sailing Yacht Symposium, Annapolis, MD, January*, pages 29–30, 1993. (Cited on page 114.)

[59] Karsten Hochkirch and Hartmut Brandt. Full-scale hydrodynamic force measurement on the berlin sailing dynamometer. In *Proc. 14th Chesapeake Sailing Yacht Symposium, Annapolis, MD, Jan*, volume 30, pages 33–44, 1999. (Cited on page 114.)

[60] F Fossati, I Bayati, F Orlandini, S Muggiasca, A Vandone, G Mainetti, R Sala, C Bertorello, and E Begovic. A novel full scale laboratory for yacht engineering research. *Ocean Engineering*, 104:219–237, 2015. (Cited on page 116.)

[61] E Canciani, A Vandone, and R Sala. Characterization of a 2-d laser scanner for outdoor wide range measurement. In *Journal of Physics: Conference Series*, volume 658, page 012008. IOP Publishing, 2015. (Cited on page 124.)

[62] Soon-Yong Park and Murali Subbarao. An accurate and fast point-to-plane registration technique. *Pattern Recognition Letters*, 24(16):2967–2976, 2003. (Cited on page 138.)

[63] Yang Chen and Gérard Medioni. Object modeling by registration of multiple range images. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, pages 2724–2729. IEEE, 1991. (Cited on page 138.)

[64] Helmut Pottmann, Stefan Leopoldseder, and Michael Hofer. Registration without icp. *Computer Vision and Image Understanding*, 95(1):54–71, 2004. (Cited on page 138.)

[65] Helmut Pottmann and Johannes Wallner. *Computational line geometry*. Springer Science & Business Media, 2009. (Cited on page 139.)

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

CMM: Coordinate Measurement Machine
NURBS: Non-Uniform Rational B-Spline
3D: Three dimensional
2D: Two dimensional
RANSAC: RANdom SAmple Consensus
PCL: Point Cloud Library
TOF: Time Of Flight
FEA: Finite Element Analysis
CFD: Computational Fluid Dynamics
CAD: Computer Aided Design
ORC: Offshore Racing Congress