Università degli Studi di Firenze
Facoltà di Ingegneria

Dottorato di Ricerca in

Ingegneria Informatica, Multimedialità e Telecomunicazioni

Sistemi Multimediali e Interazione Uomo-Macchina

Area Tecnologica ING-INF/05

Accurate computer vision via robust estimation of parametric warping transformations

Tesi di Dottorato di

Dario Comanducci

Tutori:

Prof. Alberto Del Bimbo

Prof. Carlo Colombo

Coordinatore:

Prof. Giacomo Bucci

Ciclo XXI
Periodo 2006–08

To my family

Contents

1 Introduction

This thesis deals with the development of robust and accurate estimation techniques for geometric entities, such as plane-to-plane warping transformations and planar curves. The goal is to exploit such a 2D information to get accurate 3D results, in uncalibrated computer vision scenarios.

This "image-based" approach avoids introducing (and estimating) additional parameters (in particular camera pose and orientation), that adds further noise to the desired final output since they are not usually derived from direct image data but require several intermediate entities to be computed.

Bypassing visual 3D measurements is actually common practice in other research domains. For example, in *image-based visual servoing* in robotics, the visual feedback from a camera is exploited to control the movements of a robot w.r.t. a target. The desired trajectory of the robot is planned without any intermediate 3D measurement, by relying only on 2D-2D warping transformations on the image plane.

1.1 Image Warping Methods

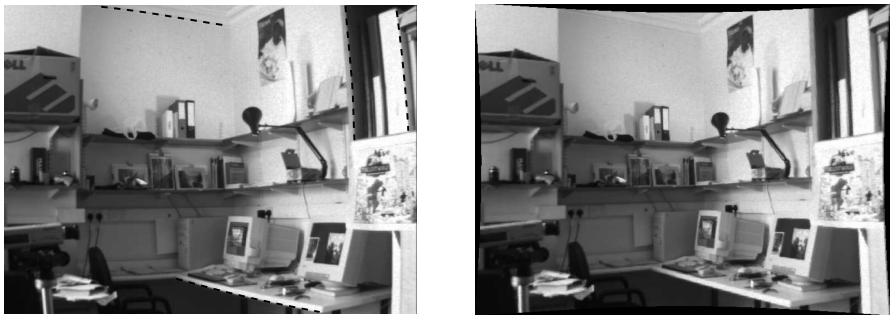
Image warping is a pair of two-dimensional functions $u(x, y)$ and $v(x, y)$ which map a position (x, y) in one image plane to position (u, v) in another plane. Such a mapping arises in many image analysis problems, whether in order

(a) (b)

Figure 1.1: Planar rectification (floor taken from [29]). If the image formation process follows the *central projection* rules, a rectifying *homography* (see §1.1.1 and Fig. 1.6) is used to remove the perspective of a imaged world plane (a) thus obtaining a front-to-parallel view (b). In central projection a ray is drawn from each 3D world point $\mathbf{X}$ through a fixed point in space, that is the camera *center of projection*. The intersection of this ray with the image plane provides the imaged position of $\mathbf{X}$.

to remove a particular viewing perspective (Fig. 1.1) or optical distortion introduced by a camera (Fig. 1.2) [29], or to compose panoramic mosaics [52] (Fig. 1.3).

Warping transformations are also powerful for 3D reconstruction of man-made objects and scenes from single images. In [18] warping transformations are also used to reconstruct the 3D scene of paintings (Fig. 1.4) drawn according to perspective laws, formally formulated for the first time by Leon Battista Alberti in 1435 [2]. In [15] the 3D shape and texture of a solid of revolution (SOR) are recovered from a single picture (Fig. 1.5). A warping transformation is applied to the imaged *apparent contour* of the SOR to obtain the perspective view of a *meridian* (a vertical section passing through the SOR symmetry axis) of the SOR (Fig. 1.5(a)); a rectifying homography recovers the metric shape of the meridian (Fig. 1.5(b)) and, finally, another warping transformation maps the imaged SOR texture onto the flattened SOR surface (Fig. 1.5(c)). The key point of both the approaches relies in the fact that man-made objects and scenes often obey to symmetrical arrangements and are usually piece-wise planar.



(a) (b)

Figure 1.2: Radial distortion removal (pictures from [29]). Low cost cameras such as webcams do not strictly follow the *pinhole* camera model commonly used in computer vision. The pinhole model is an abstraction of the image formation process based on central projection. Low cost lenses introduce several distortions in the image formation process, and radial distortion is the most noticeable. While central projection preserve straight lines, this is not true when radial distortion is tangible. In (a) the original photograph shows dashed lines which are straight in the world but curved in the image. The distorted image can be warped to a correct (compliant to a pinhole camera) image (b): The lines in the periphery of the picture are straight, while the boundary of the image is curved.

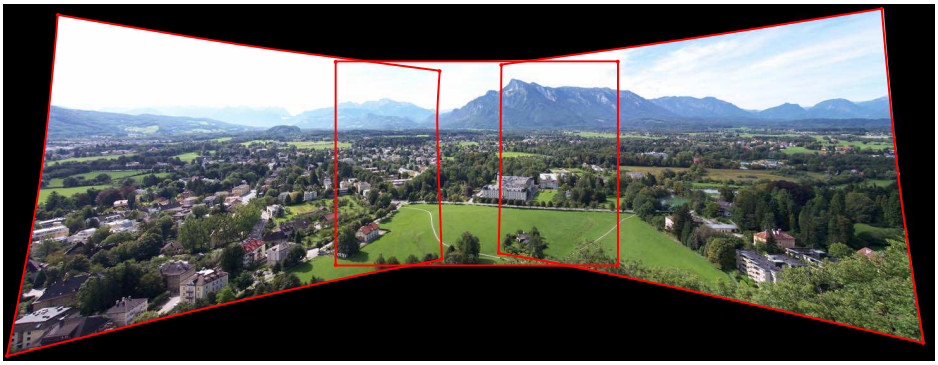
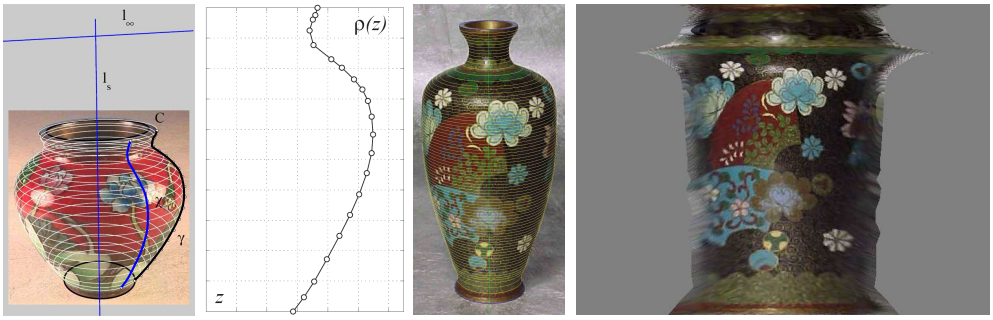


Figure 1.3: Image alignment and stitching. The left and right warped images come from rectangular photographs taken by rotating the camera around its center w.r.t. the camera orientation of the central (rectangular) image. The warping transformation (here, a radial distortion removal followed by a homography) registering a lateral image onto the central photograph can be found by using point correspondences: A wide-angle view of the landscape is then obtained.

(a) (b)

Figure 1.4: 3D scene and texture recovery from paintings (pictures from [18]). Both (a) and (b) shows the original drawing on the left, while the 3D textured model is on the right. (a): Masaccio's fresco *La Trinità* (Firenze, Santa Maria Novella; 1426), probably the first perspective image in history. (b): *St Jerome in his Study* by Henry V Steinwick (Joseph R. Ritman Collection; 1630).



(a) (b) (c)

Figure 1.5: Warping transformation for 3D shape and texture recovery of a solid of revolution (SOR). Images taken from [15]. (a): The apparent contour is warped in a imaged SOR meridian. (b): A rectifying homography is applied to the imaged SOR meridian, thus recovering the SOR metric shape. (c): The imaged texture of the SOR is mapped onto the flattened SOR surface.

The choice of warping transformation is usually a compromise between a smooth distortion and a good match. Smoothness can be assured by assuming a parametric form for the warping function or penalizing roughness, e.g. by using thin-plate splines. Matching is obtained by point alignment, local similarity or edge correspondences.

1.1.1 Parametric Warping

In Fig. 1.6, a hierarchy of the most widely used parametric warping transformation is shown. In many applications it is important to use a transformation which is no more general than it is needed. The simplest warping transformation is a pure translation; it is a very special case of a similarity transformation. Similarities are the most general transformations preserving shape, but not dimensions, of a planar object; dimensions do not vary only with translation and translation plus rotation. Beyond similarity, affinities are the simplest deforming transformations: They preserve parallelism, but angles among not parallel lines may change. An affinity can be see as a specialization of two different kind of warping methods: It is a limit case for homographies and is the most general linear (i.e. a particular polynomial) transformation. A homography maps lines into lines (being a fundamental brick of projective computer vision [29]), while this is not true for a polynomial transformation. A discussion of polynomial transformations can be found in [48].

1.1.2 Non-parametric Warping

Parametric transformations do not perform well in presence of local distortions. Piecewise affine transformations offer an alternative to polynomials in generalizing affine transformations. Given a set of matched points in the image and their Delaunay triangulation, a different affine transformation can be used in each triangle. Continuity of the transformation along the edges of triangles is thereby assured. This is a first-order or linear spline. Other

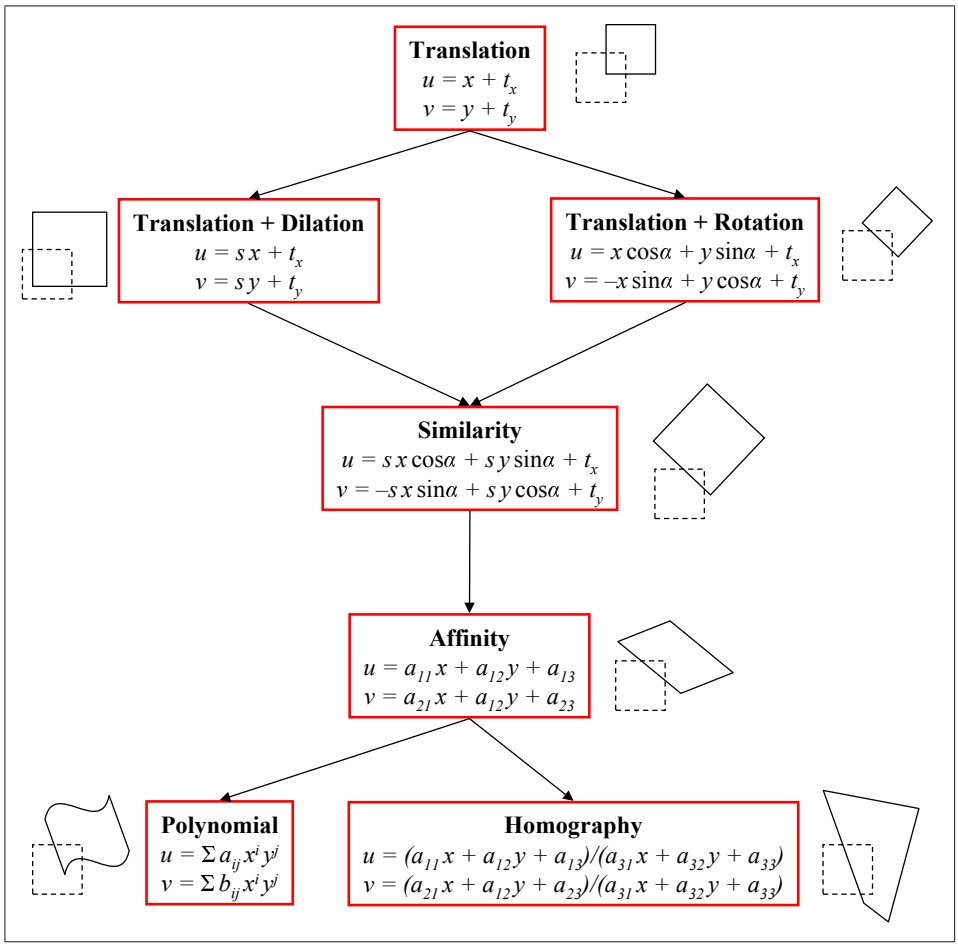


Figure 1.6: A hierarchy of several warping transformations; arrows denote generalization flow, while the pictures next to each block illustrate the effect of the corresponding transformation applied to a square.

interpolant functions are cubic spline and the *sinc* function.

For non-parametric warping methods, if smoothness considerations are not taken into account, the transformed image can be very rough. The introduction of smoothness leads to a variety of non parametric transformation, such as elastic deformations, radial basis functions plus affinity and Bayesian priors. In the first case, warping is equated with the distortion of an elastic sheet. Another physical analogue can be conveniently used to explain the radial basis function approach: $u(x, y)$ and $v(x, y)$ are regarded as a pair of two-dimensional surfaces. For example, u-surface (and similarly v) can be represented as

$$u = \sum_{i=1}^{m} c_i f\left(\sqrt{(x - x_i)^2 + (y - y_i)^2}\right) + a_{11}x + a_{12}y + a_{13}$$

where $\{(x_i, y_i)\}_{i=1}^{m}$ is a set of landmarks and $f(\cdot)$ is a function such as a multiquadratic

$$f(t) = (t^2 + t_0^2)^{\tau}, \quad 0 < \tau < 1 \ ;$$

a shifted log

$$f(t) = \log\left(\sqrt{t^2 + t_0^2}\right), \quad t_0^2 > 1 \ ;$$

a Gaussian density

$$f(t) = \exp\left(-\frac{t^2}{2\sigma^2}\right) \ ;$$

or a thin-plate spline

$$f(t) = t^2 \log t^2 \ .$$

Finally, for the Bayesian approach, the first image (I) is regarded as a template to be aligned with a second image (I') using a set of transformation parameters $W = \{w_j\}_{j=1}^{n}$. W is estimated using the posterior density, expressed in terms of likelihood and prior density

$$\Pr\{W|I', I\} \propto \Pr\{I'|W, I\}\Pr\{W\} \ .$$

A common solution to Bayesian estimation is provided by the Metropolis-Hasting algorithm, a Monte Carlo technique which has the required posterior as its limiting case [27].

1.2 Parameter Estimation Methods

Almost all problems in computer vision are related to the estimation of parameters from noisy data. Many computer vision fields involve sets of equations that are very sensitive to noise: For this reason, great care must be adopted to make these equations effective in practical applications or, in other words, to let them work.

Generally speaking, let $\mathbf{p} \in \mathbb{R}^m$ be the vector of the parameters to be estimated; together with a datum $\mathbf{z}$, it allows to define a function $f(\mathbf{p}, \mathbf{z})$ that holds $f(\mathbf{p}, \mathbf{z}) = \mathbf{0}$, when $\mathbf{z}$ satisfies the model. Since in practice data are always noisy, a solution for the system of equations $f(\mathbf{p}, \mathbf{z}_i) = \mathbf{0}$, $i = 1 \dots n$ does not exist. Hence, the problem is addressed by defining a *cost function* (or *objective function*) $F(\mathbf{p}, \mathbf{z}_1, \dots, \mathbf{z}_n)$ to be optimized. The cost function is conventionally arranged so that small values represent close agreement. The parameters of the model are then adjusted to achieve a minimum in the cost function, yielding best-fit parameters. The adjustment process is thus a problem of minimization in many dimensions.

Parameter estimation problems are customarily divided into linear and nonlinear, depending on whether the parameter of interest appears linearly or nonlinearly into the considered model formulation. Furthermore, for computer vision applications, there are two main categories of cost function: Those based on minimizing an *algebraic* error, and those based on minimizing a *geometrical* image distance.

Because of different optimization criteria and because of several possible parameterizations, a given problem can be solved in more than one way. Particular care should be given to the choice of the cost function, that should take into account the specific model adopted. There will usually be one cost function which is optimal in the sense that the parameter vector $\mathbf{p}$ minimizing it gives the best possible estimate of the model under certain (statistical) assumptions. In order to obtain a best (optimal) estimate of $\mathbf{p}$ it is also necessary to have a model for the measurement error (the "noise").

The *least squares* (LS) cost function is the most easily adopted:

$$F(\mathbf{p}, \mathbf{z}_1, \ldots, \mathbf{z}_n) = \sum_i \|\mathbf{r}_i\|^2 \ ;$$

where the non zero value $\mathbf{r}_i = f(\mathbf{p}, \mathbf{z}_i)$ is usually referred as *residual*. The residual of the fit depends on what noise was added to the data, so the probability distribution of the noise induces a probability distribution on the residual. The least square strategy has a statistical motivation in some specific scenarios; as an example, the approach provides the Maximum Likelihood (ML) estimate in linear regression analysis, under the hypothesis of an independent Gaussian measurement error (whenever you encounter a least squares strategy, there is somewhere a built in assumption of independence and Gaussian distribution). The ML solution is the parameter vector $\mathbf{p}$ maximizing the joint probability that the dataset $\{\mathbf{z}_i\}_{i=1}^n$ can be observed, provided that $\mathbf{p}$'s elements are the actual parameters for the model to be fitted.

The most dramatic deviation from Gaussian measurement error is the presence of *outliers* in the dataset $\{\mathbf{z}_i\}_{i=1}^n$. Outliers are grossly erroneous measurements, and usually they are not related to the model to be estimated. The Gaussian error hypothesis is more tenable, once outliers have been removed. The LS estimator cannot cope with outliers: one bad datum is enough to produce highly perturbed or completely wrong solutions. During the last three decades, many robust techniques have been proposed to neglect the outlier influence on parameter estimatimation. Some of them replace the squared function with other functions, "down-weighting" the heaviest residuals (the so called *M-estimators*) [62] or minimizing the median of the squared residuals (*Least MEDian of Squares*, LMEDS) instead of the overall sum [50].

Among all the robust estimation techniques, in this thesis a special role has been played by the *RANdom SAmple Consensus* (RANSAC) algorithm [24]. It exploits a threshold to discriminate the outliers in the dataset, aiming to maximize the number of data that agree (w.r.t. the threshold) with a particular solution $\mathbf{p}$ of the model.

1.3 Uncertainty Estimation

Warping transformations are used in this thesis to produce measurements in world planes, and thus a prediction of the uncertainty of these measurements is necessary too. Uncertainty analysis requires to model (i) the error selecting image points; (ii) the error in the warping transformation itself; (iii) how the errors in the image plane are propagated in the world plane. In the literature [11] [19] there are two main approaches addressing the matter: *Monte Carlo simulation* and *first order analisys*.

The Monte Carlo strategy is a numerical method that can be used to approximate covariance matrices, the usual statistical tool describing uncertainty. Given a function $\mathbf{y} = g(\mathbf{x})$ and some exact data $\mathbf{x}$, a great number N of corrupted data $\mathbf{z}_i = \mathbf{x} + \mathbf{e}_i$ is created by adding random noise $\mathbf{e}_i$ to $\mathbf{x}$. This large population is the used to infer the distribution if $\mathbf{y}$ from the samples $\mathbf{y}_i = g(\mathbf{z}_i)$. Thus mean and covariance matrix for $\mathbf{y}$ are respectively obtained as:

$$\begin{aligned} \overline{\mathbf{y}} &= \frac{1}{N}\sum_i \mathbf{y}_i \\ \Lambda_{\mathbf{y}} &= \frac{1}{N}\sum_i (\mathbf{y}_i - \overline{\mathbf{y}})(\mathbf{y}_i - \overline{\mathbf{y}})^\top \quad . \end{aligned}$$

The function $g(\cdot)$ need not be known explicity: For example, $\mathbf{y}_i$ can be obtained by numerical optimization. Despite the simplicity of the Monte Carlo approach, the drawbacks are twofold: (i) it is slow, and (ii) it provides only a solution to a specific problem (if data change, the entire work must be repeated). This method is then usually exploited when only a few results are required or speed of implementation is preferred over speed of execution. The Monte Carlo method is a useful tool to validate the first order approach, discussed hereafter.

The first order approach is an analytical method to derive a closed form expression for the covariance matrix, by truncating to the first order the Taylor series expansion of $g(\mathbf{x})$ around the expected value $\overline{\mathbf{x}}$ of $\mathbf{x}$:

$$g(\mathbf{x} + \Delta\mathbf{x}) = g(\overline{\mathbf{x}}) + \nabla g(\overline{\mathbf{x}})\Delta\mathbf{x} + o(\|\Delta\mathbf{x}\|^2) \quad , \tag{1.1}$$

thus obtaining

$$\begin{aligned}
\Lambda_{\mathbf{y}} &= E\left[(g(\mathbf{x}+\Delta\mathbf{x})-\overline{\mathbf{y}})(g(\mathbf{x}+\Delta\mathbf{x})-\overline{\mathbf{y}})^{\top}\right] \\
&\approx E\left[(g(\mathbf{x}+\Delta\mathbf{x})-g(\overline{\mathbf{x}}))(g(\mathbf{x}+\Delta\mathbf{x})-g(\overline{\mathbf{x}}))^{\top}\right] && \text{by setting } \overline{\mathbf{y}} \approx g(\overline{\mathbf{x}}) \\
&\approx E\left[\nabla g(\overline{\mathbf{x}})\Delta\mathbf{x}\Delta\mathbf{x}^{\top}\nabla g(\overline{\mathbf{x}})^{\top}\right] && \text{by Eq. 1.1} \\
&= \nabla g(\overline{\mathbf{x}})\Lambda_{\mathbf{x}}\nabla g(\overline{\mathbf{x}})^{\top} \ .
\end{aligned}$$

In [19] this general approach was applied to compute the uncertainty in the homography parameters and the uncertainty in the world plane propagated by this (noisy) homography, given the image point-wise covariance matrix.

1.4 Thesis Organization and Contributions

In chapter 2 robust estimation techniques are exploited to perform 3D model acquisition of desktop objects. The approach requires a camera viewing the planar sections induced by a laser stripe on the surface of the scanned object; a rectifying homography is applied to the imaged laser sections. Imaged laser stripes are also warped to obtain a synthetic view of the object without computing and re-projecting the 3D point cloud model. The synthetic view is then used to align the laser light sequence with a companion natural light sequence so as to assign a texture color to each laser point on the object surface, thus producing the final textured model. The advantages of the "imaged-base" approach are discussed and a comparison between the proposed strategy and a corresponding "full camera" approach is included too. Furthermore, a pyramidal version of RANSAC algorithm is discussed and exploited as a means to to recover the parameters of a special class of homography (*harmonic homology*).

Chapter 3 deals with the development of a robust tracking and remapping algorithm of iris appearance with passive computer vision, to obtain an advanced human-computer interaction device. The goal is to drag the mouse icon on the computer screen by the visual tracking of the iris in a close-up video stream of the user's eye. At startup, the user is asked to fix some points

(*landmarks*) on the screen, ao as to compute a image-to-screen remapping function. The approach exploits only natural illumination of the eye, without any additional device. Frame by frame iris localization is performed by a modified version of RANSAC. In this case a way to introduce constraints in the parameter space is proposed, thus eliminating spurious solutions due to the lack of data (top and bottom parts of the iris are missing because of eyelid occlusions).

Error analysis w.r.t. system layout displacement is carried out for the devices described in both the following chapters, even if in different ways. In particular, for 3D model acquisition, the accuracy of 3D measurements is investigated for several laser plane displacements w.r.t. the camera. A trade-off between accuracy and possibility of occlusions is then chosen. In chapter 3, first order analysis is applied to infer the remapping error on the screen plane of the imaged iris center, given three different layouts for the landmarks (required to compute the warping transformation image-screen). A comparison among the three results gives the best layout to be used.

Throughout this work a very important role is played by the estimation of ellipses, a recurrent task in computer vision problems (e.g. in image segmentation). When projected to an image, a circle becomes in general an ellipse. In chapter 2 ellipses arise from the circular motion of the rotating object and their knowledge is exploited to obtain the final result; in chapter 3 they are clearly due to the iris shape. In both the topics discussed in the two chapters, great care was addressed to get accurate estimation of the involved ellipses.

2

3D model acquisition

In this chapter it is shown how to obtain a three-dimensional, textured model of an object, by relying exclusively on image warping 2D-2D transformations. To achieve this goal, while the object undergoes a circular motion on a turntable, a dual (laser and natural light) illumination of the scene is exploited.

Traditional approaches are based on triangulation. In the warping approach, instead, shape reconstruction is based on the planar rectification and collation of laser profiles. Texture sampling uses laser profile warping and compositing rather than 3D-2D model shape reprojection, so as to synthesize a virtual view of the object at hand, that is matched against the natural light sequence to obtain color data.

Among the camera parameters, only the internal ones are required to perform the rectification of laser profiles; instead, external camera parameters are also needed in the triangulation-based approach. Internal camera calibration is crucial for obtaining a metric shape model of the object and correctly sample texture data from images. All camera parameters can be computed by exploiting the single axis motion of the turntable, and by analyzing the appearance of the volume swept by the object during a whole turntable round. Warping transformations are again involved in this analysis.

TRADITIONAL 3D MODEL ACQUISITION systems employ an *active* framework, where structured radiation (radio, incoherent or coherent light, ultrasound, etc.) is emitted, and its interaction with the objects in the scene is observed so as to obtain object shape. Active systems typically operate in

heavily structured conditions, and achieve very high accuracies (a few tenths of mm) through sophisticated hardware and relatively simple software control. Active 3D model acquisition technology has evolved considerably in the last few years (see [3], [10] for an overview). Several commercial 3D scanning devices have also appeared in the market. Among the most popular active 3D devices, *time-of-flight scanners* measure the round trip time before the reflected radiation (laser light being the most common one) is received by the device. These scanners are usually employed for far away and large objects, such as buildings. Typical laser time-of-flight scanners acquire thousand of points every second, and use motorized mirrors in order to rapidly change the direction of radiation. Another common approach to active 3D scanning suitable for medium/small size objects uses a *structured light* pattern together with a standard camera. Several devices can be employed to generate the light pattern: Laser emitter, custom white light projectors employing pattern filters, slide projectors, and digital video projectors (the latter permit a virtually infinite number of different patterns) [9]. Typical patterns consist of several light stripes, arranged in a coded or in a regular way [49]. When the pattern is projected onto an object, it is modified by the object's shape. Inferring shape from pattern deformation then follows from straightforward triangulation methods, given an accurate knowledge of parameters related to camera optics and camera-projector relative pose—the so called *active triangulation* framework [3]. Such parameters have to be obtained through a careful calibration of both the camera and the projector, typically exploiting cumbersome photogrammetric procedures and ad hoc calibration artifacts.

A more recent research trend in 3D model acquisition derives from the visual analysis of natural light video sequences. This *passive* framework is characterized by a shift of emphasis from performance to flexibility, and from hardware to software. Typical passive approaches work in unstructured or in loosely structured environments, and achieve accuracies of about 1 mm with off-the-shelf hardware and sophisticated algorithms. Most passive methods employ reasonably accurate *self-calibration* approaches based on prior knowl-

edge about scene structure [47], [37] or camera motion [26], [34] as a valid alternative to traditional calibration approaches based on photogrammetry. In *multi-view stereo*, two or more images taken from different viewpoints are used to obtain the depth of scene points by triangulation [29]. Images can either be acquired simultaneously as in a stereo setup, or in the form of a video sequence by a single camera moving with respect to the 3D scene. The computational problems to be solved in the two cases are geometrically equivalent, but with important implementation differences related to the width of the baseline between view pairs. When the baseline is relatively narrow, as in the case of continuous video sequences, either dense or sparse image correspondences between view pairs can be established by simple patch correlation [32]; yet, frame-by-frame recovery of motion and structure with a narrow baseline turns out to be quite sensitive to noise. On the other hand, a large baseline makes the 3D reconstruction task more robust, but requires more complex techniques for feature matching [40], [44], due to the fact that corresponding image patches are both geometrically and photometrically distorted. An original passive approach based on triangulation and referred to as *shape from shadows* was proposed in [7]. In this approach, object shape is obtained from the visual analysis of the shadows cast upon the object using a straight wand moved by the user in the presence of a fixed light source, such as a desktop lamp. To work properly, the approach requires both intrinsic and extrinsic calibration parameters, and the 3D location of two mutually orthogonal reference planes, together with the image of their line of intersection. In the *shape from shading* approach, the interaction between 3D shape and light is exploited, so that the surface normal and the light direction are recovered from a gradual variation of shading in the image [61]. The shape from shading problem is extremely challenging, as it deals with an underconstrained system of equations, that require additional constraints to be solved uniquely. A popular approach to practical shape from shading is *photometric stereo*, which uses two or more shaded images obtained by varying the light direction while keeping fixed both the camera and the scene, so that

every pixel always corresponds to the same point of the surface. The main advantage of photometric stereo over multi-view approaches is that it can also be applied to textureless objects, and does not require the extraction of point correspondences. However, standard photometric stereo does not allow the recovery of the full 3D geometry of a complex many-sided object, such as a sculpture. Indeed, existing photometric stereo techniques have so far only been able to extract depth-maps of a scene (e.g. [53]) with the noticeable recent exceptions of [60], [39], that recover 2.5D reconstructions from multiple viewpoints. *Volumetric methods* are based on silhouette extraction and space occupancy analysis. In particular, *space carving* [35] is a method that starts from a volume containing the scene and greedily carves out non photo-consistent voxels from that volume, until all the remaining visible voxels are consistent. The method uses a discrete representation of the surface, but does not enforce any smoothness constraint, which often results in noisy reconstructions. Differently, in the *shape from silhouettes* method, object silhouettes relative to different viewpoints are back-projected onto the 3D space to obtain object volume [56], [51]; the initial estimate of the 3D model is the maximal volume that projects inside the silhouettes, or "visual hull" [36]. Methods based on silhouettes—see also [43]—are robust and fast, and can deal with textureless objects. However, they are typically limited to simply shaped objects, e.g., surfaces that are smooth everywhere, and are not locally planar. Recent approaches combine photometric stereo and silhouettes so as to overcome the drawbacks of both methods [31].

From a system design point of view, a cost/performance trade-off exists for 3D acquisition approaches. On the one hand, structured light scanning approaches are computationally simple and very accurate, yet they typically require expensive components and good manufacturing. On the other hand, unstructured light approaches are computationally more challenging, but also more flexible. Another point of difference between active and passive methods is in the role played by object texture. Vision-based shape reconstruction approaches based on triangulation or volumetric methods typically rely on

texture information, so that it is difficult (if not impossible) for them to perform model acquisition of textureless objects. On the other hand, active light approaches work usually better in the absence of texture, due to the fact that surface colors can alter the detectability of the superimposed pattern.

Texture sampling has been typically regarded as a second step after 3D shape reconstruction. The ultimate goal is to map a set of color images onto a 3D shape object model in an efficient and accurate way. Recent trends in texture sampling are discussed in [22]. These include: (1) Surface parametrization, i.e., how to parametrize optimally the surface onto the texture space; (2) Color data integration from multiple views into a coherent texture map; (3) Extraction of the reflectance properties of the object—i.e., its true color—from the raw color appearance. A recent alternative to surface parametrization is the so called *color-per-vertex* approach, where a color value is assigned to each vertex of the mesh modeling object shape [6].

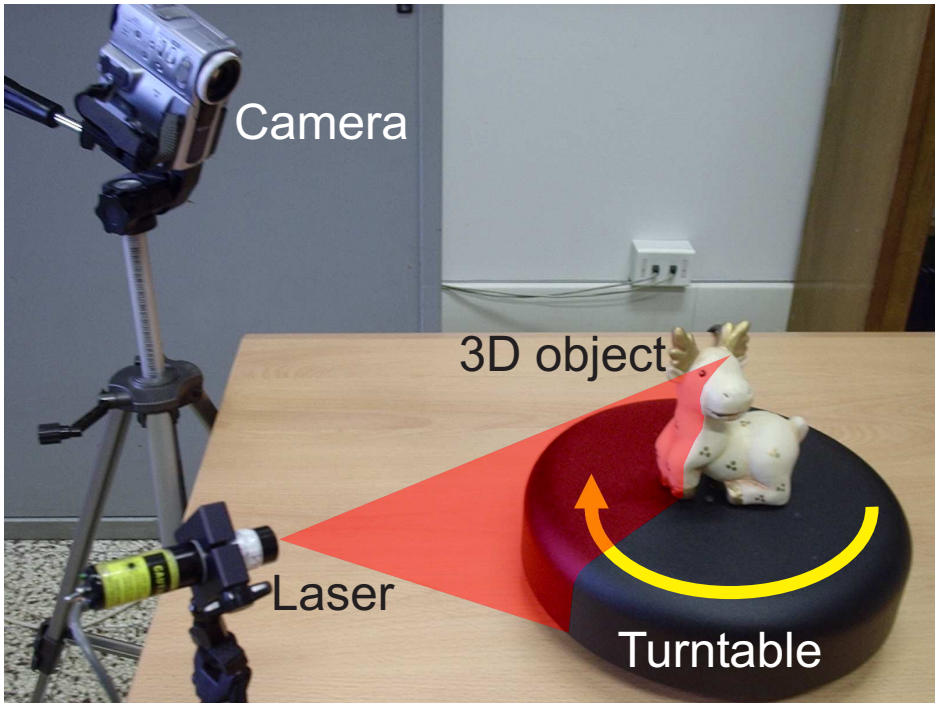
2.1 Overview

A hybrid 3D model acquisition approach based on laser illumination and turntable motion is discussed (Fig. 2.1).

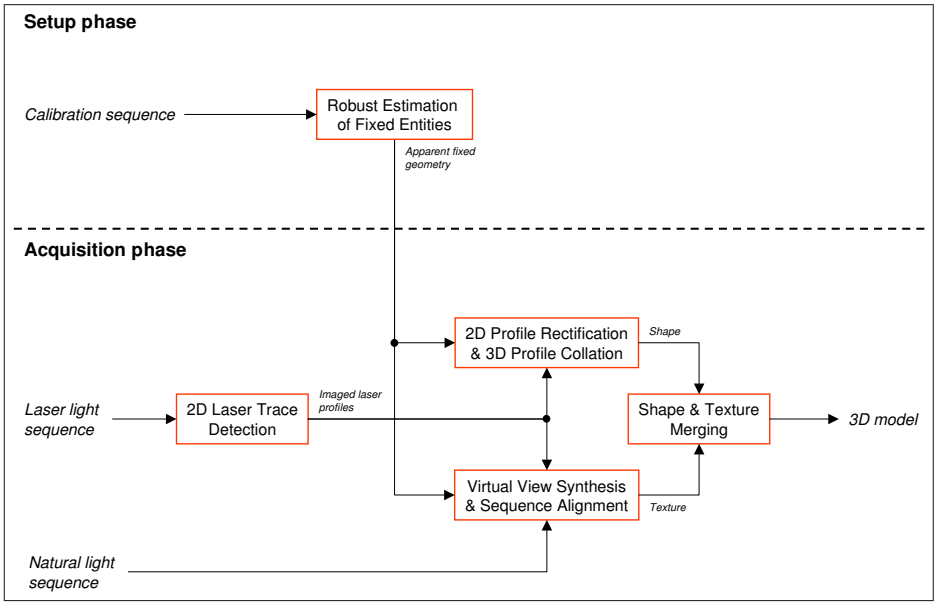
A single fixed camera is used to acquire complete (360 degrees) image sequences of the objects rotating on a turntable. At acquisition time, two distinct sequences are taken: A *laser light sequence* obtained by illuminating the scene with a laser plane, and a *natural light sequence*, during which the laser is switched off.

The imaged laser profiles are used to perform 3D object shape reconstruction by profile rectification and collation. Laser profile detection is improved by using a subpixel peak detector for the laser intensity [25] along each image row.

The laser light sequence is also combined frame-by-frame with the natural light sequence, in order to perform texture sampling. The natural light sequence is temporally aligned with the laser profile sequence through a syn-



(a)



(b)

Figure 2.1: (a) 3D scanning system layout. (b) The main blocks of the 3D model acquisition architecture.

thetic view of the object—obtained *without 3D shape information*, by compositing the laser profiles in the image.

The shape reconstruction and texture sampling algorithms both require an a priori knowledge of *apparent fixed geometry*—i.e., how the fixed elements of the scene (laser plane, turntable) are seen from the camera viewpoint. Such knowledge is actually independent of the objects to be acquired, and can therefore be computed offline, at setup time. Offline computations require a calibration sequence, obtained as the concatenation of a laser sequence with no object on the turntable, and a natural light sequence with an object rotating on the turntable.

2.2 Offline Phase: Apparent Fixed Geometry

The most relevant apparent fixed entities are (see Fig. 2.2): (1) the image line $\mathbf{l}_\lambda$ at which the laser and turntable planes intersect, (2) the imaged axis of rotation $\mathbf{l}_\perp$, (3) the imaged turntable center $\mathbf{x}_t$, (4) the vanishing point $\mathbf{v}_\infty$ of the normal direction to the plane through $\mathbf{l}_\perp$ and the camera center, and (5) the imaged circular points $\mathbf{i}$ and $\mathbf{j}$ of the turntable plane—these two points lie on the vanishing line of the turntable plane $\mathbf{l}_\infty$.

Fixed entity (1) is the only one related to both laser and turntable planes. It is estimated in a robust way by running the RANSAC algorithm on putative laser points extracted by maximum intensity search along image rows.

Fixed entities (2) through (5) depend only on turntable position and motion—they are actually independent of the appearance changes and 3D shape of the object rotating on the turntable. As shown in [34], such entities can be estimated from a turntable sequence, by tracking points of any textured object as it undergoes a complete rotation. However, in the present work a different approach is followed, inspired by the geometrical analogy between turntable sequences and solids of revolution [15]: the image of a *virtual solid of revolution* (VSOR) is constructed from image data as in [58], and its properties are analyzed with an automatic segmentation algorithm

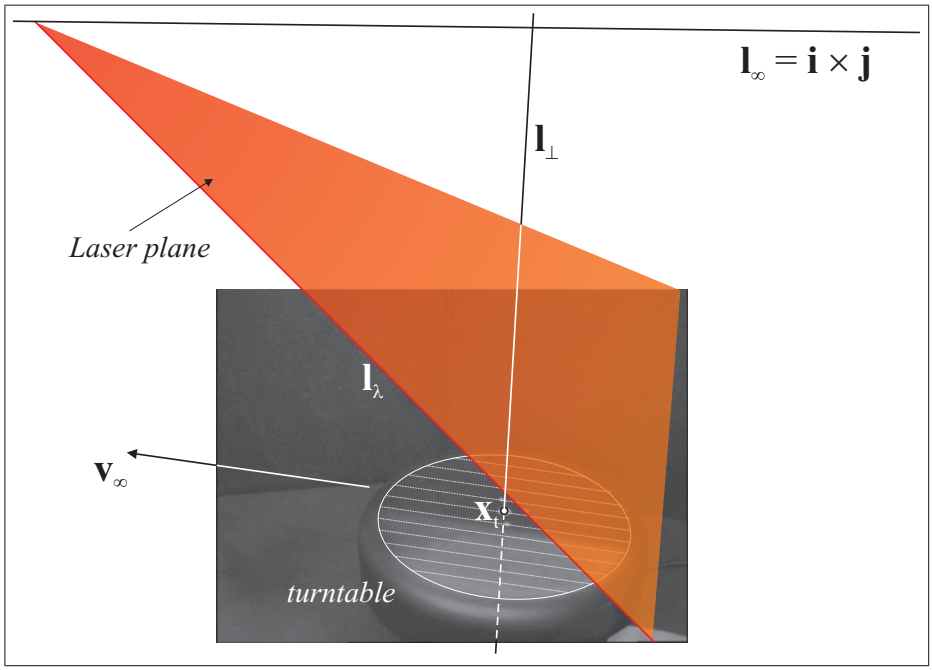
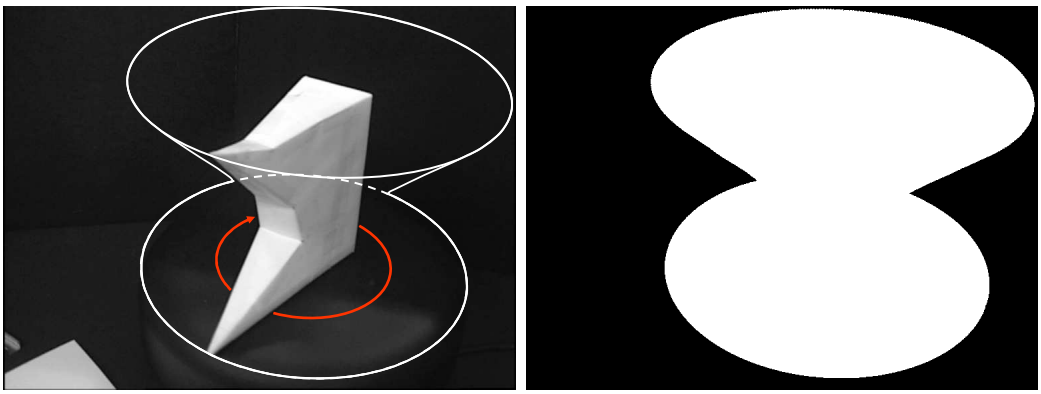


Figure 2.2: Apparent fixed geometry of interest.

specializing the work on generic SOR segmentation from a single photo described in [13] [17]. The approach does not require feature tracking and works also with textureless objects, and relies on the construction of a single image preserving the fixed entity information of the turntable sequence.

As shown in Fig. 2.3(a), the volume swept in 3D space by the rotating object forms the VSOR. The projection of the VSOR onto the image plane gives rise to two different kinds of image curves (also shown superimposed in Fig. 2.3(a)), namely the *apparent contour* and the *imaged cross sections*. The apparent contour is the image of the points at which the VSOR surface is smooth and the projection rays are tangent to the surface. Imaged cross sections correspond to coaxial circles in 3D—the common axis being the turntable axis—and arise from surface normal discontinuities. The VSOR image of Fig. 2.3(b) summarizes the whole rotating object sequence. This image includes the silhouette constructed by superposition of the binarized difference between the current and the first frame of the sequence.

Now, the four fixed entities of turntable motion can be put in a one-



(a) (b)

Figure 2.3: (a): An object rotating on the turntable (second part of the calibration sequence), with the VSOR silhouette boundaries superimposed. (b): The VSOR silhouette.

to-one correspondence with elements of the VSOR image—see Fig. 2.4. In particular, the axis of rotation coincides with the VSOR symmetry axis, and the turntable center is also the center of the VSOR bottom cross-section. Moreover, all turntable fixed entities can be extracted by segmenting and analyzing the image properties of the VSOR silhouette, as described hereafter.

2.2.1 VSOR Image Segmentation and Analysis

The problem addressed in the following paragraphs is how to estimate automatically the turntable fixed entities from the imaged VSOR silhouette.

Harmonic Homology Parameters

Being the perspective projection of an axially symmetric 3D object, the imaged VSOR silhouette is also symmetric, even if only in a projective sense. The 4-dof warping transformation characterizes such projective symmetry: It is an *harmonic homology*, transforming the imaged VSOR silhouette onto

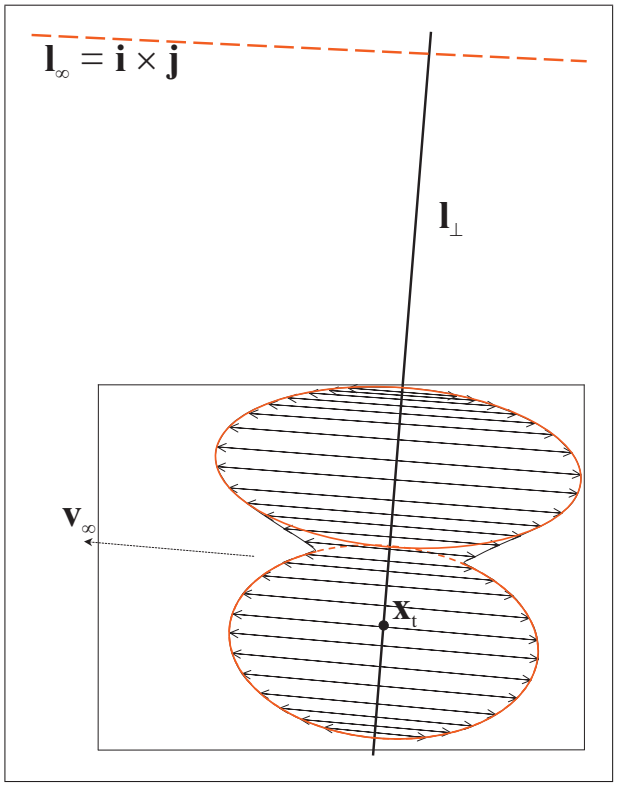


Figure 2.4: Turntable fixed geometry and its relationship with the VSOR image.

itself, and it is parameterized by the fixed entities $\mathbf{l}_\perp$ (2 dof) and $\mathbf{v}_\infty$ (2 dof):

$$\mathtt{F}(\mathbf{l}_\perp, \mathbf{v}_\infty) = \mathtt{I}_{3\times 3} - 2\frac{\mathbf{v}_\infty \mathbf{l}_\perp^\top}{\mathbf{v}_\infty^\top \mathbf{l}_\perp} \ , \tag{2.1}$$

the equal sign denoting equality up to scale with homogeneous coordinates. In Eq. 2.12, the imaged turntable axis $\mathbf{l}_\perp$ is a line of fixed points, while the "vertex" $\mathbf{v}_\infty$ is the vanishing point of the normal to the plane through the turntable axis and the camera center.

The harmonic homology is estimated in a robust way as the result of a nonlinear optimization problem using a method of solution similar to the ICP algorithm [4]. The idea is to recursively estimate the unknown $\mathbf{l}_\perp$ and $\mathbf{v}_\infty$ as the parameters of the optimal homology transformation that aligns one to the other the two projectively symmetric halves of the imaged VSOR silhouette contour. In order to avoid to get stuck in local minima, the optimization algorithm is run at multiple scales and applied to subsequent levels of a Gaussian image pyramid. At each new level ℓ, an estimate $(\mathbf{l}_\perp^\ell, \mathbf{v}_\infty^\ell)$ is obtained as a refinement to the previous one by directly minimizing the registration error

$$\begin{aligned} \mathcal{E}_r(\mathbf{l}_\perp^\ell, \mathbf{v}_\infty^\ell) &= \sum_i \|\mathbf{x}_i' - \mathtt{F}(\mathbf{l}_\perp^\ell, \mathbf{v}_\infty^\ell)\mathbf{x}_i\|^2 + \sum_i \|\mathbf{x}_i - \mathtt{F}^{-1}(\mathbf{l}_\perp^\ell, \mathbf{v}_\infty^\ell)\mathbf{x}_i'\|^2 \\ &= 2\sum_i \|\mathbf{x}_i' - \mathtt{F}(\mathbf{l}_\perp^\ell, \mathbf{v}_\infty^\ell)\mathbf{x}_i\|^2 \end{aligned} \tag{2.2}$$

using nonlinear optimization. In Eq. 2.2, $\mathbf{x}_i$ and $\mathbf{x}_i'$ are contour points corresponding under $\mathtt{F}$. To generate the initial pair of contour point sets, a first guess homology solution is computed by running the RANSAC algorithm at the coarsest level ($\ell = 0$) of the pyramid. At such a low resolution level, projective symmetry actually reduces to a simpler Euclidean axial symmetry, that can be described by the 2-dof axis $\mathbf{l}_\perp^0$ only—the vanishing point $\mathbf{v}_\infty^0$ being the point at infinity in the image direction orthogonal to the axis.

Imaged Circular Points and Turntable Center

Both the apparent contour and the imaged cross sections of the VSOR are transformed onto themselves by the harmonic homology $\mathtt{F}$ of Eq. 2.12. By

definition, imaged cross sections are ellipses, and all intersect at the turntable imaged circular points. As shown in [15], an imaged cross section pair is enough to estimate $\mathbf{i}$ and $\mathbf{j}$ uniquely, provided that additional information about the image position of the two ellipses with respect to the turntable vanishing line $\mathbf{l}_\infty = \mathbf{i} \times \mathbf{j}$ is given. In this work, the relative placement of the two ellipses is below the turntable vanishing line, reflecting the fact that the camera gets a top view of the scene (see again Fig. 2.4).

In order to estimate the turntable imaged circular points, the images of the top and bottom cross sections of the VSOR are extracted automatically from the VSOR image. The method exploits the well known property that, at any point $\mathbf{x}$ of an imaged solid of revolution contour, the local tangent line $\mathbf{l}$ to the contour is also tangent to the imaged cross section through $\mathbf{x}$ [1]. This allows us to describe the set of all possible VSOR imaged cross sections through the point pair $\mathbf{x}$ and $\mathbf{x}' = \mathtt{F}\mathbf{x}$ with local tangent lines $\mathbf{l}$ and $\mathbf{l}' = \mathtt{F}^{-\top}\mathbf{l}$ as

$$\mathtt{C}_\mathbf{x}(\varphi) = \Phi_1(\mathbf{x}, \mathbf{x}') + \varphi\, \Phi_2(\mathbf{l}, \mathbf{l}') \ . \tag{2.3}$$

Eq. 2.3 is a valid and complete parametrization of the set, since each member $\mathtt{C}_\mathbf{x}(\varphi)$ of the conic pencil satisfies both the pole-polar constraints on the set $\mathbf{l} = \mathtt{C}_\mathbf{x}(\varphi)\mathbf{x}$ and $\mathbf{l}' = \mathtt{C}_\mathbf{x}(\varphi)\mathbf{x}'$, each of which fixes two out of the five dofs of a general conic of the plane. The 3×3 homogeneous symmetric matrix in Eq. 2.3 represents a 1-dof pencil of conics, where φ is a scalar, $\Phi_1(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \times \mathbf{x}')(\mathbf{x} \times \mathbf{x}')^\top$ is the (rank 1) degenerate conic composed by the line $\mathbf{x}' \times \mathbf{x}$ through the two points $\mathbf{x}$ and $\mathbf{x}'$, and $\Phi_2(\mathbf{l}, \mathbf{l}') = \mathbf{l}\mathbf{l}'^\top + \mathbf{l}'\mathbf{l}^\top$ is the (rank 2) degenerate conic composed by the line pair $\mathbf{l}$ and $\mathbf{l}'$ tangent to the contour respectively at $\mathbf{x}$ and $\mathbf{x}'$. The member of the pencil passing through any assigned point $\overline{\mathbf{x}}$ is $\mathtt{C}_\mathbf{x}(\overline{\varphi})$, where

$$\overline{\varphi} = -\frac{\overline{\mathbf{x}}^\top \Phi_1 \overline{\mathbf{x}}}{\overline{\mathbf{x}}^\top \Phi_2 \overline{\mathbf{x}}} \ . \tag{2.4}$$

Fig. 2.5 shows three distinct members of the pencil, each meeting all the geometric constraints above.

The algorithm for extracting the bottom ellipse is as follows.

Ellipse Extraction from the VSOR Contour (Bottom Ellipse)

0. [*Initialization.*] Cut the imaged VSOR contour with the symmetry axis $\mathbf{l}_\perp$ so as to obtain the two branches ξ and ξ' of the contour, corresponding pointwise under the homology $\mathtt{F}$. Call $\mathbf{x}_{\text{inf}}$ the bottom intersection point of $\mathbf{l}_\perp$ with the contour.

1. [*Search.*] For every point $\mathbf{x} \in \xi$, do:

   1.0. Construct the conic pencil $\mathtt{C}_\mathbf{x}(\varphi)$ of Eq. 2.3, by computing the line $\mathbf{l}$ tangent to ξ at $\mathbf{x}$, and the corresponding entities on ξ' as $\mathbf{x}' = \mathtt{F}\mathbf{x}$ and $\mathbf{l}' = \mathtt{F}^{-\top}\mathbf{l}$.

   1.1. Choose the point $\overline{\mathbf{x}} \in \xi$ halfway between $\mathbf{x}_{\text{inf}}$ and $\mathbf{x}$, and compute the associated conic pencil member $\mathtt{C}_\mathbf{x}(\overline{\varphi})$ according to Eq. 2.4.

   1.2. For every point $\tilde{\mathbf{x}}$ in the VSOR contour segment $\mathbf{x}\frown\mathbf{x}'$ including $\mathbf{x}_{\text{inf}}$, compute the point-conic distance $d(\tilde{\mathbf{x}}, \mathtt{C}_\mathbf{x}(\overline{\varphi}))$ [29]. If such distance is large—say, above 1.5 pixels—, put $\tilde{\mathbf{x}}$ in the set $\mathcal{O}_\mathbf{x}$ of outliers w.r.t. $\mathtt{C}_\mathbf{x}(\overline{\varphi})$; otherwise mark it as an inlier and put it in the set $\mathcal{I}_\mathbf{x}$.

   1.3. Compute and store the difference $\Delta_\mathbf{x}$ between the cardinalities of $\mathcal{I}_\mathbf{x}$ and $\mathcal{O}_\mathbf{x}$.

2. [*First guess solution.*] Select the point $\mathbf{x} = \hat{\mathbf{x}}$ at which $\Delta_\mathbf{x}$ is maximum. Take $\mathtt{C}_{\hat{\mathbf{x}}}(\overline{\varphi})$ as a first guess solution for the bottom ellipse.

3. [*Refinement.*] Refine the first guess solution for the bottom ellipse by recursively performing inlier classification and least squares fitting with all the points of the VSOR contour. Run until the number of inliers reaches a constant value.

The segmentation algorithm for the top ellipse is the same as above, save that $\mathbf{x}_{\text{sup}}$, i.e., the top intersection point of $\mathbf{l}_\perp$ with the contour, is used in

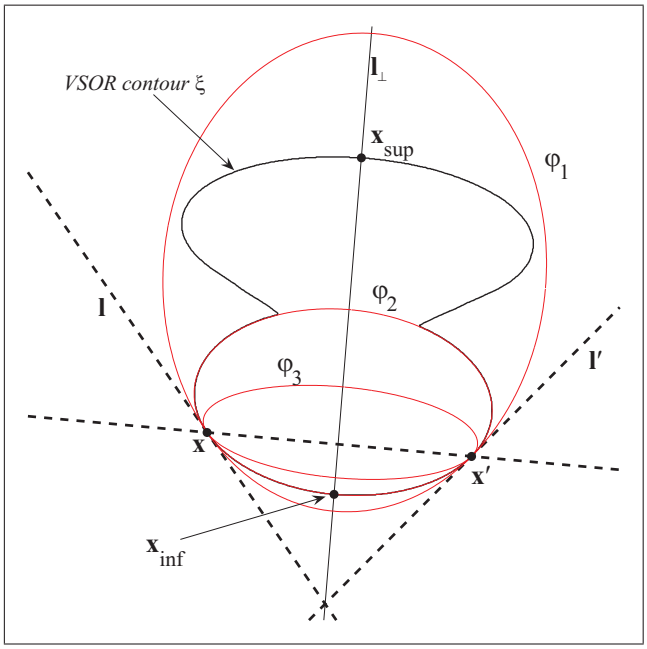


Figure 2.5: The pencil of conics at the tangent contact points $\mathbf{x}$ and $\mathbf{x}'$ of the imaged VSOR contour ξ. Three members of the pencil are reported for distinct values of the parameter φ.

the place of $\mathbf{x}_{\text{inf}}$. The maximum criterion used to estimate the first guess solution (step 1.3 of the algorithm) is motivated by the fact that, when fitting an ellipse inside a contour portion of elliptical shape, the number of expected outliers is zero. To achieve subpixel accuracy and also get a closed form expression for tangent lines, a spline-based interpolation scheme is used to represent the imaged VSOR contour as a smooth curve.

Once the circular points have been found by intersecting the imaged top and bottom VSOR cross-sections, the imaged turntable center can also be computed from the pole-polar relationship

$$\mathbf{x}_t = \mathtt{C}_t^{-1}\mathbf{l}_\infty \tag{2.5}$$

between the imaged bottom cross-section (represented by matrix $\mathtt{C}_t$) and the turntable vanishing line $\mathbf{l}_\infty$.

2.3 The Warping Approach

A textured three-dimensional object can be modelled as the set $\{(\mathbf{P}_i, \mathbf{Q}_i) \in \mathbb{R}^3 \times \mathbb{N}^3\}_{i=1}^N$, where $\mathbf{P}_i$ and $\mathbf{Q}_i$ represent respectively shape and texture through the spatial (XYZ) and chromatic (RGB) coordinates at each object point P_i. The next section deals with obtaining the point set $\{\mathbf{P}_i\}_{i=1}^N$; texture samples $\{\mathbf{Q}_i\}_{i=1}^N$ are addressed in §2.3.2.

2.3.1 3D Shape Reconstruction

This section discusses how shape reconstruction is carried out through metric rectification and collation of laser profiles.

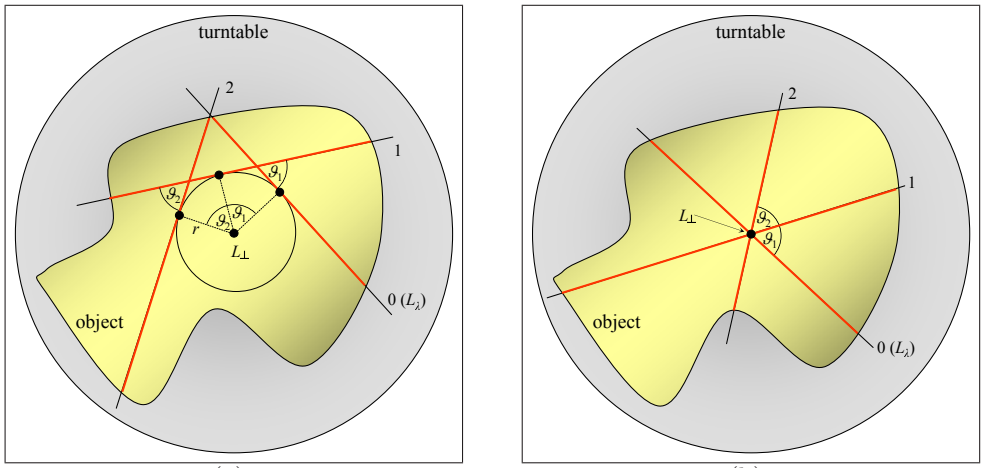


Figure 2.6: Laser-turntable geometry (top view). Laser lines 0 through 2 are obtained for three different relative laser-turntable positions. (a): The general case of laser plane *not* through the turntable axis. (b): The ideal case of laser plane through the turntable axis.

Fig. 2.6(a) shows that, in general, the laser plane is at distance r from the turntable axis; therefore, as the turntable rotates, the laser profiles are all tangent to a straight cylinder of radius r. For the sake of clarity, the main features of the shape reconstruction algorithm will be described in the next paragraph by referring to the ideal case $r = 0$, i.e., laser plane passing

through the turntable axis—see Fig. 2.6(b). The extension to the case $r \neq 0$ will then be discussed in the subsequent paragraph (The Real Case).

The Ideal Case

A well known result in uncalibrated vision is that is possible to rectify in a metric way (thus eliminating all projective distortions up to a similarity 2D transformation) the image of any plane for which the imaged circular points are known [29]. A theoretical settling of this topic with an original formula for deriving the whole family of rectifying transformations compatible with the circular points is provided in Appendix A.

Uncalibrated rectification is exploited here to obtain the metric shape of each laser profile from the laser plane's vanishing line $\mathbf{m}_\infty$ and the image of the absolute conic ω embedding internal calibration information. The imaged VSOR parameters are strictly related to the image of the absolute conic ω. In particular it holds $\mathbf{l}_\perp = \omega \mathbf{v}_\infty$. Moreover the turntable circular points in the image, $\mathbf{i}$ and $\mathbf{j}$, are also related to the image of the absolute conic as $\mathbf{i}^\top \omega \, \mathbf{i} = 0$ and $\mathbf{j}^\top \omega \, \mathbf{j} = 0$ [29]. The resulting system

$$\begin{cases} \mathbf{i}^\top \omega \, \mathbf{i} = 0 \\ \mathbf{j}^\top \omega \, \mathbf{j} = 0 \\ \mathbf{l}_\perp = \omega \mathbf{v}_\infty \end{cases} \tag{2.6}$$

provides four linear constraints on ω, whose coefficients can be computed from (the visible portions of) two imaged ellipses as shown in [15]. In that paper, it is demonstrated that only three out of the four constraints above are actually independent. Therefore, the system of Eq. 2.6 can be used to calibrate a square pixel camera (zero skew and unit aspect ratio: 3 dofs) from a single image.

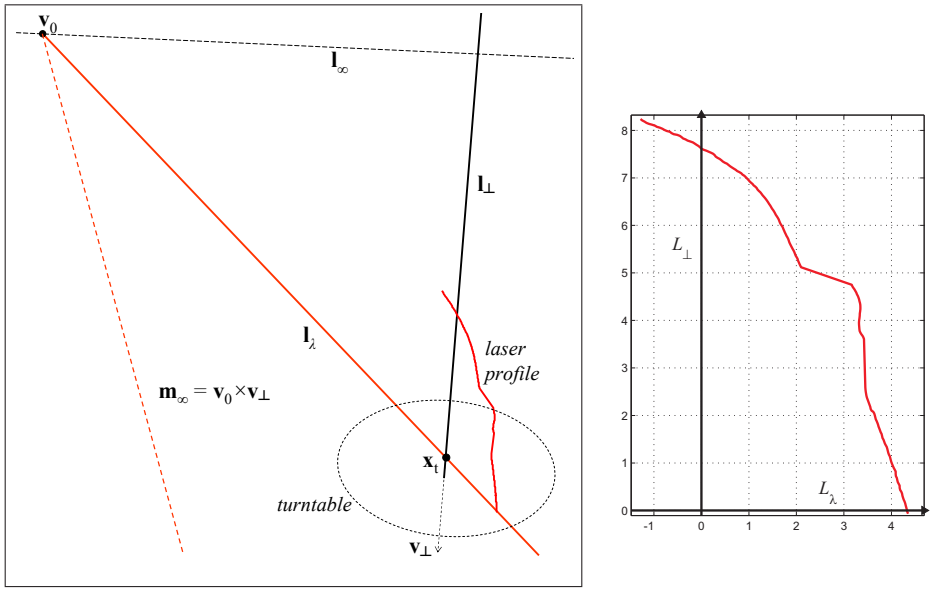
To rectify in a metric way each laser profile, the imaged circular points of the laser plane, namely $\mathbf{i}_\lambda = [\alpha + \imath\beta \;\; \gamma + \imath\delta \;\; 1]^\top$ and $\mathbf{j}_\lambda = \mathrm{conj}(\mathbf{i}_\lambda)$, can be computed by intersecting $\mathbf{m}_\infty$ with ω.

The planar homography modeling the rectifying transformation from the

image to the laser plane can then be obtained as (see Appendix A)

$$\mathtt{H}_{\mathrm{R}} = \begin{bmatrix} \delta & -\beta & 0 \\ -\gamma & \alpha & 0 \\ \delta & -\beta & -(\alpha\delta - \beta\gamma) \end{bmatrix} . \tag{2.7}$$

Fig. 2.7(a) shows a generic laser profile distorted by perspective projection, together with the geometric entities used for its rectification. Fig. 2.7(b) shows the rectified profile, where all distortions have been removed.



(a) (b)

Figure 2.7: (a): The geometry for laser profile rectification in the ideal case (image plane). (b): The rectified profile.

The vanishing line $\mathbf{m}_\infty$ is estimated as follows. First, the vanishing point of the rotation axis, $\mathbf{v}_\perp$, is obtained from the pole-polar relationship

$$\mathbf{l}_\infty = \omega \mathbf{v}_\perp \ . \tag{2.8}$$

The vanishing point of the laser-turntable line $\mathbf{l}_\lambda$ is then computed as $\mathbf{v}_0 = \mathbf{l}_\infty \times \mathbf{l}_\lambda$. Since $\mathbf{v}_0$ and $\mathbf{v}_\perp$ are the vanishing points of two distinct directions in the laser plane, $\mathbf{m}_\infty$ is simply computed as the line through these points—see again Fig. 2.7(a).

The homography of eq. 2.7 is also applied to rectify the imaged axis of rotation $\mathbf{l}_\perp$ and the imaged laser-turntable line $\mathbf{l}_\lambda$. These two lines are transformed respectively into the vertical ($L_\perp$) and horizontal (L_λ) coordinate axes of the rectified laser plane—see again Fig. 2.7(b)—and used to collate properly subsequent rectified profiles around the turntable axis at equally spaced angles. An estimate of the angle swept between subsequent frames is obtained from the number T of frames needed to cover a complete turntable rotation. This is evaluated by looking for the minimum value of the sum of square differences between the first frame and each subsequent frame of the natural light sequence.

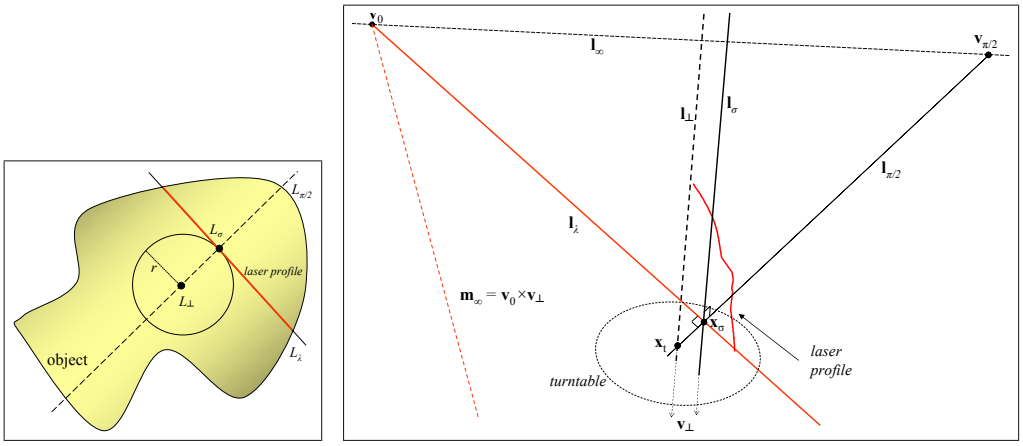
As shown in the example of Fig. 2.8, the output $\{\mathbf{P}_i\}_{i=1}^N$ of the profile collating operation has the form of a point cloud, where points are arranged in a radial way.

(a) (b)

Figure 2.8: (a): A papier-mache mask. (b): The reconstructed shape as a 3D point cloud (subsampled for visualization purposes).

The Real Case

The extension to the case $r \neq 0$ can be explained by referring to Fig. 2.9. Fig. 2.9(a) shows the laser line L_λ, the turntable axis $L_\perp$, and the vertical line L_σ where the laser plane is tangent to the cylinder of radius r—the latter two lines are orthogonal to the turntable plane, and are thus represented

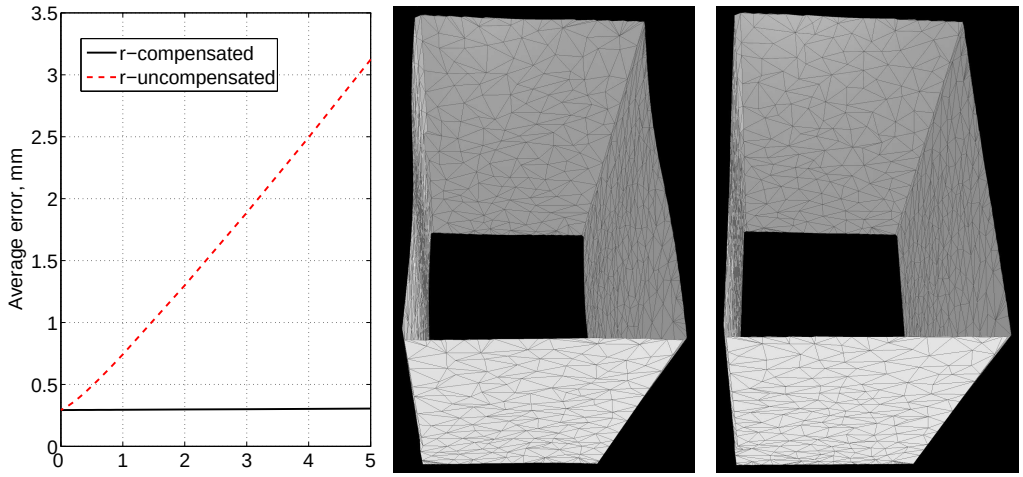


(a) (b)

Figure 2.9: Shape reconstruction in the real case $r \neq 0$. (a): 3D geometry (top view). (b): Image geometry.

in the figure as single points. Generalizing the approach explained in the ideal case $r = 0$, the three curves of the laser plane to be rectified for shape reconstruction are: The laser profile, L_λ and L_σ.

Now, while in the ideal case $r = 0$ it holds $L_\sigma = L_\perp$, in the real case L_σ has to be computed ad hoc directly from image measurements. To this aim, the geometrical method expounded in Appendix B is used to compute the imaged vanishing point $\mathbf{v}_{\pi/2}$ of the line $L_{\pi/2}$ orthogonal to L_λ in the turntable plane—see Fig. 2.9(b). The image of the point at which the vertical line L_σ pierces the turntable plane is then computed as $\mathbf{x}_\sigma = (\mathbf{x}_t \times \mathbf{v}_{\pi/2}) \times \mathbf{l}_\lambda$, where $\mathbf{x}_t$ is the imaged turntable center introduced in Subsection 2.3.1. The line L_σ is finally obtained by rectifying the image line $\mathbf{l}_\sigma = \mathbf{x}_\sigma \times \mathbf{v}_\perp$, where $\mathbf{v}_\perp$ is the vertical vanishing point computed in eq. 2.8. In the subsequent phase of profile collation, the correct placement of the rectified profile requires to compute not only the value of the rotation angle ϑ, but also that of the cylinder's radius r—refer again to Fig. 2.9(a). Before computing r, the scaling factors of the rectifying homographies of the laser and turntable planes have to be adjusted, so that any line segment on L_λ—belonging by construction to both the laser and turntable planes—has the same length after rectification



(a) (b)

Figure 2.10: The effect of compensating the distance r of the laser plane w.r.t. turntable axis for the reconstruction of the lateral surface of a box. (a): Average reconstruction error (mm) for $r \in [0, 5]$ mm. Dashed line: uncompensated case. Solid line: compensated case. (b) The reconstructed shape. Left: uncompensated case. Right: compensated case.

by either homography. That done, the radius r is obtained by rectifying the line segment connecting $\mathbf{x}_\sigma$ and $\mathbf{x}_t$, and measuring its length.

The beneficial effects of compensating for the laser-turntable distance r are now discussed. Fig. 2.10(a) shows (with solid and dashed lines, respectively) the average reconstruction error (mm) obtained with and without compensation for increasing values or r, starting from the ideal value $r = 0$. The error increases more than linearly with r in the uncompensated case, while it is basically constant in the compensated case. At a value of r as small as 1.5 mm, which is very likely to occur in a manual setup, the error in the uncompensated case is already 200% larger than in the compensated case. This clearly demonstrates the key importance of taking into account r in the shape reconstruction process. Fig. 2.10(b) shows the lateral faces of a box-shaped object reconstructed respectively without (left) and with (right) r-compensation. While in the compensated case the faces are correctly reconstructed as planar, in the uncompensated case they are quite

curved. This deformation is due to an incorrect collation of laser profiles. Another kind of deformation arises when the laser plane is not perfectly orthogonal to the turntable, affecting also the compensated model in terms of a residual reconstruction error of about 0.3 mm—see again Fig. 2.10(a). This deformation is mainly a projective one, making the lateral sides of the box to converge slightly. This is due to the fact that, since the reconstruction algorithm assumes the laser profile to be orthogonal to the turntable plane, it incorrectly confuses a straight object illuminated by a slanted laser plane with a truncated pyramidal object illuminated by a vertical laser plane.

2.3.2 Texture Sampling

Given a natural light image sequence $I_\tau(\mathbf{p})$, $\tau = 0 \ldots T-1$ (T is, as said before, the number of frames corresponding to a 360° rotation of the turntable), chromatic coordinates $\mathbf{Q}_i$ can be expressed as

$$\mathbf{Q}_i = I_{\overline{\tau}}(\mathbf{p}_i) \ , \tag{2.9}$$

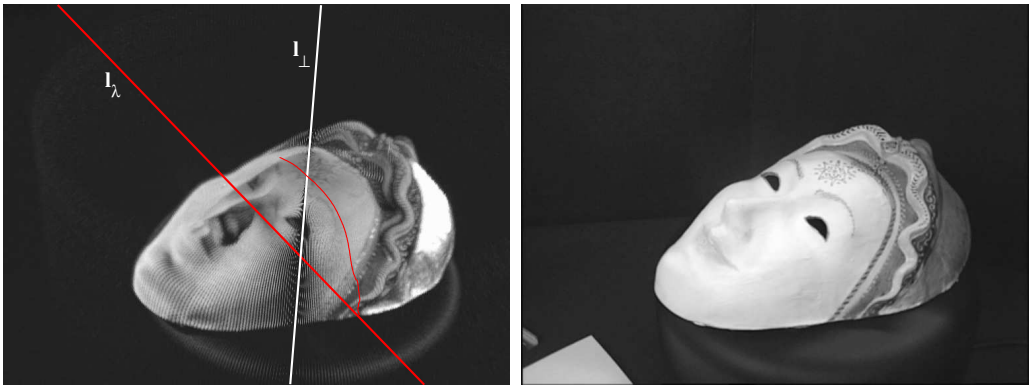
where $\mathbf{p}_i$ is the image projection of an object point P_i for some frame $\tau = \overline{\tau}$ at which the point is visible (i.e., not occluded). Texture sampling is tantamount to solve eq. 2.9 for all $i = 1 \ldots N$. This is typically carried out by a 3D-2D alignment procedure that exploits the shape point $\mathbf{P}_i$ to obtain the corresponding image point $\mathbf{p}_i$—see, e.g., [30]. The goal is to obtain one or more synthetic views of the object, each fully compatible with camera acquisition conditions and matching optimally the real object appearance in a different frame $\overline{\tau}$ of the natural light sequence. This procedure requires not only that the 3D shape model be available beforehand, but also that both internal and external camera calibration parameters be known.

In this work, a quite different approach is followed, that is based on 2D-2D alignment. With this approach, no external calibration is required. Moreover, no 3D shape model is required, so that texture sampling can take place independently from shape reconstruction, and at any time after internal calibration. A clear advantage over the traditional 3D-2D approaches is that

the 2D-2D approach relies on less parameters, thus being more robust with respect to any modelling and estimation inaccuracies. The key idea of the 2D-2D approach is *to exploit the laser light sequence* $I'_{\tau'}(\mathbf{p}),\ \tau' = 0 \ldots T-1$ *to solve eq. 2.9.* Indeed, every 3D object point P_i whose image $\mathbf{p}_i$ lies on a laser profile for some frame $\overline{\tau}'$ is also imaged *at the same point* $\mathbf{p}_i$ in some other frame $\overline{\tau}$ of the natural sequence. Hence, the texture sampling problem can be reformulated as a 2D-2D sequence alignment problem, i.e., finding the natural frame index $\overline{\tau}$ corresponding to each laser frame index $\overline{\tau}'$. Assuming a constant offset model

$$\tau = \tau' - \kappa \pmod{T} \ , \tag{2.10}$$

the sequence alignment problem is tantamount to estimating the unknown offset κ between the natural and laser sequences. Once κ is known, for any given laser frame index $\overline{\tau}'$ eq. 2.10 can be solved to get the natural frame index $\overline{\tau}$ to be used in eq. 2.9 and eventually sample the texture content for all points of the laser profile. Fig. 2.11 shows a typical result of warping and alignment of the laser profiles.



(a) (b)

Figure 2.11: (a): The synthetic view of the object obtained by frame-by-frame warping. The actual position of the laser plane and its intersection with the object are also reported. (b): The original object as seen from the same viewpoint.

An algorithmic solution for the sequence alignment problem in the ideal

case $r = 0$ is provided hereafter. The extension to the real case $r \neq 0$ will then follow.

Texture Sampling by Virtual View Synthesis (Ideal Case, $r = 0$)

1. [*Synthetic rotation of individual profiles.*] Generate a synthetic view of the laser sequence profile for every frame τ', as if it rotated around the turntable axis by an angle $\vartheta = \delta \cdot \tau'$, where δ (rad/frame) is the turntable rate. To this aim, apply the planar homology depending on ϑ

$$\mathtt{G}_\vartheta(\mathbf{l}_\perp, \mathbf{w}_\vartheta, \mu_\vartheta) = \mathtt{I}_{3\times 3} + (\mu_\vartheta - 1)\frac{\mathbf{w}_\vartheta\, \mathbf{l}_\perp^\top}{\mathbf{w}_\vartheta^\top\, \mathbf{l}_\perp} \quad , \tag{2.11}$$

   defined and computed as shown in Appendix B.

2. [*Synthetic silhouette construction.*] Construct a single silhouette image of the model by superimposing all the transformed profiles, and extract the silhouette contour.

3. [*Natural/laser sequence alignment.*] Register the synthetic silhouette contour against the real object appearance in the natural light sequence, thus obtaining the unknown natural/laser frame offset κ of eq. 2.10.

4. [*Texture sampling.*] For every profile point $\mathbf{p}_i$ of each laser frame $\overline{\tau}'$, sample the corresponding chromatic triplet $\mathbf{Q}_i$ using eqs. 2.9 and 2.10.

The extension of the previous algorithm to the real case can be explained by referring to Fig. 2.12. As shown in the figure, the laser plane and its rotated version by an angle ϑ no longer meet at the fixed line $L_\perp$, but meet instead at the line $L_h(\vartheta)$, parallel to $L_\perp$ and passing through the turntable point of intersection of the lines L_λ and $L_{(\pi-\vartheta)/2}$ (the bisectrix of the rotation angle). The image of $L_h(\vartheta)$, $\mathbf{l}_h(\vartheta)$, is the new axis of the planar homology realizing the laser plane virtual rotation in the image—see also Appendix B.

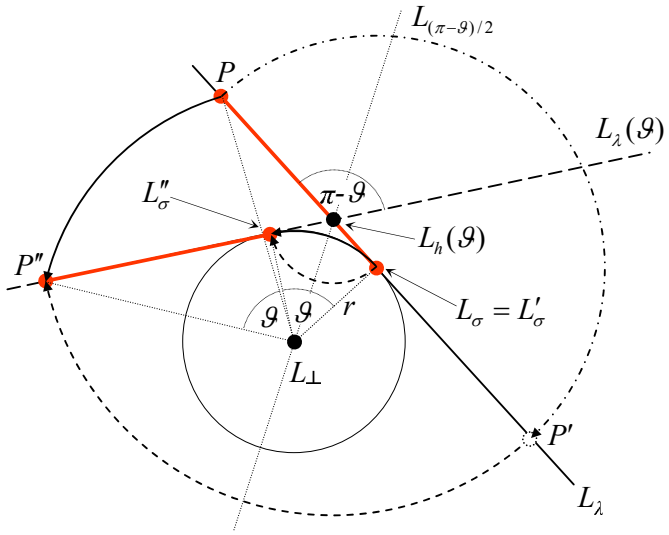


Figure 2.12: 3D geometry (top view) for texture sampling in the real case $r \neq 0$.

Such a line passes through the vertical vanishing point $\mathbf{v}_\perp$ and the intersection point between $\mathbf{l}_\lambda$ and the imaged bisectrix line $\mathbf{l}_{(\pi-\vartheta)/2} = \mathbf{x}_t \times \mathbf{v}_{(\pi-\vartheta)/2}$, where the vanishing point of $L_{(\pi-\vartheta)/2}$ is computed as in Appendix B. Fig. 2.12 also shows the laser-turntable line $L_\lambda(\vartheta)$ and the associated profile after the virtual rotation ϑ. Transforming any point P in the laser profile into its corresponding point $P''(\vartheta)$ in the virtually rotated laser plane can be accomplished in two steps. First, P undergoes a specular reflection w.r.t. L_σ. The point $P'(\vartheta)$ thus obtained is then rotated by $\pi - \vartheta$, thus making it reach its final destination $P''(\vartheta)$. In the image plane, such a two-step transformation is accomplished by the composition of the harmonic homology

$$\mathtt{F}(\mathbf{l}_\sigma, \mathbf{v}_0) = \mathtt{I}_{3\times 3} - 2\frac{\mathbf{v}_0\,\mathbf{l}_\sigma^\top}{\mathbf{v}_0^\top\,\mathbf{l}_\sigma} \tag{2.12}$$

with the planar homology $\mathtt{G}_{\pi-\vartheta}(\mathbf{l}_h(\vartheta), \mathbf{w}_{\pi-\vartheta}, \mu_{\pi-\vartheta})$ (see eq. 2.11), where the parameters $\mathbf{v}_0$, $\mathbf{w}_{\pi-\vartheta}$ and $\mu_{\pi-\vartheta}$ are computed as expounded in Appendix B. Thus, the imaged profile point $\mathbf{p}$ is transformed into its corresponding point $\mathbf{p}''(\vartheta) = \mathtt{H}(\vartheta)\mathbf{p}$ through the overall homography

$$\mathtt{H}(\vartheta) = \mathtt{G}_{\pi-\vartheta}(\mathbf{l}_h(\vartheta), \mathbf{w}_{\pi-\vartheta}, \mu_{\pi-\vartheta})\mathtt{F}(\mathbf{l}_\sigma, \mathbf{v}_0) \quad . \tag{2.13}$$

Fig. 2.13 shows an example of shape reconstruction and texture sampling in the case $r = 0.5$ mm. The acquired 3D model has a (invisible) "hole" of radius r in the forehead, due to the absence of a laser trace inside the cylinder of radius r surrounding the turntable axis.

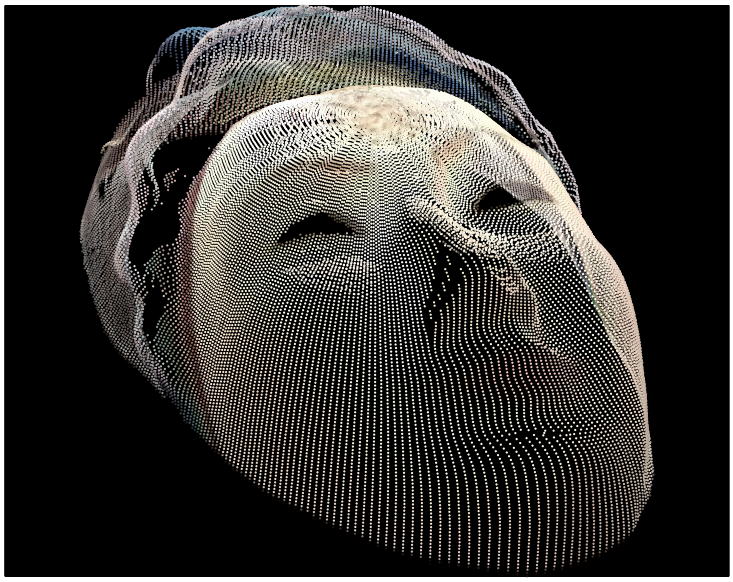


Figure 2.13: The complete (shape+texture) 3D model obtained for the object of Fig. 2.8(a). The point cloud was subsampled for visualization purposes.

2.4 A VSOR-based Triangulation Approach

The 3×4 camera projection matrix

$$\mathtt{P} = \mathtt{K}\mathtt{R}[\mathtt{I}_{3\times 3} \mid -\mathbf{c}] \tag{2.14}$$

relating a world point $\mathbf{X}$ with its image $\mathbf{x}$ as $\mathbf{x} = \mathtt{P}\mathbf{X}$ (homogeneous coordinates), represents in a compact way all information about camera optics (internal parameters, 5 dof) and camera position with respect to the world reference frame (external parameters, 6 dof). In particular, the 3×3 matrix $\mathtt{K}$ takes into account the internal camera parameters, while the external parameters are embedded in the rotation matrix $\mathtt{R}$ (3 dofs) between the world

and the camera frames, and in the 3-vector $\mathbf{c}$, expressing the camera center in (inhomogeneous) world coordinates.

Without loss of generality, the world frame origin can be taken as the center of the turntable, and the Z axis as the turntable axis; moreover, the camera center can be assumed to lie on the half plane $\mathrm{X} > 0$, $\mathrm{Y} = 0$.

2.4.1 Camera Calibration

The VSOR fixed entities of §2.2 can be exploited to perform a full (i.e. internal and external parameters) camera calibration.

Since $\omega = (\mathtt{K}\mathtt{K}^\top)^{-1}$, the internal calibration matrix $\mathtt{K}$ can be obtained by the Choleski decomposition of ω, followed by inversion [29].

External calibration parameters are computed by specializing the approach first proposed in [12]. The method exploits the knowledge of (1) the imaged turntable rotation axis $\mathbf{l}_\perp$, (2) the turntable vanishing line $\mathbf{l}_\infty = \mathbf{i} \times \mathbf{j}$, and (3) the image turntable center $\mathbf{x}_t$.

The first step is the computation of the rotation matrix $\mathtt{R} = [\mathbf{n}_\mathrm{X} \ \mathbf{n}_\mathrm{Y} \ \mathbf{n}_\mathrm{Z}]$, where $\mathbf{n}_\mathrm{X}$, $\mathbf{n}_\mathrm{Y}$, and $\mathbf{n}_\mathrm{Z}$ are unit vectors. It is well known that, given a point image $\mathbf{p}$ in homogeneous coordinates, the inhomogeneous 3-vector $\mathtt{K}^{-1}\mathbf{p}$ represents the direction (with respect to the camera frame) of the ray passing through the camera center and $\mathbf{p}$ [29]. Therefore, if any two points on the line $\mathbf{l}_\perp$ are chosen, two vectors lying on the plane $\mathrm{Y} = 0$ can be determined, whose normalized cross product equals the unit vector $\mathbf{n}_\mathrm{Y}$. The same procedure can be applied to compute the unit vector $\mathbf{n}_\mathrm{Z}$ from two points properly chosen on the vanishing line $\mathbf{l}_\infty$. Finally, the unit vector $\mathbf{n}_\mathrm{X}$ can be computed as $\mathbf{n}_\mathrm{Y} \times \mathbf{n}_\mathrm{Z}$. Fig. 2.14 shows three points that can be conveniently chosen for obtaining the rotation matrix. These are: (1) the harmonic homology vertex $\mathbf{v}_\infty \in \mathbf{l}_\infty$; (2) the image of the world origin $\mathbf{x}_t \in \mathbf{l}_\perp$, (3) the intersection $\mathbf{x}_i = \mathbf{l}_\perp \times \mathbf{l}_\infty$ between $\mathbf{l}_\perp$ and $\mathbf{l}_\infty$. As the matrix $\mathtt{R}$ thus computed is seldom a rotation matrix, a final refinement step based on the SVD decomposition is carried out to obtain the best orthogonal approximation to $\mathtt{R}$.

Referring to Fig. 2.14, the imaged Z axis ($\mathbf{l}_\perp$) is oriented from $\mathbf{x}_t$ to $\mathbf{x}_i$.

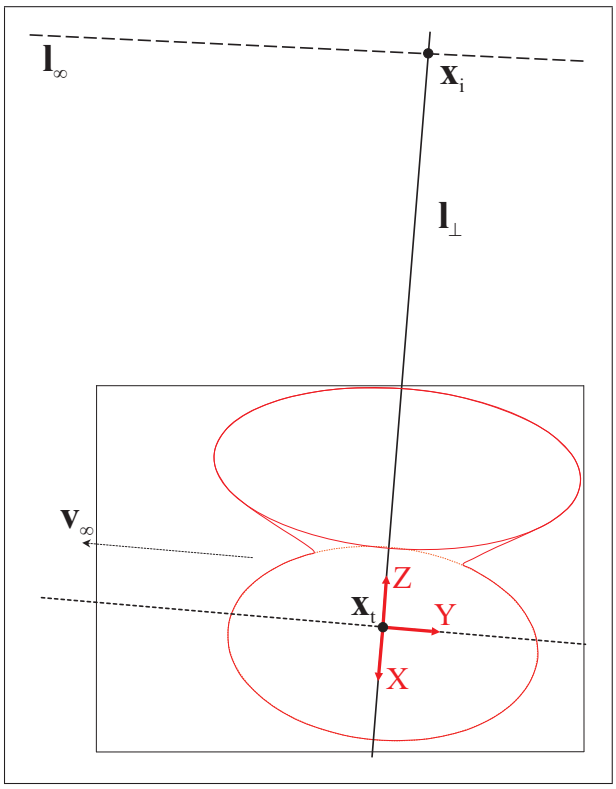


Figure 2.14: Lines and points needed for rotation matrix computation. The world frame is superimposed and centered in $\mathbf{x}_t$.

Since the X coordinate of the camera center is positive, the vector

$$\mathbf{m}_{\mathrm{Y}} = (\mathtt{K}^{-1}\mathbf{x}_t) \times (\mathtt{K}^{-1}\mathbf{x}_i) = \mathtt{K}^{\top}(\mathbf{x}_t \times \mathbf{x}_i)$$

must have the same direction as the Y axis, in order to obtain a right-hand world frame. The vector $\mathbf{m}_{\mathrm{Z}}$ orthogonal to the plane $\mathrm{Z} = 0$ and directed as the Z axis must then be obtained as

$$\mathbf{m}_{\mathrm{Z}} = (\mathtt{K}^{-1}\mathbf{x}_i) \times (\mathtt{K}^{-1}\mathbf{v}_{\infty}) = \mathtt{K}^{\top}(\mathbf{x}_i \times \mathbf{v}_{\infty}) \ .$$

The unit vectors $\mathbf{n}_{\mathrm{Y}}$ and $\mathbf{n}_{\mathrm{Z}}$ are finally obtained by normalization of $\mathbf{m}_{\mathrm{Y}}$ and $\mathbf{m}_{\mathrm{Z}}$, respectively.

To complete external calibration, the camera center $\mathbf{c}$ must be computed. Any point on the plane $\mathrm{Z} = 0$ is mapped onto the image by the homography $\mathtt{H}_0$ given by:

$$\mathbf{x} = \mathtt{P} \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_4 \end{bmatrix} \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ 1 \end{bmatrix} = \mathtt{H}_0 \begin{bmatrix} \mathrm{X} \\ \mathrm{Y} \\ 1 \end{bmatrix} \ , \tag{2.15}$$

where $\mathbf{p}_i$ is the i-th column of $\mathtt{P}$.

In particular, the center of the bottom cross section is projected onto the inhomogeneous point with pixel coordinates (x_t, y_t), whose corresponding homogeneous vector is

$$\sigma\mathbf{x}_t = \sigma \begin{bmatrix} x_t \\ y_t \\ 1 \end{bmatrix} = \mathtt{H}_0 \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \mathbf{p}_4 = -\mathtt{KR}\mathbf{c} \ ;$$

hence,

$$\mathbf{c} = -\sigma\mathtt{R}^{\top}\mathtt{K}^{-1}\mathbf{x}_t \ . \tag{2.16}$$

At this moment, let the scaling factor be $\sigma = 1$: i.e. the method supports a *metric* reconstruction. If a Euclidean reconstruction is required, the scaling factor can be fixed given one length in the scene. In [12] it is shown how to set the right value for σ if the real radius of the VSOR bottom cross-section is known; the height of the VSOR (which is also the height of the object used to generate it) can be also conveniently used for this purpose.

2.4.2 Laser Calibration

As mentioned above, calibrating the laser is tantamount to computing the position of the laser plane in world coordinates. The line $\mathbf{l}_\lambda$ is the image of a world line $\mathbf{L}_\lambda$ that lies on the turntable plane ($\mathrm{Z} = 0$). Therefore, it holds $\mathbf{L}_\lambda = \mathtt{H}_0^\top \mathbf{l}_\lambda$. Denoted the world line as $\mathbf{L}_\lambda = [L_1 \;\; L_2 \;\; L_3]^\top$, the required world representation of the laser plane is $\Pi_\lambda : [L_1 \;\; L_2 \;\; 0 \;\; L_3]^\top$ such that

$$L_1 \mathrm{X} + L_2 \mathrm{Y} + L_3 = 0 \;\; . \tag{2.17}$$

2.4.3 Metric Reconstruction

Once both the camera projection matrix $\mathtt{P}$ and the laser plane Π_λ are known, the 3D coordinates of any point $\mathbf{X} \in \Pi_\lambda$ can be computed from its image $\mathbf{x}$ as the intersection of the laser plane with the visual ray through $\mathbf{x}$:

$$\mathbf{X} = \mathtt{P}^\dagger \mathbf{x} \cap \Pi_\lambda \;\; , \tag{2.18}$$

where $\mathtt{P}^\dagger = (\mathtt{P}^\top \mathtt{P})^{-1} \mathtt{P}^\top$ is the pseudo-inverse of $\mathtt{P}$ [29].

Shape reconstruction of 3D object shape is obtained by detecting, in each frame of the turntable sequence, the imaged laser profiles, then evaluating the 3D coordinates of all points of the laser profile, and finally collating at equally spaced angles all profiles according to their subsequent positions inside the sequence. A reconstruction example is shown in Fig. 2.15.

Texture acquisition could be performed following the guidelines discussed at the beginning of §2.3.2 exploiting the reprojection of the 3D shape model onto the image via $\mathtt{P}$.

2.5 Experimental Results

Experiments were carried out using a Sony EVI-D31 camera with 768×576 pixels (corresponding to a camera resolution of about 0.4 Mpixels), an off-the-shelf (≈$100) laser illuminator, and a Kaidan Pixi-M motorized turntable with a diameter of 10” (25.4 cm) and a nominal speed of 4 rpm (24 degrees/s).

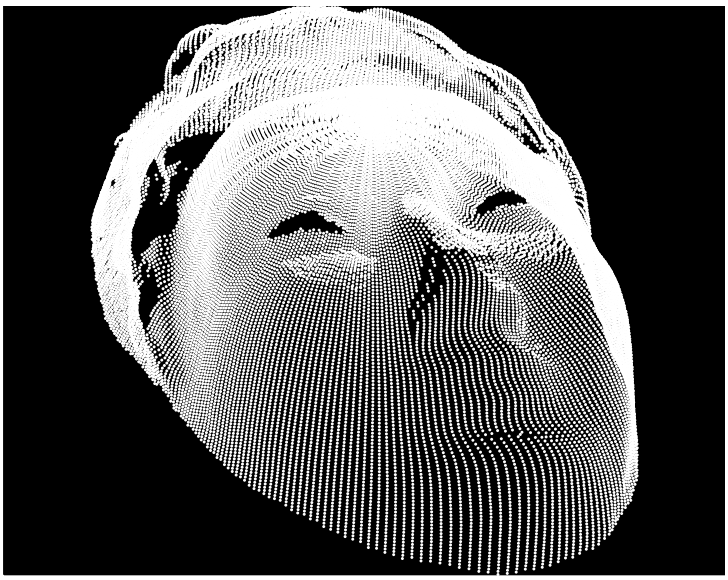


Figure 2.15: The (sub-sampled) point cloud for the mask of Fig. 2.8(a) obtained by the VSOR-based triangulation approach.

For the purpose of performance evaluation and comparison, a traditional active triangulation approach, referred to in the following as STANDARD, was developed. STANDARD employs the methods for camera calibration, shape reconstruction, and texture sampling most commonly used in commercial desktop scanners. STANDARD was compared with two different implementations of the proposed approach, referred to in the following as ARVS (acronym for "Active Rectification and virtual View Synthesis"). turntable fixed entities for self-calibration, while ARVS$_{\mathcal{C}}$ uses the same method of STANDARD, based on a checkerboard pattern. As shown in Tab. 2.1, only STANDARD requires the six external camera calibration parameters encoded in the rotation matrix $\mathtt{R}$ and translation vector $\mathbf{t}$. On the other hand, all approaches require the three natural camera internal parameters, encoded in the calibration matrix $\mathtt{K}$ such that $\omega^{-1} = \mathtt{K}\mathtt{K}^{\top}$. Full camera calibration of STANDARD method is here performed with a modified version of the classic Tsai's algorithm [55], more accurate than the VSOR-based approach of §2.4.1, because the experiment aims to compare conventional techniques vs. the proposed warping-based approach ARVS. The layout used throughout the experiments

Table 2.1: Approaches tested in the experiments.

approach	*camera calibration*		*shape reconstruction*	*texture sampling*
	device	*parameters*		
STANDARD	checkerboard	$\mathtt{K}$, $\mathtt{R}$, $\mathbf{t}$ [55]	active triangulation [3]	3D model reprojection [30]
ARVS$_{\mathcal{C}}$	checkerboard	$\mathtt{K}$ only	profile rectification (§2.3.1)	profile synthetic rotation (§2.3.2)
ARVS$_{\mathcal{T}}$	fixed entities	$\mathtt{K}$ only [15]	profile rectification	profile synthetic rotation

is the following:

- relative position of camera and turntable centers: ≈50 cm horizontal, ≈40 cm vertical;
- camera orientation: ≈10 degrees pan left w.r.t. the vertical plane including the turntable axis and the camera center, ≈30 degrees downward tilt;
- laser plane *manually* placed so as to pass through the turntable axis and be at 35 degrees w.r.t. the abovementioned camera-turntable plane.

The layout chosen allows a cylindrical acquisition volume of about 8200 cm^3.

2.5.1 Tests with a Reference Object

As a reference object for quantitative evaluations, a wooden block of dimensions (in mm) $160 \times 120 \times 80$ was used. Having sides of different length along three orthogonal directions, the chosen reference object allows investigating the shape reconstruction performance of the various approaches in general conditions. In all acquisitions, the reference object was placed upon the turntable with its longest edge in the vertical direction.

Shape Reconstruction

The overall quality of both Euclidean and metric—i.e., up to an overall scaling factor—shape reconstruction with the three approaches of Table 2.1 was checked in terms of absolute dimensions and relative proportions of the reference object. To obtain a quantitative measure of the side lengths of the reconstructed reference object, a stretchable box-like model was fitted using

a robust least squares algorithm to the 3D point cloud obtained after shape reconstruction.

Fig. 2.16 shows a top view of the block reconstructed with ARVS$_{\mathcal{C}}$—similar results are obtained with the other two methods. The encircled areas at the bottom left and top right of the figure indicate the absence, in the reconstructed point cloud, of some model points. Although illuminated by the laser, such *points of occlusion* were not visible from the camera viewpoint, due to the relative position of camera, object, and laser plane.

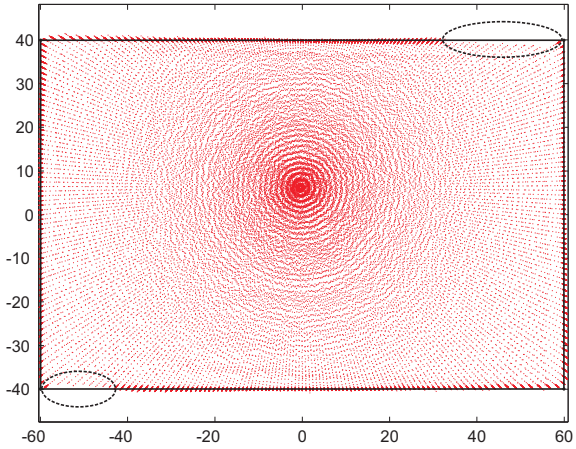


Figure 2.16: The 3D point cloud obtained for the block object with ARVS$_{\mathcal{C}}$ (top view). The least squares box model is shown with solid lines. Note the radial arrangement of the cloud points.

Assuming STANDARD's calibration method as the ground truth, ARVS$_{\mathcal{T}}$'s calibration was reasonably accurate—in terms of percentage error—for the focal (≈1%) and the x coordinate of the principal point (≈3%), while the y coordinate of the principal point was not as satisfactory (≈11%). Tab. 2.2 reports the results of shape reconstruction. Notwithstanding the worse calibration accuracy, ARVS$_{\mathcal{T}}$*'s performance was as good as* STANDARD*'s at the task of shape reconstruction*, being slightly less accurate at measuring absolute lengths, and superior at estimating real proportions—which are in fact obtained with no error at all. This suggests that ARVS$_{\mathcal{T}}$ would outperform STANDARD, if calibration data were equal.

Table 2.2: Shape reconstruction results for the reference block object. The ground truth is $160 \times 120 \times 80$ (dimensions, mm), and $2 : 1.5 : 1$ (proportions, —).

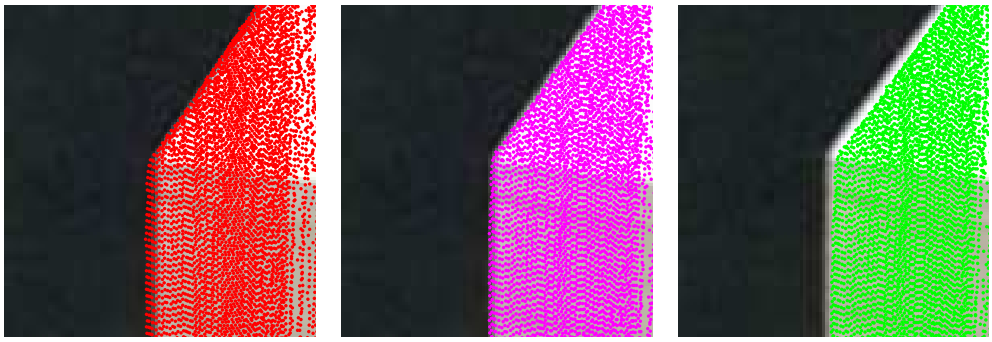
approach	*shape reconstruction*	
	Euclidean	metric
STANDARD	$160.6 \times 119.8 \times 79.9$	$2.00 : 1.49 : 0.99$
ARVS$_{\mathcal{C}}$	$160.0 \times 120.3 \times 80.2$	$2.00 : 1.50 : 1.00$
ARVS$_{\mathcal{T}}$	$159.0 \times 119.4 \times 79.5$	$2.00 : 1.50 : 1.00$

Indeed, as shown in the second row of Tab. 2.2, *the best performance is obtained by* ARVS$_{\mathcal{C}}$, that uses STANDARD's calibration method and ARVS$_{\mathcal{T}}$'s shape reconstruction method. Reconstruction results also show that the performance gap between ARVS$_{\mathcal{C}}$ and ARVS$_{\mathcal{T}}$ (which differ only in the calibration method) is not as large as the one between ARVS$_{\mathcal{C}}$ and STANDARD (which differ only in the algorithms for shape reconstruction and texture sampling). This can be explained by the fact that, to carry out shape reconstruction, both ARVS$_{\mathcal{T}}$ and ARVS$_{\mathcal{C}}$ work solely with 2D information, while STANDARD also requires 3D information—namely, the external camera parameters and the laser plane position in space—, thus being much more sensitive to any estimation inaccuracies.

Texture Sampling

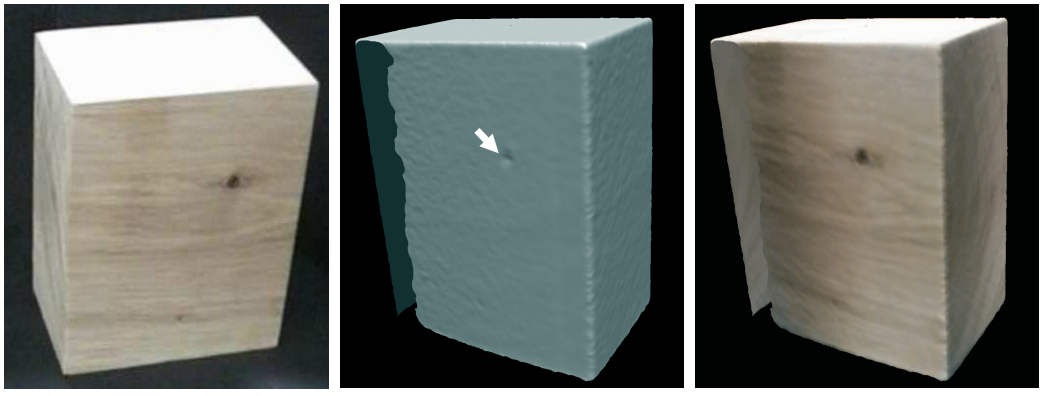
A similar sensitivity to 3D inaccuracies is exhibited by STANDARD in the texture sampling task. Here, the main difference with the other two approaches consists in the way the synthetic view of the 3D model is generated. As before, STANDARD exploits 3D information, while ARVS$_{\mathcal{T}}$ and ARVS$_{\mathcal{C}}$ use only the more reliable 2D data. Fig. 2.17 shows the synthetic views generated with the three approaches. As in the case of shape reconstruction, *the best results are obtained with* ARVS$_{\mathcal{C}}$*, while the performance of the other two approaches is similar*. In particular, while ARVS$_{\mathcal{C}}$'s virtual view is almost perfectly aligned to image data, STANDARD's is larger than the original, and ARVS$_{\mathcal{T}}$'s is smaller. Any alignment error gives rise to slightly incorrect textures, with background colors associated to some of the foreground pixels.

The complete textured model obtained with ARVS$_{\mathcal{C}}$ is shown in Fig. 2.18.



(a) (b) (c)

Figure 2.17: Accuracy of virtual view generation for texture sampling with the three approaches. The top part of the wooden block is shown with the virtual view superimposed. (a): STANDARD. (b): ARVS$_{\mathcal{C}}$. (c): ARVS$_{\mathcal{T}}$.



(a) (b) (c)

Figure 2.18: The complete model obtained with ARVS$_{\mathcal{C}}$. (a): A close view of the reference object. (b): The model, with evidence of shape details. (c): The final model with superimposed texture.

Notice that the slight shape depression corresponding to the node of the wooden object (a) was correctly located (b) and textured (c) in the model. Moreover, the figure shows the real extent of the occluded portion (see also Fig. 2.16).

To summarize the results obtained with the three approaches, ARVS$_{\mathcal{C}}$ is the best in all cases, while ARVS$_{\mathcal{T}}$ is better than STANDARD at metric reconstruction, worse at Euclidean reconstruction, and comparable at texture sampling. The fact that the STANDARD approach—representing a large class of commercial systems—and ARVS$_{\mathcal{T}}$—i.e., the most "experimental" of the three approaches, relying on a non standard and less accurate calibration method—have a similar performance, witnesses the robustness of the proposed acquisition algorithms with respect to calibration data.

2.5.2 Tests with Complex Objects

The ARVS$_{\mathcal{C}}$ approach was further tested at the production of 3D textured models of three complex-shaped objects: A papier-mache mask, a monk ceramic statuette, and a lacquered wooden horse—see Fig. 2.19. Depending on its peculiar shape and/or superficial properties, every object provided diverse acquisition conditions, thus encompassing three broad classes of object typologies. In particular, the papier-mache mask is a basically flat and wide object, while the monk has a vertically elongated shape, with a few self-occluded parts that hamper a complete object acquisition in a single scanning session. The horse has even more concavities, that give rise to several occlusions regardless of the way the object is placed upon the turntable for scanning. Another difficulty with the horse arises from object texture. Indeed, the painted decoration has several dark spots, where the laser stripe fails to be detected: This prevents these spots from being reconstructed at all, regardless of object pose. However, the presence of occlusion-induced holes in the reconstructed model is common to all scanning approaches, and, similarly, texture-induced holes are unavoidable with any laser-based approach.

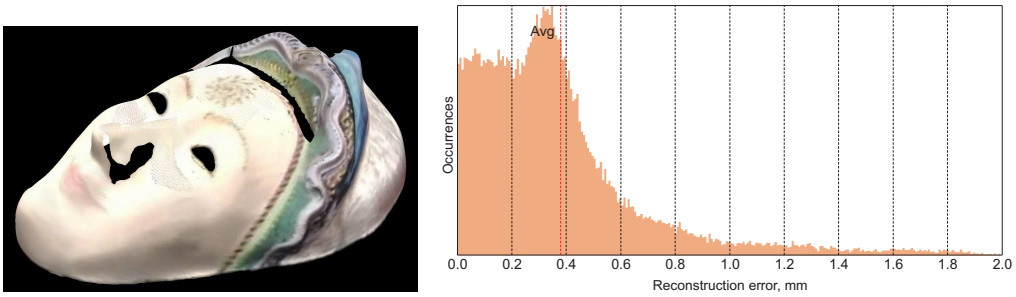
The acquired models were compared with ground truth models created

(a) (b) (c)

Figure 2.19: The three different object typologies using for the tests. (a): Papier-mache mask (size: 104×175×270 mm). (b): Ceramic statuette (size: 165×72×93 mm). (c): Lacquered wooden horse (size: 168 × 150 × 45 mm).

with the NextEngine commercial desktop 3D scanner, whose technical specifications are available on the web at `https://www.nextengine.com`. The shape reconstruction error is obtained as $|\mathcal{E}|$, where $\mathcal{E}$ is the signed distance between each reconstructed 3D point and the ground truth surface after registration of the acquired model against the ground truth.

Fig. 2.20 shows the results of a single 3D acquisition session for the mask. During acquisition, the mask was placed horizontally on the turntable. Beside the unavoidable occlusions (occurring around the nose and along the top



(a) (b)

Figure 2.20: (a): Textured model for the mask. (b): Distribution of reconstruction error $|\mathcal{E}|$ (mm).

forehead for the chosen relative camera-laser-object layout), the obtained 3D textured model faithfully reproduces both the shape and the texture of the original—see Fig. 2.20(a). Error statistics are shown in Fig. 2.20(b): The maximum error is 1.85 mm in the far tail of the distribution, while the average error is 0.37 mm.

The acquisition results (single session) for the monk statuette are reported in Fig. 2.21. The figure shows two different views of the reconstructed textured model. As in the previous case, some occlusions occur. Notice, in particular, that the open book prevented the laser light to reach a part of the monk's habit, giving rise to a "hole" in the reconstructed object.

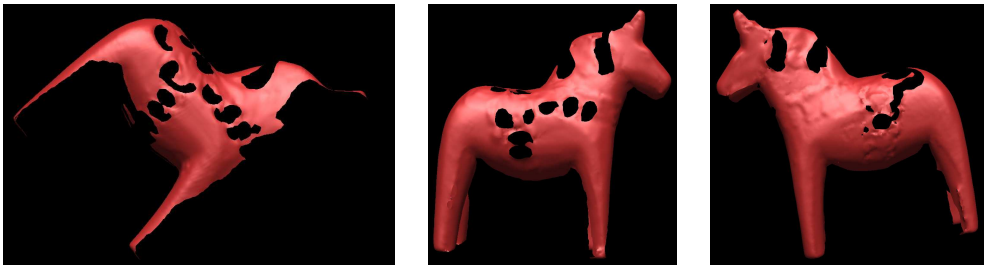
(a) (b) (c)

Figure 2.21: (a) and (b): Textured model for the monk statuette using two different viewpoints. (c): Signed reconstruction error $\mathcal{E}$ (mm). Color codes: white for $\mathcal{E} > 0$, light gray for $-1 \leq \mathcal{E} \leq 0$, dark gray for $\mathcal{E} < -1$.

The shape reconstruction has a maximum value of 2.68 mm (far distribution tail), and an average value of 0.42 mm. Fig. 2.21(c) uses a color code to display the sign and quantized magnitude of the signed error $\mathcal{E}$: Dark gray corresponds to a large negative error, light gray to a small negative error, white to a positive error. Notice that the positive errors are concentrated in

the bottom part of the object, while the negative errors are in the top part. This pattern is explained by considering that, due to calibration inaccuracies and to a slight departure from the vertical of the laser plane's orientation, the reconstructed model is affected by a residual 3D projective distortion (see also Subsection 2.3.1). The effects of this distortion are to dilate the object at the bottom—i.e., near the turntable—and to shrink it progressively going towards the top. Since the extent of the distortion is very small, its effects are visible only if the object scanned has a vertically elongated shape, while they are negligible for basically flat objects, like the mask.

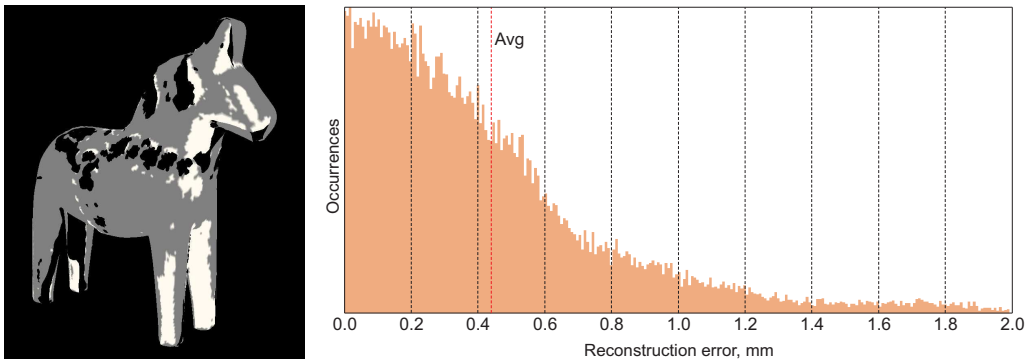
The acquisition results for three different scanning session for the horse are reported in Fig. 2.22. In particular, Fig. 2.22(a) shows the partial model obtained with the object standing on the front hooves and the muzzle, while Figs. 2.22(b) and (c) show the texture-induced holes in the partial models obtained by laying out the object on its left and rights sides.



(a) (b) (c)

Figure 2.22: Horse models acquired with the object in different positions on the turntable. (a): Prone position. (b): Laid out on the left side. (c): Laid out on the right side.

In Fig. 2.23, the shape reconstruction error is shown. The maximum error has a magnitude of 1.99 mm, while the average error is 0.44 mm. The partial models of Fig. 2.22 were merged together with a 3D graphics software so as to obtain the complete model of Fig. 2.24. While ARVS$_{\mathcal{C}}$, that exploits a 360 degrees object rotation, required only three object poses to obtain a complete model, six scanning sessions (front, back, left, right, top and bottom side) were required with NextEngine to obtain the complete ground truth model.



(a) (b)

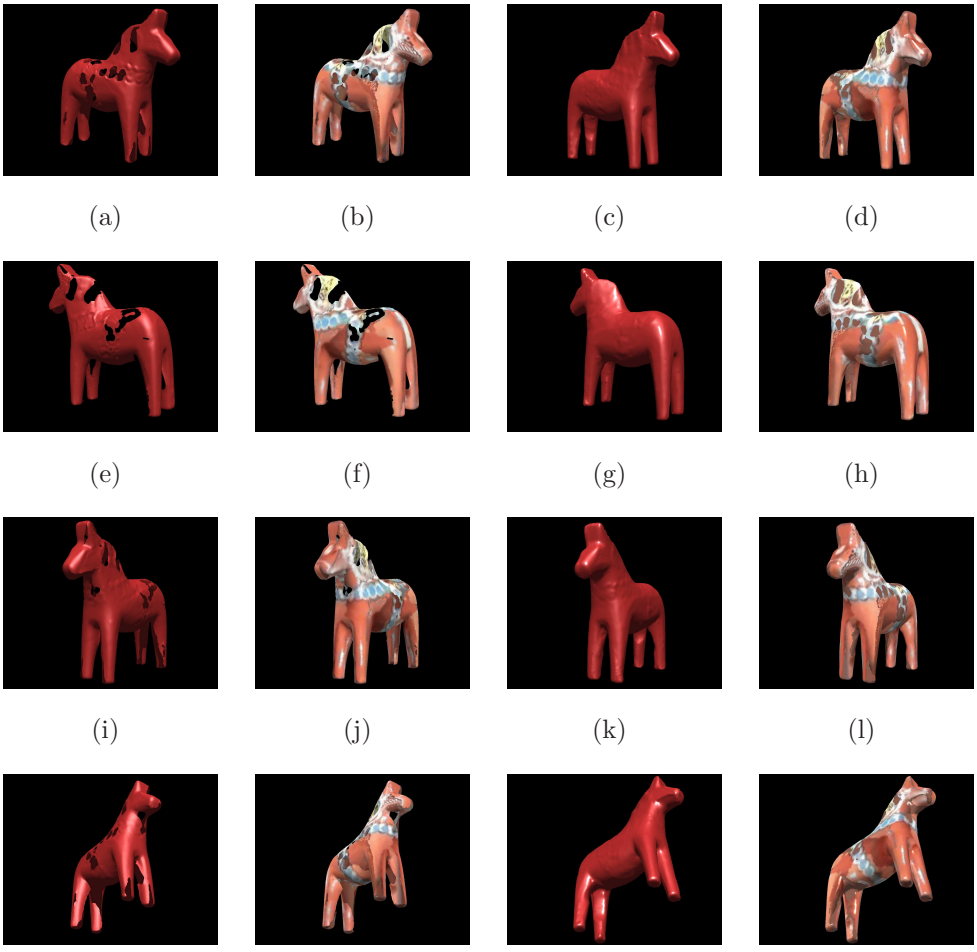
Figure 2.23: (a): Signed reconstruction error $\mathcal{E}$ for the horse model. Color codes: Gray for $\mathcal{E} > 0$, white for $\mathcal{E} \leq 0$. (b): Distribution of $|\mathcal{E}|$ (mm).

2.5.3 Error Analysis

The results of the previous Section indicate that, for a camera resolution of 0.4 Mpixels and the layout reported at the beginning of this section, the shape reconstruction error $|\mathcal{E}|$ has a low rms statistical distribution, with an average of about 0.4 mm. The shape reconstruction error can be regarded as the combination of two errors:

- A *setup error*, due to inaccuracies both in the estimation of apparent fixed entities and in the layout of system elements (e.g., laser plane not orthogonal w.r.t. the turntable—see Subsection 2.3.1 and comment to Fig. 2.21(c));
- A *resolution error*, due to the fact that any inaccuracy in the estimation of laser profiles gives rise to a reconstruction error that is inversely proportional to the spatial resolution in the acquisition volume.

In particular, the resolution error is the reciprocal of the average spatial resolution (in pixels/mm), and depends on both the camera-turntable-laser relative placement, and camera resolution (in Mpixels) employed. The effects of the resolution error on the overall reconstruction error can be controlled by a proper choice of the camera resolution and the angle between the laser plane



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

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

Figure 2.24: Merged model for the horse observed from several viewpoints. First column: Shape with holes, due to both occlusions and dark texture spots. Second column: Textured shape with holes. Third column: Shape with manually filled holes. Fourth column: Textured shape with filled holes.

and the plane through the camera center and the turntable axis ("camera-laser angle"). Fig. 2.25 shows the expected resolution error at each pixel of the imaged laser plane. In the figure, darker pixels correspond to higher error values. The error increases with point distance from the camera.

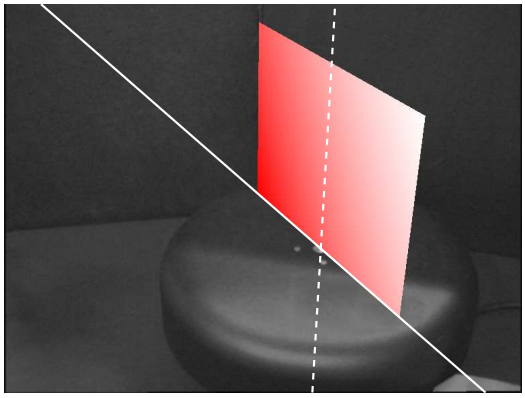
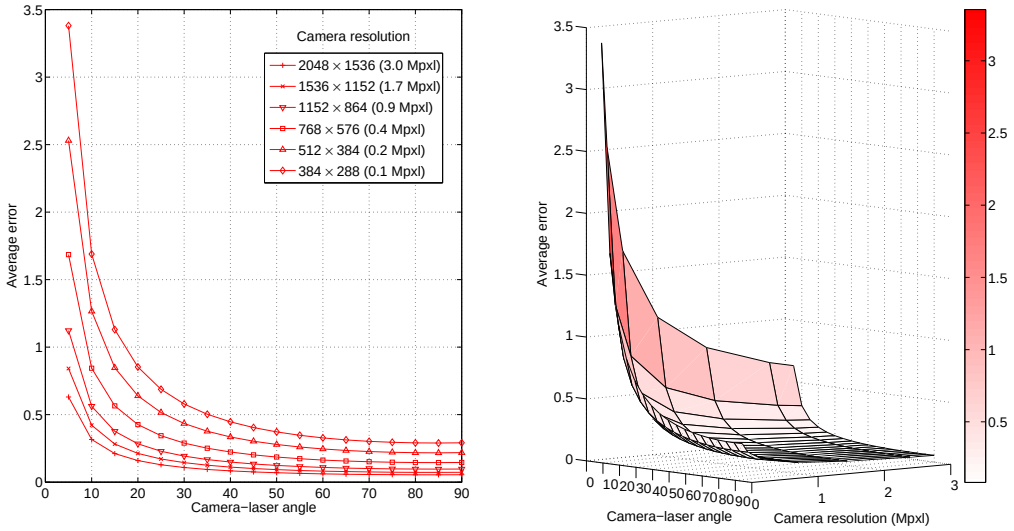


Figure 2.25: Behavior of the resolution error for a camera-laser angle of about 35 degrees. The darker the pixel, the larger the error. At a resolution of 0.4 Mpixels, the resolution error range is 0.22–0.31 mm.

The resolution error was evaluated for different values of camera resolution and camera-laser relative angle, the uncertainty in the (subpixel) estimation of the laser profile being less than $\pm\frac{1}{3}$ pixels. Fig. 2.26 reports the average resolution error obtained with resolutions ranging from 384×288 (0.1 Mpixels) to 2048×1536 (3 Mpixels), and angles between 0 and 90 degrees. The closer to 90 degrees is the camera-laser angle, the smaller is the resolution error. However, as the camera-laser angle increases, the risk of occlusions also increases, being it more probable that the laser profile be hidden from the camera by parts of the object. On the other hand, a singularity condition occurs when the camera-laser angle is zero, i.e., when the camera center lies on the laser plane. For angle values around the singularity—say, less than 20 degrees—the reconstruction error is quite large, whatever the camera resolution. The value of 35 degrees for this angle was found experimentally as a good compromise between accuracy and occlusions. For angles larger than 20 degrees, the error rapidly decreases for increasing resolutions.



(a) (b)

Figure 2.26: Average resolution error (mm) as a function of camera-laser angle (degrees) and camera resolution (Mpixels). (a): Parametric 2D plot. (b): 3D plot.

In particular, at a camera resolution of 0.4 Mpixels and a camera-laser angle of 35 degrees (i.e., the values used in the experiments of the previous Subsections), the expected resolution error is 0.25 mm. *This value is below the usual resolution error value of 0.35 mm reported by most state-of-the-art structured light approaches, employing camera resolutions of 0.8 Mpixels or higher.*

2.6 Conclusions and Future Work

In this chapter a hybrid approach for shape reconstruction and texture sampling suited to desktop applications was presented, combining the high accuracy of structured light methods with the flexibility of passive uncalibrated vision methods. The approach is based on active rectification and collation of laser profiles for shape reconstruction, and the synthesis of a virtual view of the object for texture sampling. Complex and robust computer vision

algorithms were designed so as to make the approach effective also with a simple manual setup, and off-the-shelf hardware. All the phases of 3D model acquisition are carried out by estimating and applying 2D-2D image warping transformations—namely, harmonic homologies, planar homologies, and planar homographies. This avoids estimating and using any intermediate 3D data, with beneficial effects on both model accuracy and speed of operation. Two versions of the approach have been presented and tested, that differ in the way calibration is performed. Experiments with a reference object have proved that both the approaches have comparable results w.r.t. standard acquisition techniques. Experiments with different typologies of real life objects and error analysis have shown that the proposed algorithms are competitive with state-of-the-art 3D acquisition solutions.

Future work will address adding further operational flexibility to the approach. A richer camera model incorporating non linear lens distortions will be introduced so as to deal even with extremely low quality cameras such as webcams. The laser-turntable orthogonality constraint will be removed, by exploiting auxiliary laser traces. This will further simplify the manual setup, and reduce the influence of setup on model accuracy. Finally, the actual turntable rotation between subsequent frames will be explicitly estimated, so as to deal with not constant turntable speeds and/or video acquisition frame rates.

3

Eyemouse

In this chapter a single camera iris tracking and remapping approach based on passive computer vision is presented. Tracking is aimed at obtaining accurate and robust measurements of the iris/pupil position. To this purpose, a robust method for ellipse fitting is used, employing search constraints so as to achieve a better performance with respect to the standard RANSAC algorithm. Tracking also embeds an iris localization algorithm (working as a bootstrap multiple-hypotheses generation step), and a blink detector, that can detect voluntary eye blinks in human-computer interaction applications. On-screen remapping incorporates a head tracking method capable of compensating for small user head movements.

An evaluation method for the choice of the layout of both hardware components and calibration points is described. Experiments also investigate the importance of providing a visual feedback to the user, and the benefits gained from performing head compensation, especially during image-to-screen map calibration.

Gaze estimation systems are aimed at determining the direction of gaze and/or the location pointed at by a user. Such systems play an important role in several scientific and application domains, such as ophthalmology, psychology and neurology, marketing and advertising, human-computer interaction and new generation interfaces, aids to disabled people, etc.

Pupil and iris are the most common eye parts being monitored with gaze estimation techniques. Existing techniques can be categorized according to

degree of intrusiveness, technology employed (e.g., active vs passive), cost, and target application domains.

Intrusive techniques require some equipment to be put in physical contact with the user, e.g., electrodes, contact lenses or head mounted devices [23]. Non intrusive techniques are mostly vision based, i.e., they use cameras to capture images of the eye. Most of the commercial devices adopting non intrusive techniques often rely on the analysis of infrared light generated by an emitter and reflected by the eye: The effect of such a reflection is to enhance the contrast between the pupil and the iris [46]. Such systems are fairly accurate, but also require special and expensive hardware and, being based on *active* light emission, retain a certain degree of intrusiveness. Moreover, sunlight and glasses can seriously disturb the reflective properties of IR light. A common requirement for commercial systems is that the head maintains perfectly still during use—this is typically achieved by means of special supports for the chin. This is another factor that limits the degree of usability of gaze estimation systems. IR-based eye trackers generally use generally the center of eye and the glint (reflection of the IR light on the eye surface): Assuming a static head, the glint acts as a reference point and the vector from the glint to the center describes the gaze direction. Exploiting a neural network-based approach, in [33] a technique based on IR is presented, that reaches the (not so good) accuracy of about 5°, yet it allows head movements.

Other vision-based approaches avoid the use of active illumination, relying exclusively on natural light. Such *passive* approaches typically use off-the-shelf hardware, and monitor eye gaze shifts by performing iris localization and tracking. Indeed, the human iris is a good part of the eye to track under passive vision, due to its perfectly circular shape (giving rise to ellipses under image projection) and its chromatic contrast against the white region surrounding it—the sclera. By the way, the requirement of robustness in uncontrolled conditions of illumination, image quality and iris appearance does not allow, for the purpose of iris detection/localization, the use of stan-

dard techniques employed in biometric identification [42], [21] working under controlled conditions (i.e., unoccluded iris). In [54] a robust iris localization approach is presented, in order to develop a bootstrapping or failure recovery module for an eye tracker. The ellipse describing the iris is fitted by a simulated annealing approach, maximizing a criterion that compares the intensity variation across the ellipse perimeter with a model derived from observations. Pure eye localization (i.e., without a temporal tracking of the estimated gaze shifts) approaches such as this, even when accurate at 99%, are not suitable for some applications—e.g., those where gaze estimates are fed back to the user for human-computer interaction purposes. As a matter of fact, a 1% failure rate means, at a video rate of 25 frames per second, one wrong estimate every four seconds. In [28] an active contour tracker is presented, combining particle filtering with the Expectation-Maximization algorithm. The tracker is complemented with a gaze estimation system based on a projective model of the image-to-gaze direction map. The tracker works at multiple scales, and reaches a good accuracy in the image plane, especially with close-up views of the eye. However, the gaze estimation method proposed requires that the head remains fixed, thus limiting the usefulness of the approach. In [57] the iris contours are modelled as two planar circles and their projections on the retinal plane are estimated. Given some anthropometric knowledge and user distance, gaze determination is obtained from the elliptical shape of the projected iris with a 0.5° error. In [5] and [45] the problem of avoiding the calibration of the image-to-gaze direction map is addressed, and solved in both cases by the use of a stereo camera pair. Besides methods—such as those cited above—exploiting features extracted from the image such as contours and eye corners, other approaches are appearance-based and use all the raw image data as input. For example, in [59] a neural network is fed with 2000 training examples images, and a gaze estimation accuracy of about 1.5° is obtained.

Among the application domains of gaze estimation systems mentioned above, advanced human-computer interaction is one of the most interesting

for both its social and commercial impact. The field of human-computer interaction has largely benefited from the recent advances in computer vision technology. Indeed, several systems have been developed in the recent past, where user body parts such as head, arms, hands, and eyes are tracked in the image, and the results of image analysis are used for the purpose of controlling the interaction space [16]. In this context, gaze estimation systems are among the most difficult to desing, due to the fact that the remapping transformation of image pupil position onto the interaction space—usually, but not exclusively, the screen of a computer—is time-dependent, and changes whenever the user moves his head.

3.1 Overview

A single camera iris tracking and remapping approach based on passive computer vision is presented. The approach aims to obtain accurate measurements of the iris/pupil position for applications in the area of eye-commanded human-machine interaction systems, employed, e.g., to support severely disabled people in their interaction and communication needs. To this purpose, tracking is performed with a RANSAC-like robust method for ellipse fitting that incorporates search constraints so as to increase the overall accuracy with respect to the standard RANSAC algorithm. The tracking procedure also incorporates an iris localization algorithm, with the role of providing first guesses for the iris position either at bootstrap time, or when the tracker has to be re-initialized after a failure. A simple and effective blink detector has also been developed, that can set the tracker to an idle state in the absence of useful measurements (eye closed), and be used to detect voluntary eye blinks related to selection actions in human-computer interaction applications. Tracking is complemented with an on-screen remapping procedure, capable of compensating for small user head movements.

In addition, an evaluation method for the choice of the layout of both hardware components and calibration points is described, which is based on

the statistical propagation of measurement uncertainty.

3.2 Iris Tracking

The overall tracking approach can be described in terms of an automaton composed by two main states: *iris localization* and *iris tracing* (see Fig. 3.1). In the tracing state, the new position of the iris—modelled in the image as an ellipse—is searched for starting from its last estimate. If an ellipse is found in the neighborhood of the previous one, the automaton remains in the tracing state, updates the estimate, and waits for the next frame. If a tracing loss occurs instead, the iris localization state is reached, where several iris candidates are selected, and passed back to the tracing state. Tracing is then re-initialized with the iris candidate that best fits image data. The automaton reaches a third state (*wait*) any time the eye is found to be closed. This can happen either involuntarily (in which case, eye blinks are rather fast) or voluntarily (as connected to the use of the eye as a human-computer interaction device: Eye blinks are in this case akin to mouse button clicks). An eye blink detector based on image analysis serves to control the transitions to and from the wait state.

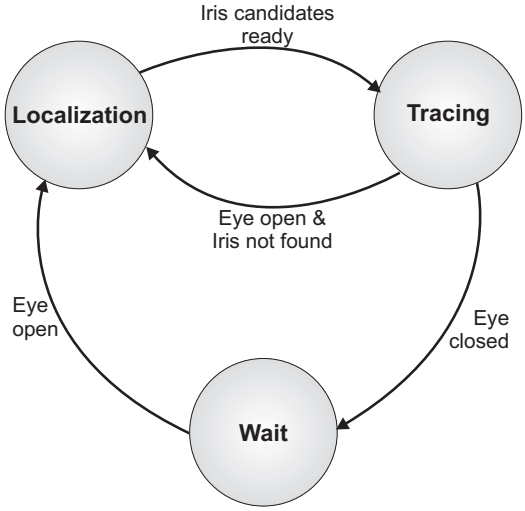


Figure 3.1: The automaton describing the iris tracking algorithm.

3.2.1 Iris Localization

In this state, the current frame—featuring a close-up view of the eye region—is analyzed so as to generate a set of first guesses for the iris location. After linear equalization, the image is filtered using morphological operators (erosion/dilation) so as to emphasize the contrast between the iris and the sclera. The *interest operators* $H_y(x)$ and $H_x(y)$ are then computed along the x and y image axes. They are cumulative intensity histograms based on image intensity $I(x, y)$. In particular, the interest operator along the x axis—the one along the y axis being obtained in a similar way—is defined as

$$H_y(x) = \sum_y \sum_{u=-\nu}^{\nu} \delta(x + u, y) \ , \tag{3.1}$$

where $\delta(x, y) : \mathbb{R}^2 \mapsto \{0, 1\}$ is a Boolean function such that

$$\delta(x, y) = 1 \ \text{iff} \ I(x, y) < \tau \ .$$

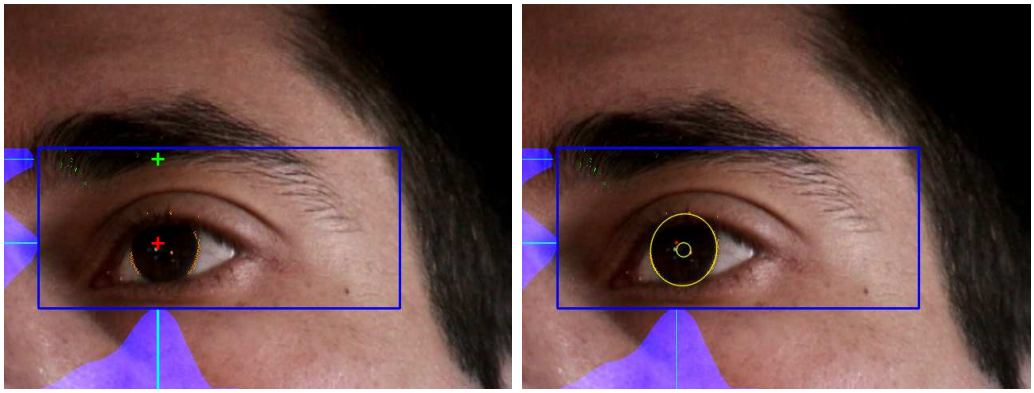
The parameters ν and τ control respectively the smoothness and selectivity of the operator. The interest operators define the two sets

$$X = \{x_i | x_i \ \text{is a local maximum of} \ H_y(x)\}$$

and

$$Y = \{y_j | y_j \ \text{is a local maximum of} \ H_x(y)\} \ ,$$

whose Cartesian product $X \times Y = \{(x_i, y_j) | x_i \in X, \ y_j \in Y\}$ forms the set of putative locations of the iris center. Fig. 3.2(a) shows the interest window inside which the operators are computed. In the figure, the x operator has one maximum corresponding to the iris position, while the y operator has two local maxima, corresponding respectively to the eyebrow and iris. In this case, two iris hypotheses are issued, one of which is correct. The figure also shows the edge points extracted from the image, which are used during iris tracing operation.



(a) (b)

Figure 3.2: (a): The x and y interest operators, and the two iris location hypotheses (indicated by crosses) associated to their local maxima. (b): The estimated iris position (indicated by an ellipse) after multiple hypotheses testing during iris tracing.

3.2.2 Iris Tracing

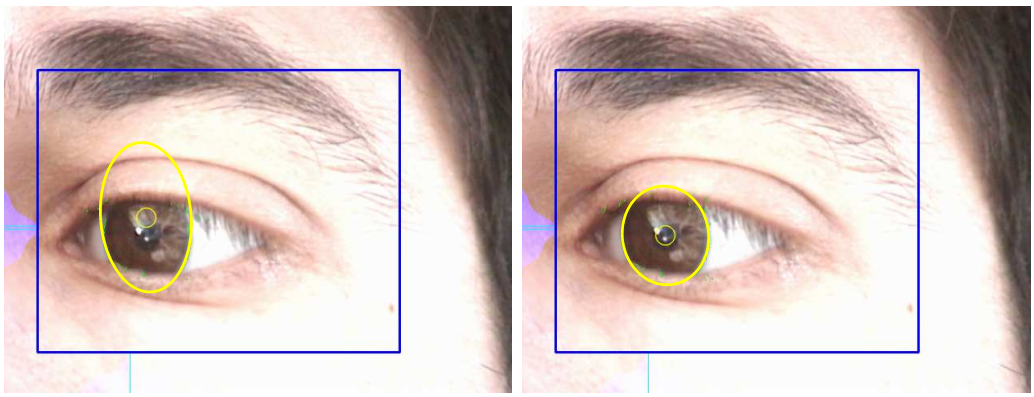
The iris tracing module searches for the new iris position starting from the last estimated iris position. Iris search is performed as follows. First, the last estimated ellipse is uniformly sampled into N points. At each sample point, edge points are searched for along a horizontal line using a multi-scale ternary search [20], and the location of the strongest edge is recorded. Once the set of all N strongest edge points is formed, a robust ellipse fitting algorithm is run, aimed at finding the new image location and shape of the iris. The fitting algorithm is a modification of the RANSAC approach [29]. Ellipse fitting with standard RANSAC requires (1) selecting five points at random inside the edge set, (2) estimating the (unique) ellipse through these points, (3) computing the inliers with respect to the ellipse found—i.e., the edge points whose distance from the ellipse is below a given threshold. Steps (1) through (3) are repeated several times (typically, for a fixed number of iterations, so as to ensure real-time performance), after which the ellipse with the highest number of inliers (or *consensus*) is chosen as best estimate, and its parameters are further refined through a least square fitting involving all

the inliers found. In order to check if the inliers related to the RANSAC solution possess really an elliptical arrangement—hence, the solution can be considered acceptable as the new ellipse instance—, a probabilistic validation is introduced, similar to the one used in [8] for validating homography estimates. Specifically, a verification model is used to compare the probabilities that the set of inliers/outliers was generated by a correct or false ellipse in the image. Denoted with N_i the number of inliers found among the N edge points, the RANSAC solution is considered to be acceptable with a probability higher than 97% if $N_i/N > 0.25$. If this condition is not met, a tracing failure occurs, and our "constrained RANSAC" tracing algorithm (from now on, C-RANSAC) is run again using as starting ellipse the circles of radius R centered at the iris candidate locations generated by the iris localization algorithm. Fig. 3.2(b) shows the recovery from a tracing failure: Notice how the new estimated iris corresponds to one of the two hypotheses of Fig. 3.2(a). Notice also that all the edge points estimated starting from the wrong candidate (in fact, the eyebrow) do not provide an acceptable solution, as they are not arranged in an elliptical way.

Standard RANSAC is known to be robust with respect to outliers (in our case, outliers are all edges not belonging to the border between the iris and the sclera). However, for the task at hand an extension to the standard method has been specifically developed in order to improve tracking performance: The idea is to embed into RANSAC further knowledge about the tracing task. More explicitly, such knowledge concerns the range of possible ellipse dimensions. Constraints on ellipse size can be expressed as

$$\alpha R > a \geq b > \beta R \ , \tag{3.2}$$

where a and b are respectively the major and minor ellipse axes, $\alpha = 1.1$ and $\beta = 0.7$ (these values were chosen after extensive experimentation), and R is a reference iris radius that depends on the camera zooming level, and can be chosen according to a rough knowledge of eye appearance. Embedding such constraints into RANSAC is achieved by discarding, after step (2), all ellipse instances not satisfying Eq. 3.2. Fig. 3.3 shows a frame where standard



(a) (b)

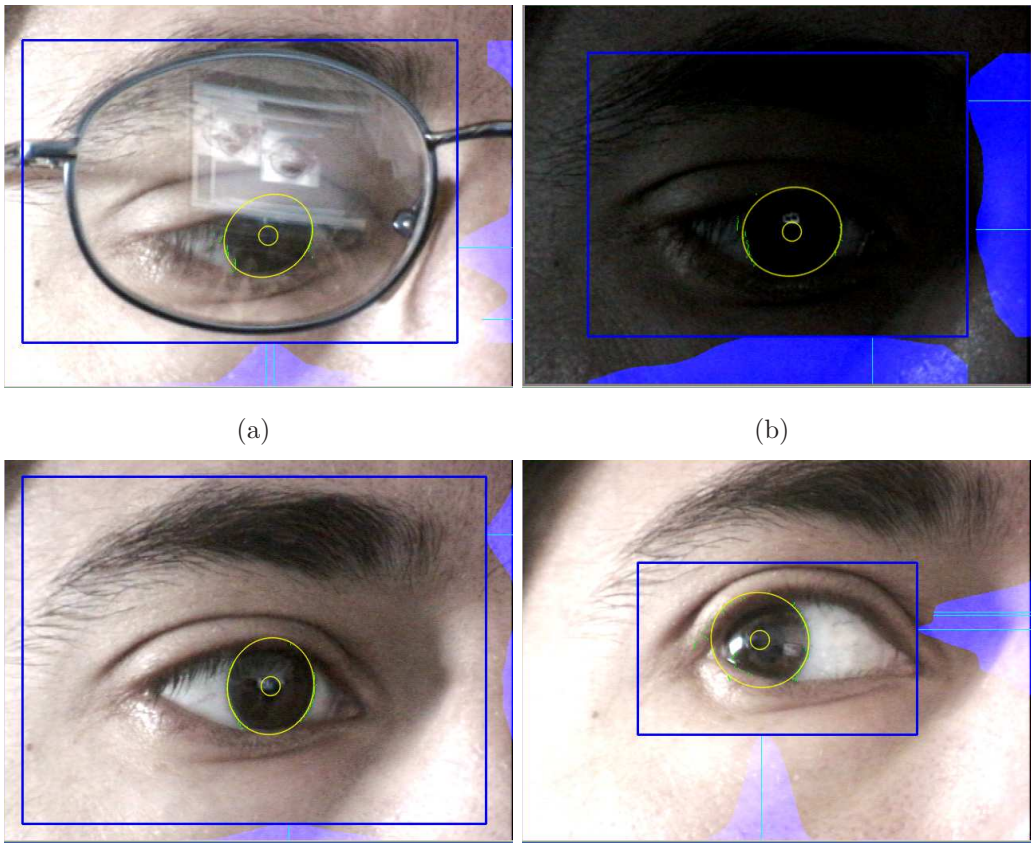
Figure 3.3: A typical failure of standard RANSAC (a), and the corresponding result with C-RANSAC (b).

RANSAC fails, while C-RANSAC provides a correct solution.

Fig. 3.4 shows qualitative tracking results with C-RANSAC in four critical conditions of use: (a) user wearing eyeglasses, (b) dim environmental light, (c) presence of the eyebrow—acting as a distractor—in the interest window, (d) lateral eye gaze. In all cases, the fitting performance appears to be quite good, even if in case (d) an ellipse slightly bigger than the actual one is estimated.

3.2.3 Eye Blink Detection

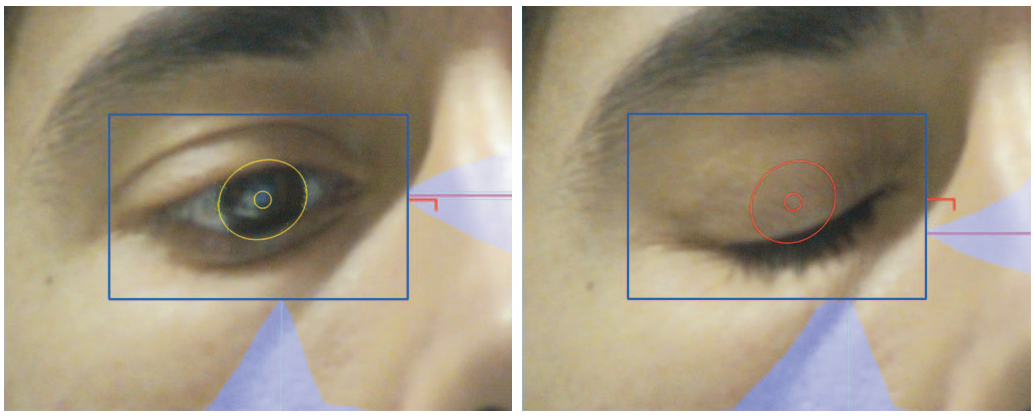
The eye blink detection algorithm controls the transitions to and from the wait state, thus disabling/enabling the tracking algorithm. It exploits a simple heuristics based on the behavior of the interest operator along the y axis: During eye blinks, the peak of $H_x(y)$ corresponding to the current instance of the iris center undergoes a dramatic downward drift. Let us consider the difference Δ_y between the y coordinate of the iris center and the local maximum $\hat{y}$ of $H_x(y)$ nearest to it. When the eye is open this difference is small, since $\hat{y}$ likely corresponds to the new position for the iris center, as shown in Fig. 3.5(a). When the eye is closed instead, $\hat{y}$ is not due to the dark region inside the iris, but is mainly due to eyelashes (see Fig. 3.5(b)). To



(a) (b)

(c) (d)

Figure 3.4: Ellipse estimation in critical operation conditions. (a): eyeglasses. (b): dim light. (c): eyebrow in the interest window. (d): lateral eye.



(a) (b)

Figure 3.5: Eye closure detection based on downward shifts of the vertical cumulative histogram. (a) No shift: eye open. (b) Shift: eye closed.

discriminate between these two situations, a threshold $\gamma = \frac{1}{16}R$ proportional to the reference iris radius is introduced, thus taking into account possible different zoom levels. When the difference Δ_y is greater than γ, the eye is considered closed.

3.3 Remapping

The mapping between the iris center position in the image, $\mathbf{x}$, and the screen location $\mathbf{p}$ pointed at, is modelled as a planar homography. This model extends to full perspective the one proposed in [14], where an affine camera model was used.

Let us assume for the moment that the head is fixed. Then, the (non linear) remapping transformation can be written as $\left[\mathbf{p}^\top \ 1\right]^\top \sim \mathrm{M}\left[\mathbf{x}^\top \ 1\right]^\top$, where M is a constant 3×3 homography matrix (eight degrees of freedom), and $\sim$ denotes equality up to a scaling factor (depending on $\mathbf{x}$). Assume now that the head begins to move, while the eye maintains fixation on the same screen point $\mathbf{p}$. Then, the mapping becomes

$$\begin{bmatrix} \mathbf{p} \\ 1 \end{bmatrix} \sim \mathrm{M}_t \begin{bmatrix} \mathbf{x}_t \\ 1 \end{bmatrix} \ , \tag{3.3}$$

where M_t and $\mathbf{x}_t$ are both time-dependent. In particular, the pupil shift in the image

$$\mathbf{x}_t - \mathbf{x}_0 = \Delta^h \mathbf{x}_t + \Delta^e \mathbf{x}_t \tag{3.4}$$

related to a movement of the head, is the composition of two shifts: $\Delta^h \mathbf{x}_t$, due to the eyeball center movement as part of the head, and $\Delta^e \mathbf{x}_t$, due to the eyeball rotation around its center required to maintain fixation on $\mathbf{p}$. In order to use Eq. 3.3 to remap the current iris center onto the screen, the map M_t must be estimated on a frame by frame basis. Assuming a multiplicative update law for the map, i.e.,

$$\mathrm{M}_t = \mathrm{M}_0 \mathrm{U}_t \ , \tag{3.5}$$

the image-to-screen map estimation process can be split into two distinct phases: (*i*) *calibration* of M_0 executed once at startup ($0 \leq t \leq T_c$), (*ii*) estimation of the map update matrix U_t at run time ($t > T_c$). Calibrating the map M_0 does not mean, of course, having to calibrate the camera, i.e., estimating the camera internal parameters: From this point of view, the computer vision approach developed in this work is an uncalibrated one. Indeed, the matrix M_0 can be estimated in a robust way by first collecting a set of image measurements $\{\mathbf{x}^i\}_{i=1}^k$, $k \geq 4$, and then using Eq. 3.3 as a model for the RANSAC algorithm. Image measurements must be related to at least four non collinear points on the screen, referred to as "calibration points." As the head can move also during the calibration phase, estimating head motions in the image and compensating for them is required at any time $t > 0$.

3.3.1 Compensation for Head Motion

To estimate the effects of rigid head motions in the image, the Lucas & Kanade feature tracker [41] is applied to the two image regions that are respectively above and below the interest window including the eye. Such regions include portions of the face that approximately have the same orientation of the eye plane, and are thus suitable for measuring the image motion

patterns related to 3D movements of this plane. The image region inside the interest window is excluded, as it includes the eyeballs, whose rotations are actually independent of rigid head movements. The assumption here is that the user does not change expression during eye tracking: A change of expression would introduce a non rigid term in the image motion pattern, that is also independent of head movements.

Image head motion is modelled by the affine transformation A_t (six degrees of freedom) such that

$$\mathrm{A}_t \begin{bmatrix} \mathbf{x}_0 + \Delta^h \mathbf{x}_t \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{x}_0 \\ 1 \end{bmatrix} . \tag{3.6}$$

Since for all imaged head points the term $\Delta^e \mathbf{x}_t$ in Eq. 3.4 vanishes identically, the matrix A_t can be estimated in a robust way by collecting a set of image measurements $\{\mathbf{x}_t^i - \mathbf{x}_0^i = \Delta^h \mathbf{x}_t^i\}_{i=1}^n$, $n \geq 3$, and using Eq. 3.6 as a model for the RANSAC algorithm.

Once estimated, A_t is exploited, as said above, both during calibration and point remapping. At calibration time, A_t is used to transform the current image measurement of the iris center $\mathbf{x}_t$ into its corresponding reference location $\mathbf{x}_0$. This is done considering that, for small head movements, the term $\Delta^e \mathbf{x}_t$ in Eq. 3.4 can be neglected also for pupil shifts, thus allowing Eq. 3.6 to be applied. At remapping time, updating the image-to-screen map is simply done by using A_t in the place of U_t in Eq. 3.5. Indeed, under the "small head movements" hypothesis above, it is easy to show, by combining Eqs. 3.3 through 3.6, that the map update matrix and the affine motion matrix are actually coincident.

3.4 Experimental Results

Several experiments were performed to assess the performance of the software modules implementing the algorithms developed for this work. The hardware employed for the experiments involved a digital camera with a 12× optical zoom and an image resolution of 640 × 480 pixel, a 19" computer screen

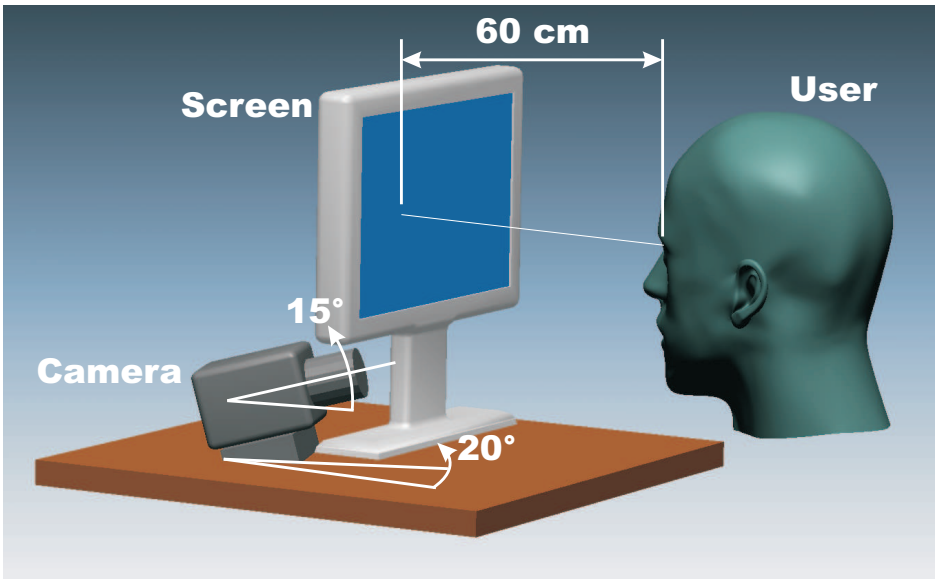


Figure 3.6: The experimental layout.

working at a resolution of 1024×768 pixel, and a standard laptop with a 1.73 GHz processor. Several users alternated during experiments: Performance figures will be provided for the average user. For the experiments, a general, non symmetric layout of user, camera and screen was chosen (see Fig. 3.6). The users sits in front of the screen at a distance of approximately 60 cm from it. The camera is placed on the lower left side with respect to the user: The optical axis is at an angle of about 15° with respect to the ground plane, and 20° with respect to the normal to the screen plane. In all experiments, the constant ν in Eq. 3.1 is set to $R = 60$ pixel. The threshold τ is set to a fixed value of 32, empirically chosen after several tests done under different lighting conditions. To ensure real-time performance (25 frames per second), 260 tracing RANSAC iterations per frame are run.

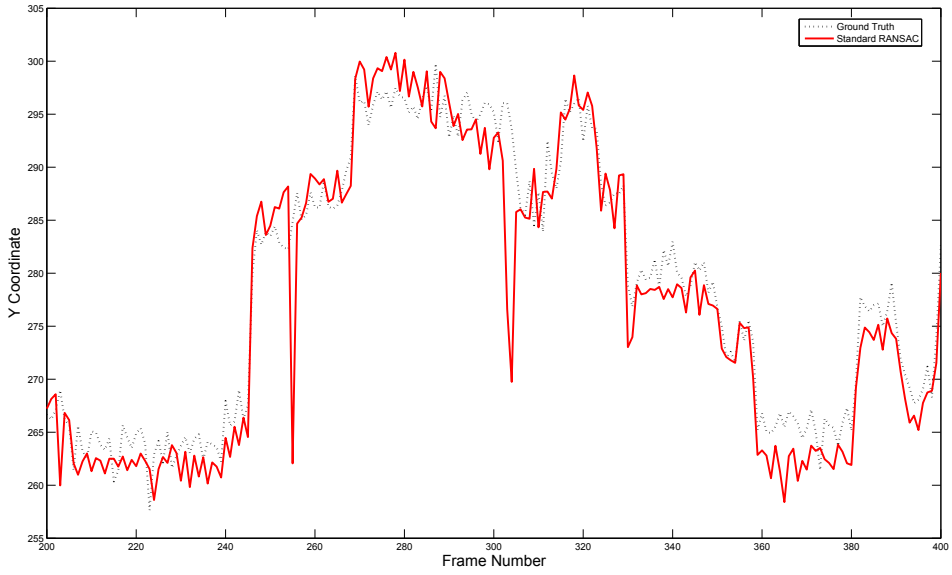
3.4.1 Tracking

In order to test the tracking behavior, a real image sequence of 594 frames (for a total duration of 23.7 seconds) with several eye movements was recorded: The iris position and shape in each frame of this test sequence was manually

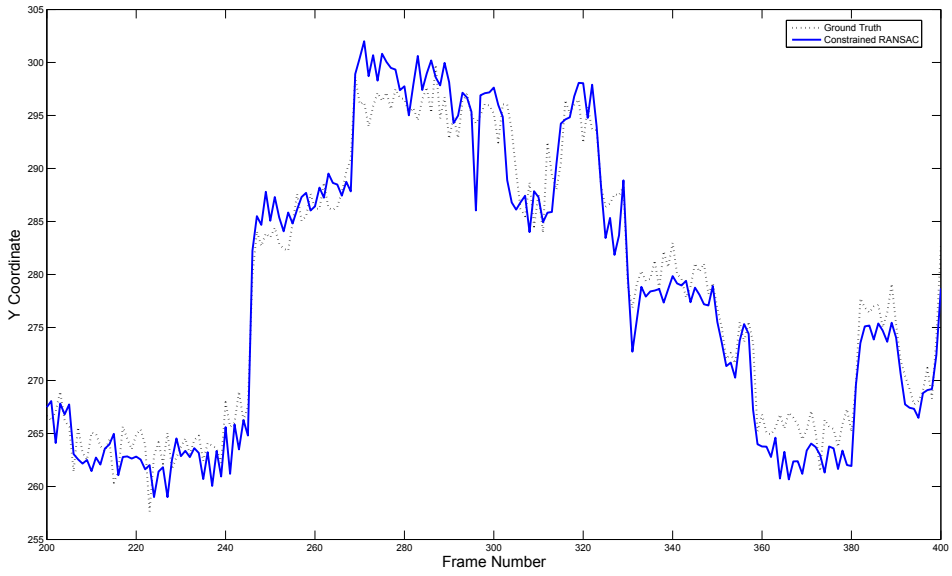
annotated, and compared against the automatic tracking results. Fig. 3.7 shows tracking accuracy with both RANSAC and C-RANSAC algorithms for the central portion of the test sequence, in which both smooth pursuit and saccadic eye movements take place. The y (i.e., vertical) coordinates of the estimated vs ground truth ellipse centers are shown. From a comparison of the two diagrams it appears that the tracking behavior of RANSAC is affected by two "spikes" (occurring at frames #255 and #304) that significantly decrease performance with respect to C-RANSAC. Spikes arise from partial iris occlusions due to the eyelids: Such occlusions reduce the number of detectable points for the upper and lower parts of the iris, and consequently make it more difficult for the fitting algorithm to estimate an ellipse with the correct major/minor ellipse axis ratio—refer again to Fig. 3.3, showing what happens at frame #255. Fig. 3.8 presents a comparison of tracking errors for the last portion of the test sequence, involving head movements. Errors are reported for both the x and y components of the ellipse centers. Also in this case, C-RANSAC performs better than RANSAC. However, the x errors are quite similar, due to the smaller influence of eyelid occlusions on the estimation of the horizontal components of the ellipse centers.

In Fig. 3.9, the distribution of the tracking errors along the y axis for the whole test sequence is reported for both the standard and constrained algorithms. The diagrams shows that the latter approach has a less dispersed error than the former, which exhibits instead relevant tails in the distribution, and a nonzero average.

To gain a further insight on fitting behavior, two different Monte Carlo simulations were also carried out, using as reference the ground truth ellipse for frame #99. This ellipse is parameterized as $\mathbf{x}(\vartheta) = (a\cos(\vartheta) + x_0, b\sin(\vartheta) + y_0)$, where (x_0, y_0) is the center and a, b are the lengths of the two semi-axes. In the first simulation, the two point sets $\{\mathbf{x}(\vartheta)|\vartheta \in [-\pi/5, \pi/5]\}$ and $\{\mathbf{x}(\vartheta)|\vartheta \in [4/5\pi, 6/5\pi]\}$, corresponding respectively to the left and right visible portions of the iris, were corrupted with additive white Gaussian noise $\mathcal{N}(0, \sigma^2)$, $\sigma \in [0.1, 1.0]$. For each noise level, 1000 ellipse fitting tests were

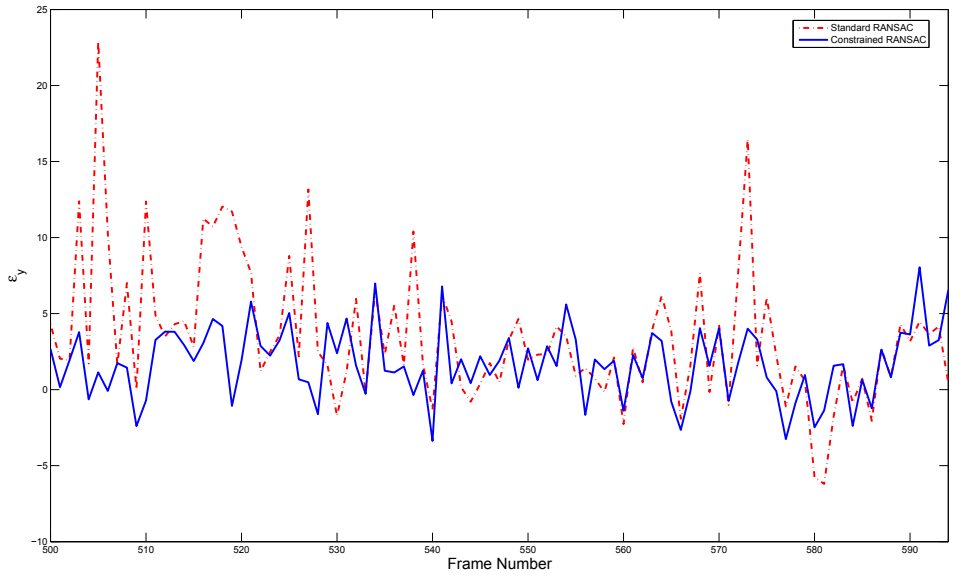


(a)

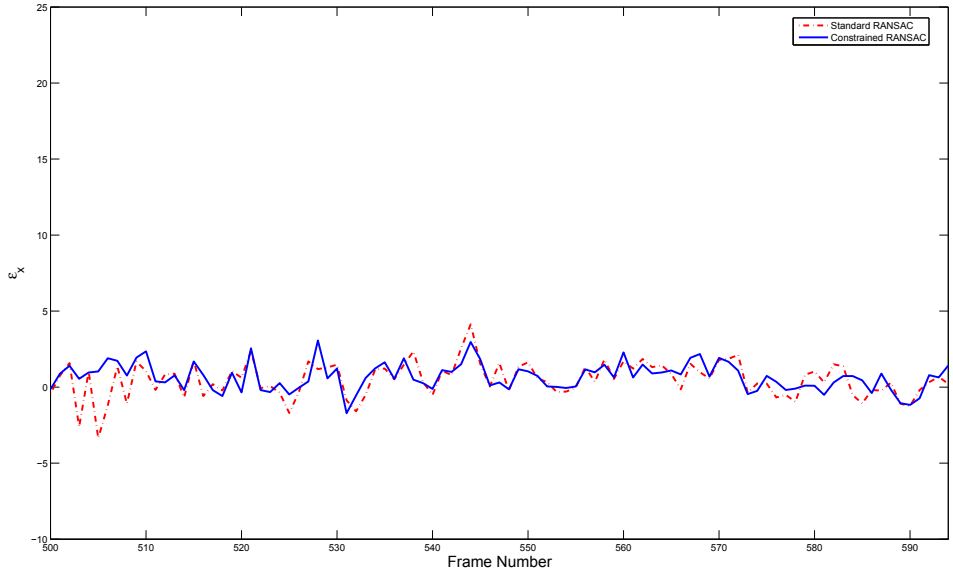


(b)

Figure 3.7: Tracking accuracy (frames 200–400): y coordinate of the ellipse center. (a): RANSAC. (b): C-RANSAC. The dotted line is the ground truth.

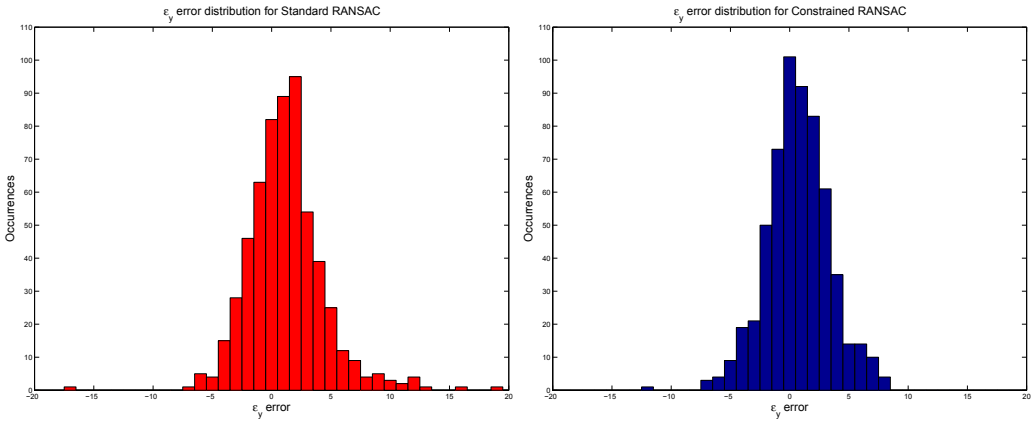


(a)



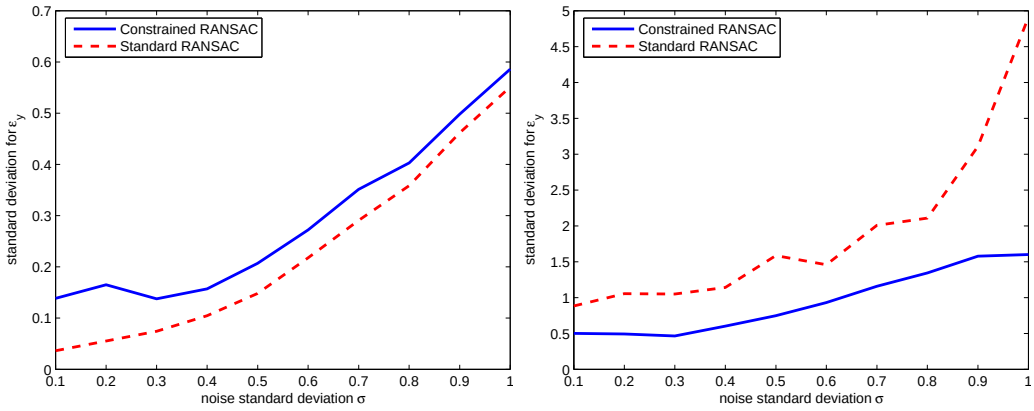
(b)

Figure 3.8: Comparison of tracking errors (frames 500-594): RANSAC (dashed) vs C-RANSAC (solid). (a): y coordinate of the ellipse center. (b): x coordinate of the ellipse center.



(a) (b)

Figure 3.9: Comparison of tracking error distributions: y coordinate of the ellipse center (frames 1-594). (a): RANSAC. (b): C-RANSAC.



(a) (b)

Figure 3.10: Comparison of tracking error distributions: y coordinate of the ellipse center (frame #99), RANSAC (dashed) vs C-RANSAC (solid). (a): Gaussian noise only. (b): Gaussian noise plus distractors.

run. In Fig. 3.10(a), the standard deviation of the y error is shown: In the presence of pure Gaussian noise, the performance of RANSAC is slightly better than C-RANSAC. Fig. 3.10(b) reports instead the results for the second simulation, aimed at testing the fitting performance in the presence of non Gaussian distractors—e.g., spurious edge points due to the eyelids or eyelashes, light reflections and so on—, whose effects were simulated by adding extra random points to the noisy ellipse data of the previous test. In this case, C-RANSAC clearly outperforms standard RANSAC. Moreover, from a comparison of Figs. 3.10(a) and (b) it emerges that, at high noise values, the performance of RANSAC gets worse by about 8.0 times, while that of C-RANSAC decreases only by 2.5. The beneficial effects of the C-RANSAC strategy on tracking robustness are due to the fact that the solution of the fitting problem is taken from a limited space, thus guaranteeing a graceful performance degradation even in the presence of highly perturbed data, such as those occurring in practice. The price to pay for this, is a slight performance drop in the ideal case of pure Gaussian perturbations.

3.4.2 Calibration

The influence of image-to-screen map calibration on remapping accuracy is addressed here. To calibrate, the user is required to sequentially point at eight points on the screen: 100 measurements of the iris center are collected for each calibration point, gazed at by the user for four seconds. Calibration points are so arranged: Four at the corners of the screen, and four (with a rhomboidal layout) at its center.

Fig. 3.11(a) shows iris center measurements in the image plane for both the cases of compensated and uncompensated head movements. Although the user was required to maintain his head fixed, an involuntary head drift is clearly evidenced by the uncompensated image clusters. The beneficial effects of the head compensation algorithm are also clear: *The compensated clusters are more compact and arranged in a more reasonable way with respect to the uncompensated ones.* As a matter of fact, using the uncompensated data

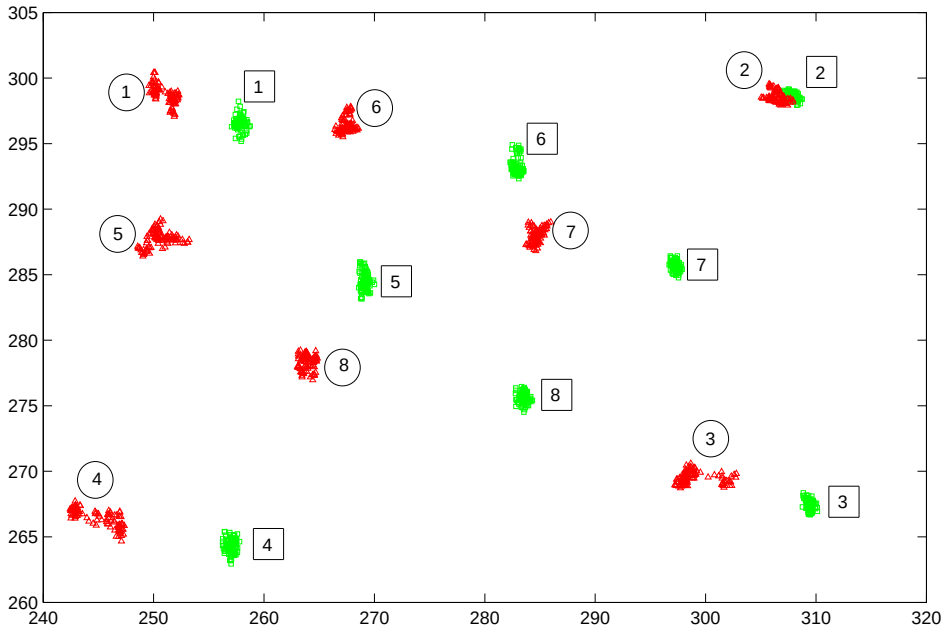
for calibration would produce a grossly incorrect remapping law. The screen point clusters obtained by remapping image measurements onto the screen using the calibrated map are shown in Fig. 3.11(b) together with ellipses summarizing their 2nd-order spatial distribution ("uncertainty ellipses").

The calibrated map obtained as described above was used together with the screen/image point pairs (the screen points pointed at are assumed to be the centers of the calibration circles) to infer the value of remapping uncertainty at any screen point. To this aim, the 2×2 covariance matrix Λ_p at any point $\mathbf{p}$ in the screen is evaluated following the approach in [19]. Given the covariance matrix, the uncertainty at $\mathbf{p}$ can be expressed in terms of the elliptical confidence region

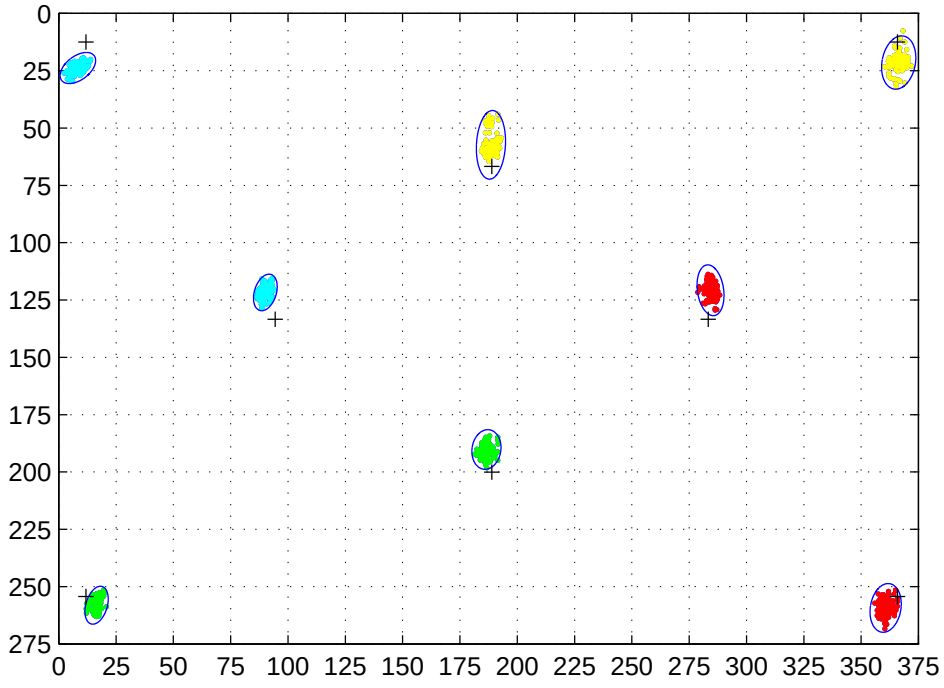
$$(\mathbf{q}-\mathbf{p})^\top \Lambda_p^{-1} (\mathbf{q}-\mathbf{p}) \leq \chi^2 \ , \tag{3.7}$$

i.e., the locus of points $\mathbf{q}$ that with a given probability will be remapped to when the true screen point is $\mathbf{p}$. The parameter χ^2 controls the size of the confidence region based on the chosen probability value. Under the hypotheses of normal distribution for $\mathbf{p}$, $\chi^2 = 5.99$ yields an ellipse large enough to contain the true value of $\mathbf{p}$ with a probability of 95%. Remapping uncertainty propagation was investigated under three different choices of number and location of calibration points.

Fig. 3.12 reports, at all screen points, the value of the major semi-axis of the confidence ellipse for (a) only the four screen corner points, (b) only the four central points, (c) all of the eight points. In the first case, uncertainty is maximum at the top-right screen corner, and minimum at bottom-left. Also, uncertainty is higher at top-left than at bottom-right. This pattern can be explained by recalling that, during the test, the camera was placed to the lower left side with respect to the user, sitting exactly in front of the screen. Therefore, the iris shift in the image during a gaze shift around the bottom-left screen corner is the largest possible, and hence the most accurate to measure. Concerning the case (b), the uncertainty pattern is radially symmetric with respect to a point located at the middle-bottom of the screen: This is also easily explained by the fact that the iris shifts occurring around

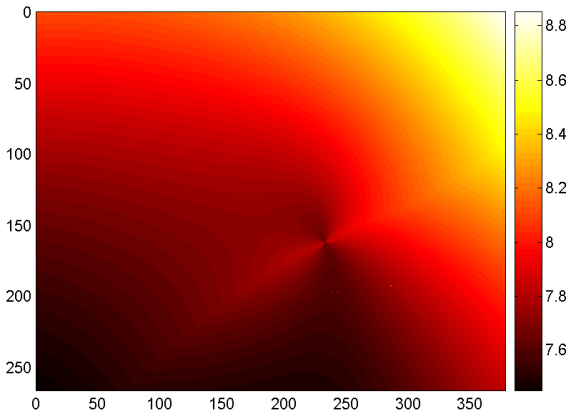


(a)

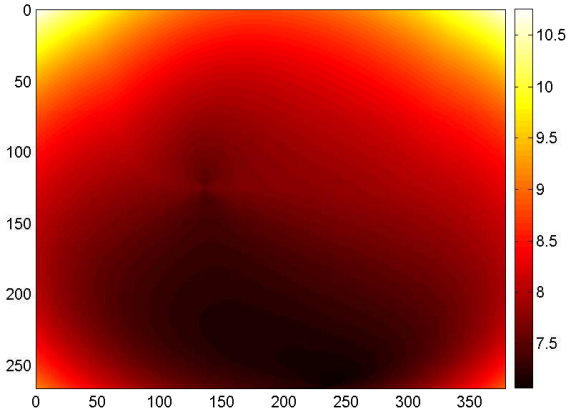


(b)

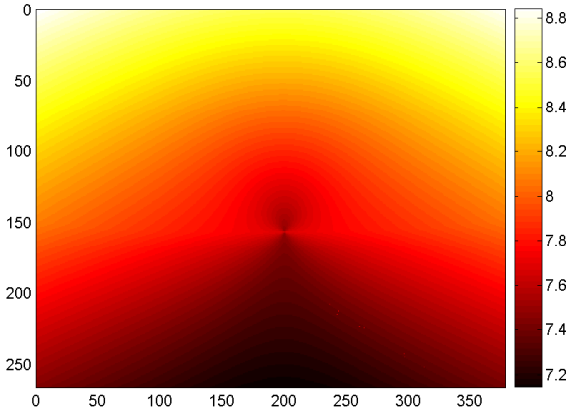
Figure 3.11: (a): Effect of the head compensation algorithm (image plane). The clusters labelled with squares correspond to compensated measurements; circular labels denote uncompensated measurements. (b): The compensated clusters of (a) as remapped onto the screen after calibration. Crosses indicate the centers of the calibration circles. For each cluster, the corresponding "uncertainty ellipse" is drawn (see text).



(a)



(b)



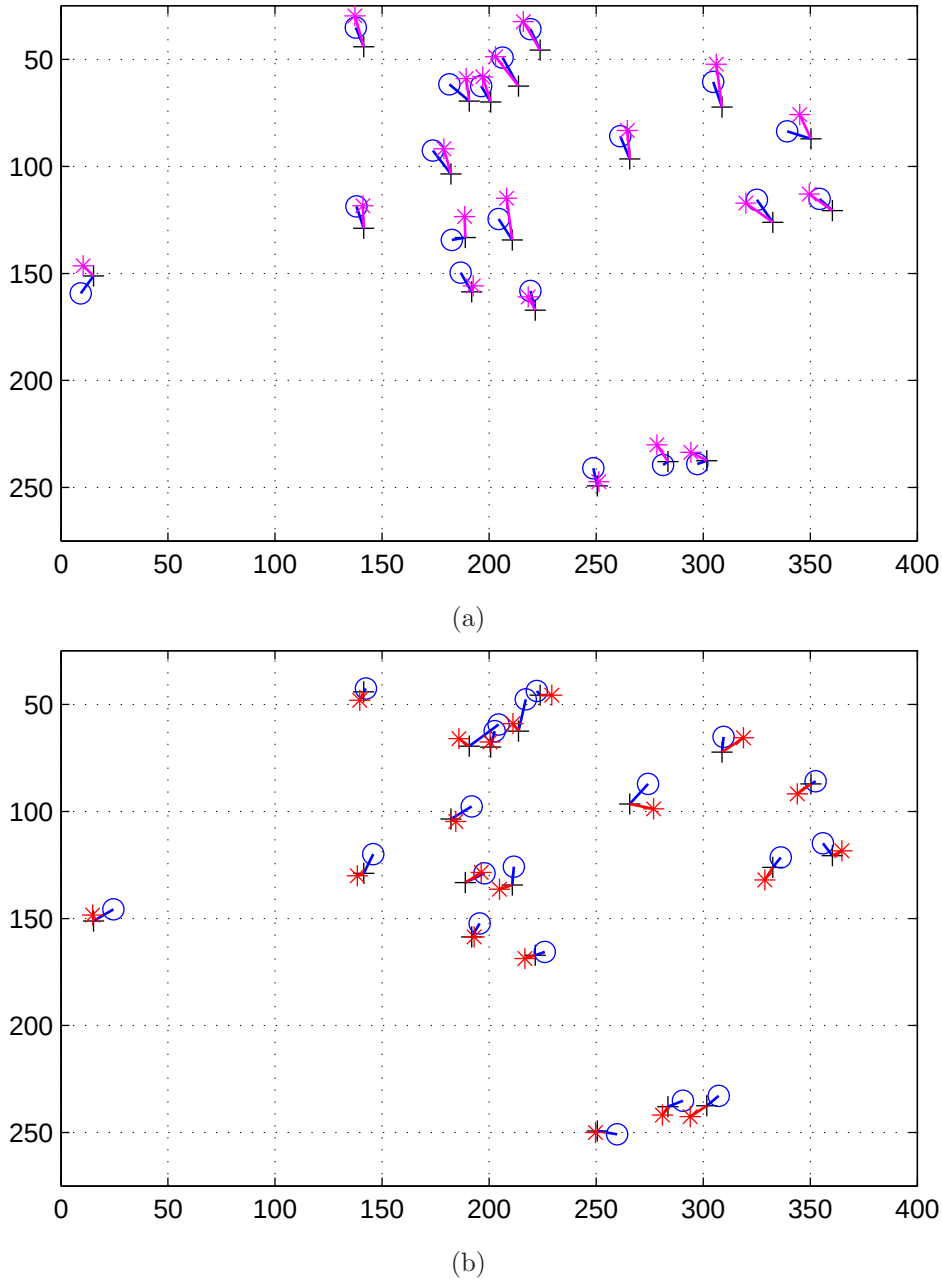
(c)

Figure 3.12: Pointwise remapping confidence, using for calibration (a) the four corner points only, (b) the four central points only, (c) all of the eight points. All values are expressed in mm. Intensity is inversely proportional to confidence, hence directly proportional to uncertainty.

the central calibration points are almost equal in size. Notice also that, since in this case calibration points are closer to each other than in the previous case, the corresponding gaze shifts are also smaller, thus increasing the value of the maximum uncertainty. Finally, in case (c) where all of the calibration points are taken into account, the uncertainty pattern is somewhat a mixture of the two patterns above, and exhibits an overall axial symmetry with respect to the vertical middle line of the screen, with uncertainty values increasing from bottom to top. Several calibration tests were carried out, always providing comparable results. The value for the major semi-axis of the confidence ellipses was ranges from 7.1 to 8.8 mm for the performed tests, while the minor semi-axis ranges from 5.4 to 8.0 mm. Experiments of the kind described above can be used *to evaluate the best number and location of calibration points as a function of the expected accuracy, and of the chosen camera-user-screen layout.*

3.4.3 Remapping

In this section, experiments on eye gaze remapping are described and discussed. For all tests, the image-to-screen map was calibrated using the eight point pattern described in the previous section. Head compensation was performed during calibration. Two experiments were done to gain an insight into the remapping accuracy while pointing at a sequence of twenty screen points generated at random, and presented to the user one at a time. The error between the true screen points and the estimated (remapped) ones was recorded, and an overall accuracy measure was computed in terms of average and standard deviation of the error. In the first experiment, devoted to assessing the temporal degradation of remapping accuracy, no feedback was provided to the user about the remapped screen position. The random sequence was shown to the user twice: The first time soon after calibration, and the second time ten minutes later. Results are shown in Fig. 3.13(a). The average error for the first sequence is 3.2 mm, with a standard deviation of 0.9 mm. In the second sequence, the average error is 4.0 mm, with



(a)

(b)

Figure 3.13: Remapping errors while pointing at a sequence of twenty random screen points (denoted by crosses). Coordinates are in mm. (a): Fixation without feedback. Results obtained soon after calibration (circles) and ten minutes later (stars). (b): Fixation with feedback. With head compensation (circles) and without head compensation (stars).

a standard deviation of 1.8 mm. Recalling that the distance between user and screen was 60 cm, the average angular gaze estimation errors are 0.31° and 0.38° respectively. These results are comparable with those obtained with commercial systems employing active technologies. The second experiment aimed at investigating the remapping accuracy in the presence of a visual feedback of user action. The user had eye control of the mouse icon, and was asked to place it on each of the random screen points by suitable eye movements. Results are shown in Fig. 3.13(b), where circles and stars refer respectively to measurements with and without head compensation. In the first case, the average error was 2.6 mm (corresponding to 0.25°), with standard deviation 1.3 mm. In the second case the average is 1.7 mm (0.16°), with deviation 0.9 mm. Results show that performance improves in the presence of visual feedback. Another important conclusion is that *in the presence of visual feedback, it is better* not *to compensate for head motion*: Indeed, remapping performance is even better without head compensation. This is explained by the fact that, if provided with feedback about his action, the user himself unconsciously compensates for remapping errors with slight head movements. (However, users with particular disabilities may not be able to control the head properly: For such users, head compensation is mandatory to ensure an acceptable performance.) Fig. 3.14 illustrates this point even better. The figure shows a 2D sketch made by the user by exploiting visual feedback without head compensation. The user was asked to reproduce, using the eye-controlled mouse icon, a line drawing displayed on the screen. The task was successfully performed through the execution of smooth pursuit eye movements.

3.4.4 An Interaction Scenario

To conclude the experimental section, a realistic human-computer interaction application using the proposed framework is described. The reference scenario is that of a severely disabled person, whose only residual motor capabilities are those of the eyeballs. The user's task is to control a conventional

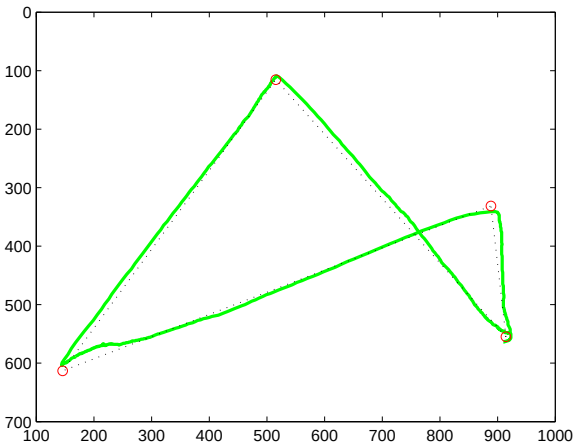


Figure 3.14: Exploitation of smooth pursuit eye movements for on-screen sketching. The ground truth drawing is shown with a dotted line.

"Window, Icon, Menu, Pointer" interface with ocular control. I.e., eye shifts and blinks are used respectively for controlling pointer icon dragging and mouse button clicking. Eye blink detection has a success rate of about 100% with trained users, who perform eye blinking by gently closing the eye—i.e., as when falling asleep. Success performance degrades to about 80% when untrained users wink: In this case, the histogram-based eye blink recognition method can be distracted by face wrinkles, eyebrows entering the interest window, etc. Fig. 3.15 illustrates the task of reaching (top and middle) and selecting (bottom) a thumbnail on a computer screen. Since the remapping uncertainty of the approach is lower than the size of typical menu icons at a standard resolution, users can successfully control an interface of this kind.

3.5 Conclusions and Future Work

In this chapter a robust, single camera, real-time eye tracking algorithm was presented, also embedding an eye blink detector working equally well for both voluntary and involuntary eye closures. A constrained RANSAC approach for iris tracking was proposed, that performs better than standard RANSAC in the presence of distractors and occlusions in the image sequence. The

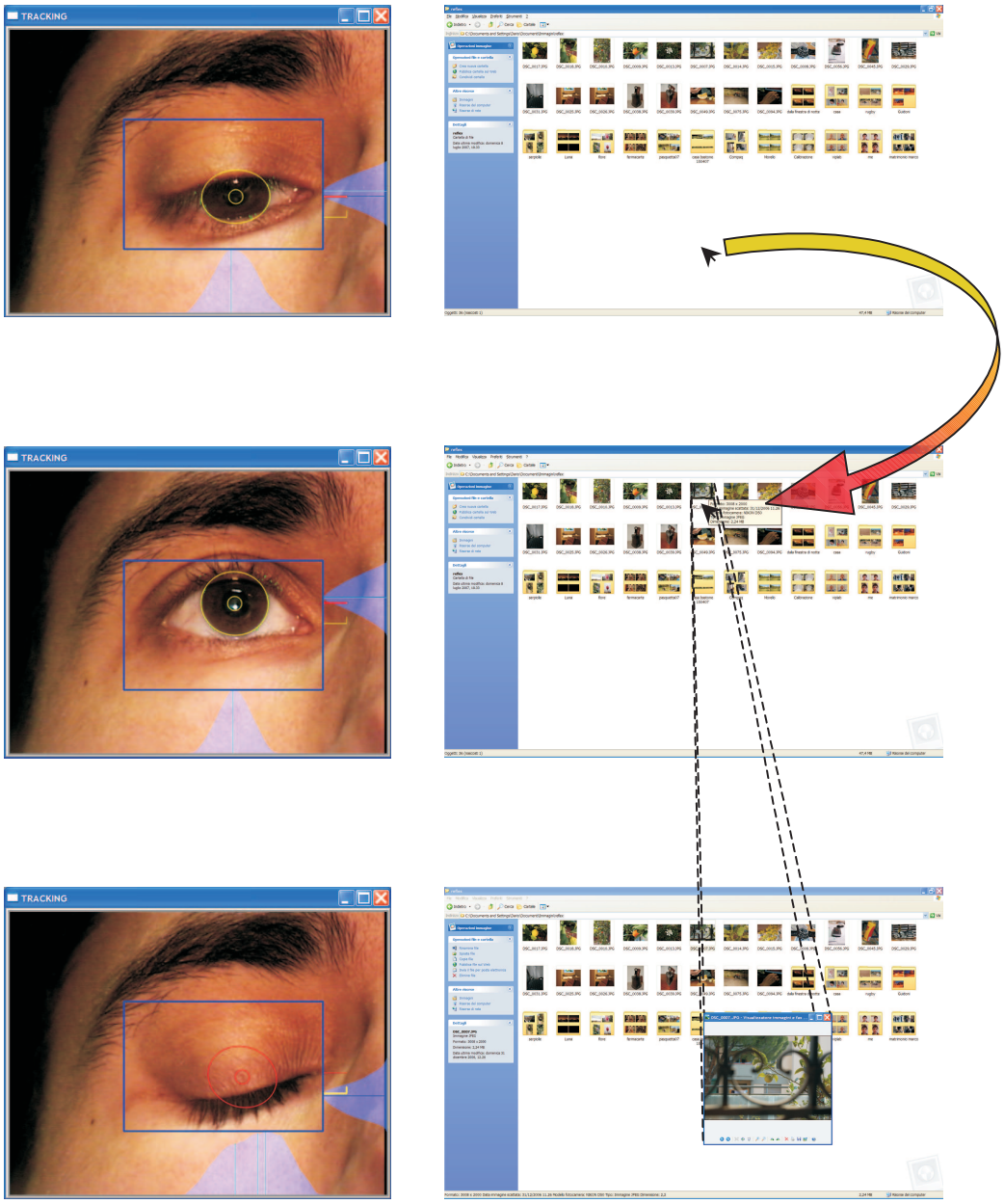


Figure 3.15: A real application scenario: ocular control of an interface. (top and middle): mouse icon placement through a gaze shift. (bottom): item selection after an eye blink.

approach is complemented with an on-screen remapping method, capable of compensating for small head movements. An extensive experimental section was provided, illustrating the characteristics and limitations of the various components of the framework. In particular, an evaluation method for the choice of the layout of both hardware components and calibration points was presented, which is based on the statistical propagation of measurement uncertainty. Experiments also outlined the importance of providing a visual feedback to the user, and the benefit gained from performing head compensation, especially during image-to-screen map calibration.

In the following, four directions for future research are provided. (*i*) To improve further the image-to-screen mapping model, by taking explicitly into account the spherical shape of the eyeball, and typical biometric characteristics of the human eye and oculomotor system. (*ii*) To relax the "neutral expression" constraint set for head compensation, by decoupling face motion in the image into its rigid (head movements) and non rigid (facial expressions) components. (*iii*) To generalize the approach to passive interaction surfaces such as books, newspapers, and paintings. (*iv*) To extend the framework to the problem of determining the 3D coordinates of a location pointed at in space. This will possibly involve the use of two cameras, and/or the image analysis of both eyes.

A

Appendices

A.1 Rectifying Homographies Given the Circular Points

Given a plane π, a planar homography $\mathtt{H}_\pi$ maps any point $\mathbf{X}_\pi \in \pi$ onto the image plane as $\mathbf{x}_\pi \sim \mathtt{H}_\pi \mathbf{X}_\pi$, where points are expressed in homogeneous coordinates, and the symbol "$\sim$" denotes equality up to a (complex) scale factor. In particular, the homography maps the canonical circular points on π, $\mathbf{I}_\pi \doteq [1 \;\; \imath \;\; 0]^\top$ and $\mathbf{J}_\pi = \mathrm{conj}(\mathbf{I}_\pi)$, respectively as

$$\mathbf{i}_\pi \sim \mathtt{H}_\pi \mathbf{I}_\pi \tag{A.1}$$

and $\mathbf{j}_\pi = \mathrm{conj}(\mathbf{i}_\pi)$. A homography $\mathtt{H}_\mathrm{R}$ is said to *rectify the image of* π *in a metric sense* if and only if $\mathtt{H}_\pi^{-1} = \mathtt{H}_\mathrm{R}\mathtt{H}_\mathrm{M}$ for some 4-dof metric (similarity) transformation of the plane $\mathtt{H}_\mathrm{M}$ [29]. Hence, to rectify a plane in a metric sense, only four out of the eight dofs of $\mathtt{H}_\pi$ are required, so that there exist ∞^4 possible rectifying homographies compatible with $\mathtt{H}_\pi$. The fundamental property any rectifying homography must meet is to act as the inverse of $\mathtt{H}_\pi$ at circular points:

$$\mathtt{H}_\mathrm{R}\, \mathbf{i}_\pi \sim \mathbf{I}_\pi \;\; ; \tag{A.2}$$

this is because the circular points pair is invariant to metric transformations.

A general expression for the 4-parameter family of rectifying homographies is now derived, where $\mathtt{H}_{\mathrm{R}}$ will be expressed in terms of the four free parameters and the four fixed values encoding circular point information. Let us start by recalling that the original planar homography can be decomposed as the product of a projective transformation (2 dofs), $\mathtt{H}_{\mathrm{P}}$, and an affine one (6 dofs), $\mathtt{H}_{\mathrm{A}}$:

$$\mathtt{H}_{\pi} \sim \mathtt{H}_{\mathrm{P}}\,\mathtt{H}_{\mathrm{A}} \ . \tag{A.3}$$

The latter transformation can be written as

$$\mathtt{H}_{\mathrm{A}} = \begin{bmatrix} \mathtt{A} & \mathbf{t} \\ \mathbf{0}^{\top} & 1 \end{bmatrix} \ , \tag{A.4}$$

where the invertible matrix $\mathtt{A} = \left[\begin{smallmatrix} a & b \\ c & d \end{smallmatrix}\right]$ controls scale, rotation, and skew along two orthogonal directions, and $\mathbf{t} = \left[\begin{smallmatrix} r \\ s \end{smallmatrix}\right]$ controls rigid translations. Affine transformations can move circular points, but they cannot make them leave the line at infinity:

$$\mathtt{H}_{\mathrm{A}}\mathbf{I}_{\pi} = \begin{bmatrix} a + \imath b \\ c + \imath d \\ 0 \end{bmatrix} \ . \tag{A.5}$$

eq. A.5 also shows that $\mathbf{I}_{\pi}$ is not changed by a translation: This is a consequence of the abovementioned invariance of circular points with respect to general similarity transformations. The 2-dof projective transformation brings the ideal point $\mathtt{H}_{\mathrm{A}}\mathbf{I}_{\pi}$ to the finite point $\mathbf{i}_{\pi} \doteq [\alpha + \imath\beta \ \ \gamma + \imath\delta \ \ 1]^{\top}$ of the image plane. Its inverse can be written as

$$\mathtt{H}_{\mathrm{P}}^{-1} = \begin{bmatrix} \mathtt{I}_{2\times 2} & \mathbf{0} \\ \mathbf{l}_{\infty\pi}^{\top} \end{bmatrix} \ , \tag{A.6}$$

where $\mathbf{l}_{\infty\pi} \doteq \dfrac{1}{2i}(\mathbf{i}_{\pi} \times \mathbf{j}_{\pi}) = [\delta \ \ -\beta \ \ -(\alpha\delta - \beta\gamma)]^{\top}$ is the vanishing line of π, passing through both the imaged circular points. (In order for $\mathtt{H}_{\mathrm{P}}$ to be invertible, $\det(\mathtt{H}_{\mathrm{P}})^{-1} = (\alpha\delta - \beta\gamma)$ must be nonzero.) The effect of $\mathtt{H}_{\mathrm{P}}^{-1}$ is to

bring back the imaged circular point onto the line at infinity, by changing to 0 the third component without touching the first two:

$$\mathtt{H}_{\mathrm{P}}^{-1}\mathbf{i}_{\pi} = \begin{bmatrix} \alpha + \imath\beta \\ \gamma + \imath\delta \\ 0 \end{bmatrix} \quad . \tag{A.7}$$

Now, combining eqs. A.1 and A.3 yields $\mathtt{H}_{\mathrm{P}}^{-1}\mathbf{i}_{\pi} \sim \mathtt{H}_{\mathrm{A}}\mathbf{I}_{\pi}$, so that for every nonzero (complex) value $\lambda = p + \imath q$ there must exist an affine transformation compatible with the imaged circular points:

$$\begin{bmatrix} a + \imath b \\ c + \imath d \\ 0 \end{bmatrix} = (p + iq) \begin{bmatrix} \alpha + \imath\beta \\ \gamma + \imath\delta \\ 0 \end{bmatrix} \quad . \tag{A.8}$$

Therefore, the required ∞^4 rectifying homographies are obtained by all possible choices of the parameter 4-tuple (p, q, r, s), by solving eq. A.8 for (a, b, c, d), thus finding the expression of all the affine transformations compatible with the imaged circular points:

$$\mathtt{H}_{\mathrm{A}}(p, q, r, s) = \begin{bmatrix} p\alpha - q\beta & q\alpha + p\beta & r \\ p\gamma - q\delta & q\gamma + p\delta & s \\ 0 & 0 & 1 \end{bmatrix} \quad , \tag{A.9}$$

where $\det(\mathtt{H}_{\mathrm{A}}(p, q, r, s)) = \det(\mathtt{A}(p, q)) = (p^2 + q^2)(\alpha\delta - \beta\gamma)$, and finally obtaining, from eqs. A.1 through A.3:

$$\mathtt{H}_{\mathrm{R}}(p, q, r, s) \sim \mathtt{H}_{\mathrm{A}}^{-1}(p, q, r, s)\, \mathtt{H}_{\mathrm{P}}^{-1} \quad . \tag{A.10}$$

A particularly expressive form for the rectifying homography is obtained for $r = s = 0$:

$$\mathtt{H}_{\mathrm{R}}(p, q, 0, 0) \sim \begin{bmatrix} \mathtt{A}^{-1}(p, q) & \mathbf{0} \\ \mathbf{l}_{\infty\pi}^{\top} & \end{bmatrix} \quad . \tag{A.11}$$

It is worth noting that the entries of the last row of this homography are nothing but the components of the vanishing line of π. The above expression generalizes the particular solution reported in [38], which is obtained for

$p = \frac{\delta}{\gamma^2+\delta^2}(\alpha\delta - \beta\gamma)^{-1}$, $q = \frac{\gamma}{\gamma^2+\delta^2}(\alpha\delta - \beta\gamma)^{-1}$, and also allows the derivation of the all-simple rectifying transformation of eq. 2.7, which is obtained for $p = (\alpha\delta - \beta\gamma)^{-1}$, $q = 0$.

A.2 Geometry of Virtually Rotated Views

The mapping between two views of a plane Λ rotating around a fixed axis $L_\perp \in \Lambda$ in space is a planar homology, i.e., a special 5-dof projective transformation having a line of fixed points, a vertex not on that line, and two equal and one distinct real eigenvalues [29]. The planar homology is used here to obtain a synthetic view of a 3D planar laser profile rotated by an angle ϑ around the turntable axis with respect to its original position.

Fig. A.1(a) shows a top view of the turntable plane Π, including the original laser line $L_\lambda = \Pi \cap \Lambda$ and its rotated version $L_\vartheta = \Pi \cap \Lambda_\vartheta$. These two lines are respectively the orthogonal projections onto the turntable plane of the original Λ and rotated Λ_ϑ laser planes. The lines connecting pairs of homologous points on these two planes are all parallel among them, and form a line pencil orthogonal to the plane $\Lambda_{\vartheta/2}$ bisecting Λ and Λ_ϑ, with orthogonal projection $L_{\vartheta/2}$. Now, planar homologies relate the image of any two planes whose homologous points are joined by concurrent lines: The homology vertex is the imaged intersection of these lines, while the axis is the imaged intersection of the two planes. Since in the case at hand these lines are parallel and meet at a point at infinity, the mapping *must* be a planar homology.

The problem of computing the planar homology

$$\mathtt{G}_\vartheta(\mathbf{l}_\perp, \mathbf{w}_\vartheta, \mu_\vartheta) = \mathtt{I}_{3\times 3} + (\mu_\vartheta - 1)\frac{\mathbf{w}_\vartheta\, \mathbf{l}_\perp^\top}{\mathbf{w}_\vartheta^\top\, \mathbf{l}_\perp} \tag{A.12}$$

for a given value of ϑ is discussed in the following. In eq. A.12, the imaged turntable axis $\mathbf{l}_\perp$ (2 dof, and actually independent of ϑ) is the line of fixed points, the vanishing point of the direction orthogonal to the bisectrix of the rotation angle $\mathbf{w}_\vartheta$ (2 dof) is the vertex. The characteristic invariant μ_ϑ (1

dof) is the ratio of the distinct eigenvalue to the repeated one.

Being one of the apparent fixed entities, the homology axis $\mathbf{l}_\perp$ is assumed to be known here. Hence, to obtain $\mathtt{G}_\vartheta$ there remain to be computed the homology vertex $\mathbf{w}_\vartheta$, and the characteristic invariant μ_ϑ. The latter can be expressed as

$$\mu_\vartheta = \{\mathbf{v}_\vartheta, \mathbf{v}_0; \mathbf{w}_\perp, \mathbf{w}_\vartheta\} \ , \tag{A.13}$$

where $\{\}$ denotes the usual cross ratio of four points. The four points are shown in Fig. A.1(b). Since they all belong to $\mathbf{l}_\infty$, they all are vanishing

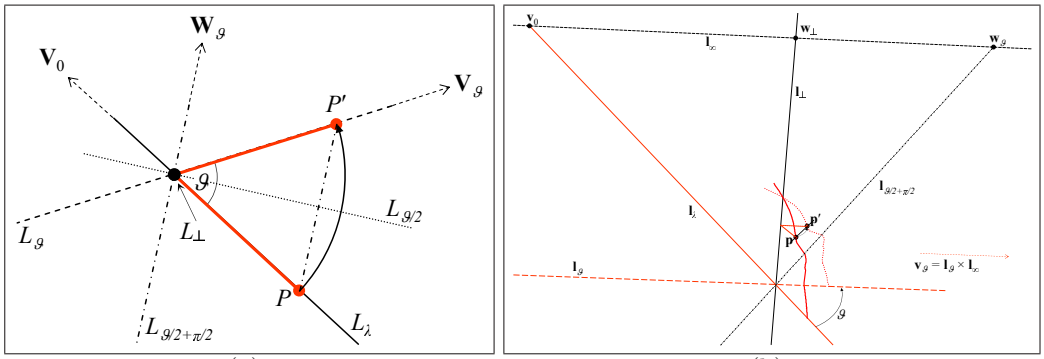


Figure A.1: Modeling a synthetic rotation of the laser plane around the turntable axis. (a): 3D geometry (top view). (b): Image geometry (camera view).

points of lines in the turntable plane. In particular, $\mathbf{v}_\vartheta$, $\mathbf{v}_0$, and $\mathbf{w}_\vartheta$ are respectively the images of the directions of the lines L_ϑ, L_λ and $L_{\frac{\vartheta}{2}+\frac{\pi}{2}}$ of Fig. A.1(a), while $\mathbf{w}_\perp$ is the image of the direction of the line obtained by intersecting the turntable plane with the plane through the turntable axis and the camera center. Being independent of ϑ, $\mathbf{w}_\perp$ and $\mathbf{v}_0$ can be computed once and for all as

$$\begin{aligned} \mathbf{w}_\perp &= \mathbf{l}_\perp \times \mathbf{l}_\infty \ , \\ \mathbf{v}_0 &= \mathbf{l}_\lambda \times \mathbf{l}_\infty \ , \end{aligned} \tag{A.14}$$

where the vanishing line $\mathbf{l}_\infty = \mathbf{i} \times \mathbf{j}$ is obtained from the (known) imaged circular points $\mathbf{i}$ and $\mathbf{j}$ of the turntable. Concerning $\mathbf{v}_\vartheta$ and $\mathbf{w}_\vartheta$, these points can be obtained as follows. First, the direction of the line L_λ is obtained from the normalized point at infinity $\mathbf{V}_0 = [\cos\vartheta_0 \ \sin\vartheta_0 \ \ 0]^\top$ computed as

$\mathbf{V}_0 \sim \mathtt{H}_t\mathbf{v}_0$, where $\mathtt{H}_t$ is the rectifying homography backprojecting imaged turntable points onto the turntable plane computed from the imaged circular points $\mathbf{i}$ and $\mathbf{j}$ as in Appendix A. Second, the points at infinity of L_ϑ and $L_{\frac{\vartheta}{2}+\frac{\pi}{2}}$ are obtained respectively as $\mathbf{V}_\vartheta = [\cos(\vartheta_0+\vartheta)\ \sin(\vartheta_0+\vartheta)\ 0]^\top$ and $\mathbf{W}_\vartheta = [-\sin(\vartheta_0+\frac{\vartheta}{2})\ \cos(\vartheta_0+\frac{\vartheta}{2})\ 0]^\top$. Finally, projecting these points onto the image yields the required vanishing points:

$$\begin{aligned} \mathbf{v}_\vartheta &= \mathtt{H}_t^{-1}\mathbf{V}_\vartheta \ , \\ \mathbf{w}_\vartheta &= \mathtt{H}_t^{-1}\mathbf{W}_\vartheta \ . \end{aligned} \tag{A.15}$$

The computational method presented above for obtaining the vanishing point of a line intersecting, in a world plane, a reference line with a given Euclidean angle, is alternative to the computation of the inverse of the Laguerre's formula discussed in [15].

Acknowledgments

I would like to acknowledge the efforts and input of my supervisors, Professor Alberto Del Bimbo and Carlo Colombo, and my colleagues of VipLab and MICC, who provided me their help during all these years. In particular, a special thanks goes to Carlo Colombo who shared with me many research problems and discussions for the development of the work presented in this document.

I would like also to remember the No Profit Italian Association "Famiglie SMA" (Families for the research on Spinal Muscular Atrophy), that founded in part the Eyemouse project, and the Gaspari family for their collaboration. Concerning 3D model acquisition, such a research work was founded in part by the Italian Ministry of University and Education (MIUR), in the context of the national project LIMA3D - "Low cost 3D imaging and modeling automatic system". I would also like to thank Federico Pernici for his support in the preliminary phase of the project.

Finally, thank to my family.

Bibliography

[1] S.M. Abdallah. *Object Recognition via Invariance*. PhD thesis, The University of Sydney, 2000.

[2] L.B. Alberti. *De Pictura*. (Reproduced by Laterza in 1980), 1435.

[3] F. Bernardini and H.E. Rushmeier. The 3d model acquisition pipeline. *Computer Graphics Forum*, 21(2):149–172, 2002.

[4] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[5] D. Beymer and M. Flickner. Eye gaze tracking using an active stereo head. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[6] L. Borgeat, G. Godin, F. Blais, P. Massicotte, and C. Lahanier. Gold: Interactive display of huge colored and textured models. *ACM Trans. on Graphics*, 24(3):869–877, 2005.

[7] J.Y. Bouguet and P. Perona. 3d photography using shadows in dual-space geometry. *International Journal on Computer Vision*, 35(2):129–149, 1999.

[8] M. Brown and D. Lowe. Recognising panoramas. In *Proc. 9th International Conference on Computer Vision*, 2003.

[9] B.-T. Chen, W.-S. Lou, C.-C. Chen, and H.-C. Lin. A 3d scanning system based on low-occlusion approach. In *Proc. International Conference on 3D Digital Imaging and Modeling*, 1999.

[10] F. Chen, G.M. Brown, and M. Song. Overview of three dimensional shape measurements using optical methods. *Optical Engineering*, 39(1):10–22, 2000.

[11] J.C. Clarke. Modelling uncertainty: A primer. Technical report, University of Oxford, 1998.

[12] C. Colombo, D. Comanducci, and A. Del Bimbo. Camera calibration with two arbitrary coaxial circles. In *Proc. 9th European Conference on Computer Vision*, 2006.

[13] C. Colombo, D. Comanducci, A. Del Bimbo, and F. Pernici. Accurate automatic localization of surfaces of revolution for self-calibration and metric reconstruction. In *Proc. IEEE Workshop on Perceptual Organization in Computer Vision*, 2004.

[14] C. Colombo and A. Del Bimbo. Interacting through eyes. *Robotics and Autonomous Systems*, 19(3-4):359–368, 1997.

[15] C. Colombo, A. Del Bimbo, and F. Pernici. Metric 3D reconstruction and texture acquisition of surfaces of revolution from a single uncalibrated view. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(1):99–114, 2005.

[16] C. Colombo, A. Del Bimbo, and A. Valli. Visual capture and understanding of hand pointing actions in a 3D environment. *IEEE Trans. on Systems, Man, and Cybernetics, Part B: Cybernetics*, 33(4):677–686, 2003.

[17] D. Comanducci. Segmentazione automatica per ricostruzione 3D di un solido di rivoluzione da singola immagine non calibrata. Master's thesis, Università degli Studi di Firenze, 2004.

[18] A. Criminisi. *Accurate visual metrology from single and multiple uncalibrated images*. PhD thesis, The University of Oxford, 1999.

[19] A. Criminisi, I. Reid, and A. Zisserman. A plane measuring device. *Image and Vision Computing*, 17(8):625–634, 1999.

[20] R. Curwen and A. Blake. Dynamic contours: Real-time active splines. In A. Blake and A. Yuille, editors, *Active Vision*. MIT Press, 1992.

[21] J. Daugman. How iris recognition works. *IEEE Trans. on Circuits and Systems for Video Technology*, 14(1):21–30, 2004.

[22] M. Dellepiane, M. Callieri, F. Ponchio, and R. Scopigno. Mapping highly detailed color information on extremely dense 3d models: The case of david's restoration. In *Proc. EuroGraphics*, 2007.

[23] A. Duchowsky. *Eye Tracking Methodology: Theory and Practice*. Springer Verlag, 2003.

[24] M.A. Fischler and R.C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24(6):381–395, 1981.

[25] R.B. Fisher and D.K. Naidu. A comparison of algorithms for subpixel peak detection. In J. Sanz, editor, *Advances in Image Processing, Multimedia and Machine Vision*. Springer Verlag, 1996.

[26] A.W. Fitzgibbon, G. Cross, and A. Zisserman. Automatic 3D model construction for turn-table sequences. In R. Koch and L. Van Gool, editors, *3D Structure from Multiple Images of Large-Scale Environments*. Springer Verlag, 1998.

[27] C.A. Glasbey and K.V. Mardia. A review of image warping methods. *Journal of Applied Statistic*, 25(2):155–171, 1998.

[28] W. Hansen and A. Pece. Eye tracking in the wild. *Computer Vision and Image Understanding*, 98(1):155–181, 2005.

[29] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[30] W. Heidrich, H. Lensch, and H.-P. Seidel. Automated texture registration and stitching for real world models. In *Proc. Pacific Graphics*, 2000.

[31] C. Hernandez, G. Vogiatzis, and R. Cipolla. Multiview photometric stereo. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(3):548–554, 2008.

[32] M. Irani and P. Anandan. All about direct methods. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*. Springer Verlag, 1999.

[33] Q. Ji and Z. Zhu. Eye and gaze tracking for interactive graphic display. *Machine Vision and Applications*, 15(3):139–148, 2004.

[34] G. Jiang, H.T. Tsui, L. Quan, and A. Zisserman. Geometry of single axis motions using conic fitting. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(10):1343–1348, 2003.

[35] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *International Journal on Computer Vision*, 38(3):199–218, 2000.

[36] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994.

[37] D. Liebowitz, A. Criminisi, and A. Zisserman. Creating architectural models from images. In *Proc. EuroGraphics*, 1999.

[38] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

[39] J. Lim, J. Ho, M.H. Yang, and D. Kriegman. Passive photometric stereo from motion. In *Proc. IEEE International Conference on Computer Vision*, 2005.

[40] D.G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal on Computer Vision*, 60(2):91–110, 2004.

[41] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. International Joint Conference on Artificial Intelligence*, 1981.

[42] L. Ma, T. Tan, Y. Wang, and D. Zhang. Personal identification based on iris texture analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(12):1519–1533, 2003.

[43] P.R.S. Mendonça, K.-Y.K. Wong, and R. Cipolla. Epipolar geometry from profiles under circular motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(6):604–616, 2001.

[44] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal on Computer Vision*, 65(1/2):43–72, 2005.

[45] C. Morimoto and M. Mimica. A novel approach to 3-d gaze tracking using stereo cameras. *IEEE Trans. on Systems, Man, and Cybernetics*, 34(1):234–245, 2004.

[46] C. Morimoto and M. Mimica. Eye gaze tracking techniques for interactive applications. *Computer Vision and Image Understanding*, 98(1):4–24, 2005.

[47] M. Pollefeys. *Self-calibration and metric 3D reconstruction from uncalibrated image sequences*. PhD thesis, K.U. Leuven, 1999.

[48] J.A. Richards. *Remote Sensing Digital Image Analysis: An Introduction*. Springer-Verlag, 1986.

[49] C. Rocchini, P. Cignoni, C. Montani, and R. Scopigno. A low cost 3d scanner based on structured light. *Computer Graphics Forum (Eurographics 2001 Conf. Issue)*, 20(3):299–308, 2001.

[50] P.J. Rousseeuw. *Robust Regression and Outlier Detection*. Wiley, 1987.

[51] S. Sullivan and J. Ponce. Automatic model construction and pose estimation from photographs using triangular splines. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(10):1091–1097, 1998.

[52] R. Szeliski. Image alignment and stitching: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2(1):1–104, 2006.

[53] A. Treuille, A. Hertzmann, and S.M. Seitz. Example-based stereo with general brdfs. In *Proc. European Conference on Computer Vision*, 2004.

[54] E. Trucco and M. Razedo. Robust iris location in close-up images of the eye. *Pattern Analysis and Application*, 8(3):247–255, 2005.

[55] E. Trucco and A. Verri. *Introductory Techniques for 3-D Computer Vision*. Prentice Hall, 1998.

[56] R. Vaillant and O.D. Faugeras. Using extremal boundaries for 3-d object modeling. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):157–173, 1992.

[57] J. Wang and E. Sung. Gaze determination via images of irises. *Image and Vision Computing*, 19(12):891–911, 2001.

[58] K.-Y.K. Wong, P.R.S. Mendonça, and R. Cipolla. Camera calibration from surfaces of revolution. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(2):147–161, 2003.

[59] L. Xu, D. Machin, and P. Sheppard. A novel approach to real-time non-intrusive gaze finding. In *Proc. British Machine Vision Conference*, 1998.

[60] L. Zhang, B. Curless, A. Hertzmann, and S.M. Seitz. Shape and motion under varying illumination: Unifying structure from motion, photometric stereo, and multi-view stereo. In *Proc. IEEE International Conference on Computer Vision*, 2003.

[61] R. Zhang, P.S. Tsai, J.E. Cryer, and M. Shah. Shape from shading: A survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(8):690–706, 1999.

[62] Z. Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76, 1997.