UNIVERSITY OF TOULOUSE

DOCTORAL SCHOOL MITT

IMAGE, INFORMATION, HYPERMEDIA

PHD THESIS

to obtain the title of

PhD of Science

of the University of Toulouse

Speciality : COMPUTER SCIENCE

Defended by

Viorica PĂTRĂUCEAN

Detection and Identification of Elliptical Structure Arrangements in Images: Theory and Algorithms

defended on January 19, 2012

Jury :

President:	Luce MORIN	-	INSA of Rennes
Examinator:	Florica MOLDOVEANU	-	University Politehnica of Bucharest
Reviewers:	Lionel MOISAN	-	Paris Descartes University
	Luce MORIN	-	INSA of Rennes
	Peter STURM	-	INRIA Grenoble Rhône-Alpes
Advisors:	Jean-Pierre JESSEL	-	University of Toulouse
	Victor-Valeriu PATRICIU	-	Military Technical Academy of Bucharest
Co-Advisor:	Pierre GURDJOS	-	University of Toulouse

Contents

Chapter 1

Introduction

Contents

1.1 Motivation

Computer vision gains more and more ground in various practical applications, as it opens up the path to technological improvement. From digital cameras able to chase a smile to self-parking cars, or remote surgery systems, it is obvious that enriching a machine with vision capabilities can only widen its functionality and improve its precision and effectiveness. The purpose of a machine that *sees* can go from assisting (helping) humans in different activities to completely substituting the need of human presence.

In the last decade, the security systems based on recognising biometric features (fingerprint, face, iris, hand), which use extensively computer vision tools, have become very popular in person identification. Generally speaking, *identification* is the process by which the identity of a user is established. A different concept is the *authentication*, which represents the process by which a service confirms the claim of a user to use a specific identity. A biometric system exploits the irrefutable and indestructible bond between a person and his/her biometric features. The identification in this case contains intrinsically a mark of *authenticity*.

A similar system in the objects world would be highly needed, as the counterfeit becomes an issue encountered in the everyday life. The traditional (1D or 2D) bar codes or the *RFID* (Radio Frequency Identification) serve as identification means, but no proof of authenticity is available, as they can be easily produced and reproduced. An important step in this direction could be the use of the Bubble Tag™, the solution patented by the Prooftag[1] company and proposed as means for object authentication (figure 1.1).

Bubble Tag™

According to the manufacturer's description, the Bubble Tag™ is the result of a chaotic self-generation of air bubbles within a transparent polymer. The randomness of the process makes

[1] www.prooftag.com

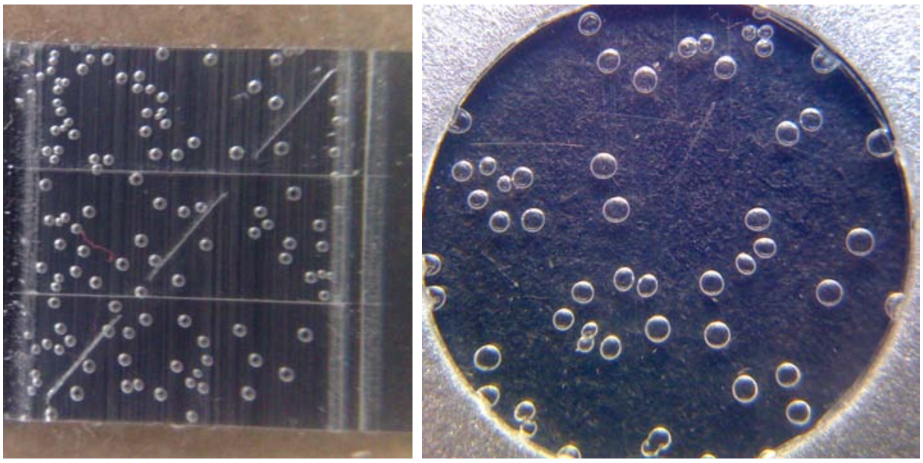


Figure 1.1: *Different versions of a Bubble Tag™*

nearly impossible the task of anticipating or planning the bubbles arrangement. The probability of obtaining two identical configurations tends to zero. Moreover, none of the existing technologies is able to reproduce this three-dimensional code, as it is impossible to generate vacuums with identical shapes, sizes and positions within a solidified material. The fact that it cannot be reproduced, not even by the manufacturer, allows qualifying the Bubble Tag™ as *unique* and *non-reproducible*, which is suitable for use in authentication purposes. By sticking a Bubble Tag™ on an object, the object becomes itself unique.

The Geowine Project

> *Geowine does* ***not*** *try to prove that there is truth in the wine (in vino veritas), but to prove that the wine is true.*

The Geowine project profits from the qualities of the Bubble Tag™ in order to put in place an innovative system for wine bottle traceability and authentication. Developed within the Agrimip Innovation Pole, the Geowine project gathers six partners (Plaimont Producteurs, CCI du Gers, École d'Ingénieurs de Purpan, LEREPS, IRIT and Prooftag), led by a twofold motivation. On the one hand, they try to anticipate a foretold directive of the European Committee concerning the wine bottle labeling, and on the other hand, they seek for an improved approach to fight against wine counterfeiting. The Geowine system will enable the final clients to track the path of their wine bottle back to its producer. Whilst the traceability is not a novelty in the wine market (similar projects already exist, e.g. Tagdevin[2]), guaranteeing the authenticity of the retrieved information is an original feature, and is achieved by using the Bubble Tag™. Providing the computer vision tools needed to deal with the automatic Bubble Tag™ identification is the motivation of this thesis.

[2] http://www.tagdevin.com/vin-qr-code-barre-2D-prestations.html

1.2 Challenges and Proposed Solutions

A classic scenario of Bubble Tag™ object authentication is similar to a biometric-based person identification. First, the Bubble Tag™ is read, e.g. one or more images of it are taken, using ordinary cameras or dedicated readers. The images serve as raw information for computer vision tools, which extract some distinctive features (landmarks), like bubble positions and sizes. The extracted data, encoded in a convenient form, yields a discriminant *signature* which is stored in a database, as a *reference signature*. This represents the *enrollment* phase. The feature extraction procedure must be repeatable, as it will be applied again, later, when the stricto sensu *authentication* takes place. In this phase, another image of the Bubble Tag™ attached to the object we wish to authenticate, is taken, and its signature is computed. This *query signature* is compared to the reference signature stored in the database, and a decision on the similarity of the two is taken. Moreover, two protocols are possible. In the *"1 to 1"* case, each Bubble Tag™ comes with an id (e.g. an alphanumeric code or a bar code), which is used as index in the database to retrieve the corresponding reference signature, and then the two signatures are compared. In a *"1 to many"* protocol, no additional id is provided, and the identification is carried out exclusively using the Bubble Tag™ data, i.e. the query signature is compared with all the entries from the database, and its correspondent, if it exists, is returned.

The current industrial solution proposed by Prooftag for the Geowine project, consists in a quasi-realisation of a *"1 to 1"* identification protocol. Once generated, each Bubble Tag™ is associated with a unique identifier, namely a DataMatrix (figure 1.2), which serves as search key in the database. At the authentication time, a preregistered image of the Bubble Tag™ associated to the query DataMatrix is returned, and the user can visually verify the correspondence between the Bubble Tag™ attached to the wine bottle he wishes to authenticate and the Bubble Tag™ from the retrieved image. The implementation of a *"1 to many"* protocol, which would automate the identification process, drew the research directions of our work.

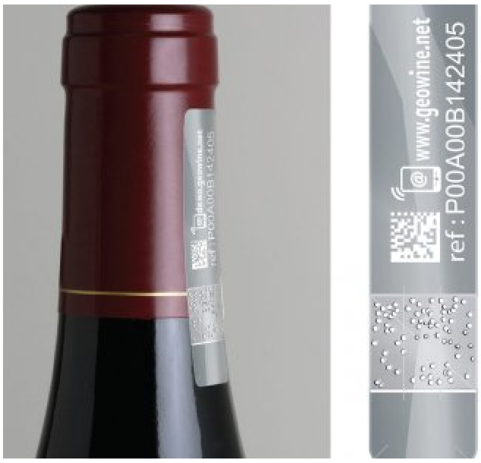


Figure 1.2: *A Bubble Tag™ and its DataMatrix identifier.*

In this context, the Bubble Tag™ identification in images amounts to matching the candidate tag contained in a query image against a database of tag images registered beforehand. An immediate strong constraint comes from the size of the database. Due to inherent variations between the shooting at the enrollment time and at the identification time, the query signature and the reference one will not necessarily be identical, an issue equally encountered in the biometric systems. Thus, searching the correspondent entry in the database for a given query signature becomes a problem of finding the nearest (not exact) match, in the sense of a predefined metric. When the database is large, the linear search, i.e. comparing the query signature with all the signatures stored in the database and keeping the one that resembles the best, becomes rather inconceivable. The uncontrollable signature variability and the size of the database entail that **the matching must use techniques specific to research in high-dimensional spaces**, for efficiency reasons (time and memory consumption). Some possible choices are the hashing procedures or the *k-d* trees. The *locality sensitive hashing* techniques prove to be suitable, as explained later. This brings the problem in the context of *image hashing.*

In general, the goal of image hashing is to map images into relatively short binary strings, to ease the database searching process. Two distinct steps can be identified in the image hashing process (figure 1.3): *feature vector extraction* and *feature vector compression* [Johnson 2003].

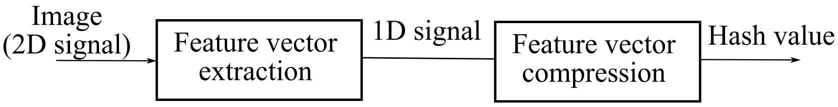


Figure 1.3: *Block diagram of image hashing process*

Our work concentrates on the former step, whereas for the latter we only recommend the use of a particular family of *locality sensitive hashing* functions [Indyk 1998, Andoni 2006], that we successfully tested.

The feature vector compression expects as input a vectorial (1D) signal. Thus, the feature extraction step is itself a special form of dimensionality reduction: the two dimensional, real (integer)-valued image is mapped to a reduced, one dimensional, real-valued vector. This entails a constraint for the signature extraction procedure, namely **the designated signature must be vectorial**.

The feature vector summarises the qualities of the image, so that two images that look alike, end up having similar feature vectors in the sense of a predefined metric (e.g. Euclidean distance), and conversely, two completely different images should have clearly distinct feature vectors. The extraction problem can be tackled at two levels. On the one hand, *low-level* techniques aim at detecting image features that are perceptually significant, such as edges, shapes (lines, ellipses) or particular spatial relations (e.g. parallel lines [Lowe 1987], concentric circles [Calvet 2011]). This is accomplished either by considering the geometric and the photometric principles underlying the image formation or by machine learning procedures [Mairal 2010]. On the other hand, *high-level* approaches use global image transforms, meant to capture image features in terms of *abstract* measures, without focusing on the local content of the image. For example, the principal component analysis highlights uncorrelated variables which account for as much of the variability in the data as possible. We suggest that the feature extraction process for Bubble Tag™ images should be carried out at a low level, because of various factors, such as the repetitiveness of the pattern, the lack of texture and possibly the very large database size[3]. Only a low level approach can capture the perceptual specificity of a bubble configuration, able to lead to its identification. To join the efforts of the industrial partner Prooftag, we followed the approach which considers the geometry of the image formation. The position and the size of the bubbles will constitute the features supporting the Bubble Tag™ identification, concentrating the research work around various aspects linked to circular features in images: detection, fitting, recognition.

Bubble Detection

The feature extraction comes thus to a problem of primitive detection in images. Depending on the Bubble Tag™ version (figure 1.1), the detection procedure seeks for ellipses (assuming that the bubbles can be represented by circles in the Euclidean plane), and possibly for line segments[4] (figure 1.1 left). The poor quality of the images, the reduced, but variable size of the bubbles, the relative big number of bubbles in one image (a Bubble Tag™ may contain up to 100 bubbles), which can moreover form groupings of two or three bubbles, raise particular challenges in the detection problem.

Classic detection methods usually apply an edge detector, followed by a fitting method (e.g. based on least-squares methods or on voting techniques), and finally use detection thresholds to decide if the detection is valid or not. Each phase requires setting up a certain number of parameters. Several drawbacks of the classic methods when applied to the bubble detection problem can be immediately stated. The parameter tuning can reveal itself as a very costly operation which needs to be repeatedly performed each time the (type of the) image changes. First, the parameters of the edge detector must be carefully chosen, to avoid eliminating useful information. Second, the detection thresholds are crucial and they directly influence the false detections. Too stringent thresholds introduce *false negatives*, whereas the permissive ones favour the *false positives.* Furthermore, the task of setting up the detection thresholds is particularly difficult, as usually they bring no control on the overall detection result. **The bubble detection must be carried out through a parameterless approach, controlling the number of false detections**, in order to guarantee reasonable results on images particularly difficult, as the Bubble Tag™ images. To this end, we chose to put the detection problem in the statistical framework of the *a contrario* methods.

The *a contrario* approach, pioneered by Desolneux, Moisan and Morel in 2000, sets itself as objective the task of providing a statistical framework which allows obtaining parameterless computer vision algorithms. Grounded on perception principles (namely the Helmholtz principle) and theories (the Gestalt theory), the *a contrario* reasoning provides the necessary mathematical tools to keep under control the number of false detections in digital images. Indeed, by formalising the Helmholtz principle, which says that no perception should be reported in a noise image, the *a contrario* approach defines what should **not** be accepted as valid detection, by using as image model a noise image. This way, when given an image x to be analysed, every

[3] A principal component analysis approach for example, is not suitable when the number of training samples is very large, as the extracted coefficients might become non discriminant.

[4] The "Z" marker is inlaid after the Bubble Tag$^{\text{TM}}$ formation, in order to ease the handling of similarity transformations (rotation, translation, scaling).

potential candidate of x, which is susceptible to appear in a noise image X of the same size as x, will be rejected, as its occurrence is considered potentially accidental.

The main concept of the *a contrario* approach is the *number of false alarms (NFA)*, which denotes two distinct quantities. On the one hand, the *NFA* of a candidate gives a measure of how likely the candidate is to appear in a noise image, by computing the expected number of candidates at least as *good*[5] as the given one in a noise image. Thus, the smaller the *NFA*, the more meaningful the candidate is. A threshold ε is used as cut-off value to distinguish the meaningful candidates. For a given ε, a candidate s is accepted as valid if it is *ε-meaningful*, i.e. if $NFA(s) \leq \varepsilon$. On the other hand, at the image level, *NFA(ε)* represents the overall expected number of ε-meaningful candidates. It is proven that, if the computation of the candidates' *NFA* takes into account the total number of possible candidates in an image, then $NFA(\varepsilon) \leq \varepsilon$. Thus, ε is the unique detection threshold of the a contrario approach, which moreover up-bounds the expected number of detections in a noise image. The precision of the result has a log-dependence on ε and the simple value $\varepsilon = 1$ reveals as convenient for practical applications. With this choice, we assume the risk of accepting on average one false positive per image.

This framework has been successfully applied in various computer vision topics (line segment detection, segmentation, fundamental matrix estimation). In our work, we use it as validation procedure in the ellipse detection problem. The proposed ellipse detector performs actually a *generalised* primitive detection (to efficiently deal with Bubble Tags™ as the one in figure 1.1 left), which includes the detection of line segments, circular arcs and elliptical arcs. Due to targeting multiple primitive families, the *NFA* of a candidate becomes a decision criterion in the model selection problem. The advantages of the proposed detector, compared with other state-of-the-art algorithms, reside mainly in its ability to control the number of false detections and to give reasonable results on all kind of images, without the need of prior parameter tuning. These qualities yield the proposed algorithm a general primitive detector, which can be applied as is, on any kind of image, regardless its content/type/origin.

Bubble Pattern Signature

The features extracted from the Bubble Tag™ images, i.e. the position and size of bubbles, carry the mark of the image formation process, namely the distortions induced by the underlying projective transformation[6]. Under projective transformations, the classic metric properties (lengths, angles, parallelism) are not preserved, thus the features extracted from two Bubble Tag™ images, taken with non-calibrated cameras, whose position is non-rigid with respect to the Bubble Tag™, are not directly comparable. A common approach to deal with this issue is to perform the Euclidean reconstruction of the plane containing the Bubble Tag™, yielding exploitable metric properties. Nonetheless, considering the constraint previously enunciated on the vectorial form of the signature required in the image hashing procedure, and for efficiency reasons, **the designated vectorial signature must be invariant under projective transformations**. To this end, we propose an original technique which computes directly (i.e.

[5] In chapter 4, we will detail the tools needed to assess the *goodness* of a candidate.

[6] We do not consider here the optical distortions that may occur in the image formation process, nor the small possible deformations induced by the curvature of the surface onto which the tag is stuck, the bottle in this case. We assume only the case where the tag has a planar surface and the images are taken using cameras that respect the (ideal) pinhole camera model.

without passing through an Euclidean reconstruction step) a signature which is left invariant under projective transformation, thus the signatures extracted from two different images of the same Bubble Tag™ will be directly comparable.

The constraints mentioned above, imposed by the industrial target application, namely the Bubble Tag™ identification, guided our work towards different computer vision topics related to circular primitives, and implicitly to the study of their projections in images, namely to ellipses. Although the Bubble Tag™ is the leitmotif of this thesis, the applicability of the proposed solutions is not restrained in any way to the Bubble Tag™ problem. The contributions on the detection, fitting and identification of circular/elliptical primitives in images yield general algorithms, which can be applied per se to any computer vision application.

1.3 Thesis Outline

The manuscript starts with a brief reminder of some basic notions on the projective and affine geometry of conics (Chapter 2).

Chapter 3 tackles the problem of conic fitting on scattered points, which is a recurrent problem in various scientific domains. In our work, the proposed bubble detection procedure requires the use of a conic fitting technique, that fulfils certain criteria, such as low complexity (the detection of about one hundred bubbles in an image entails a repeated use of the fitting procedure, yielding a considerable execution time) and good performance on incomplete data (when dealing with groupings of bubbles, the detection targets elliptical arcs, thus it is important to have a good performance when fitting is carried out on points distributed only along small circular/elliptical arcs). The chapter presents mainly a bibliographic work on existing methods for conic fitting, granting a special attention to fitting on incomplete data. A method to improve the precision of the direct algebraic fitting operators is introduced, based on the use of additional information available in images, namely the gradient orientations.

Chapter 4 studies the problem of ellipse detection within the *a contrario* framework. Due to the use of the *a contrario* validation technique, the proposed ellipse detector efficiently controls the number of false detections, while being parameterless. For a better illustration of the *a contrario* reasoning, an analogy between the *a contrario* approach and the multiple hypothesis testing framework is stated. Equally, the statistical background model is discussed, along with the usage of *NFA* as model selection criterion.

Chapter 5 deals with projective geometry topics that support the proposed Bubble Tag™ signature extraction technique, which leads to Bubble Tag™ identification in images, without the need of prior Euclidean reconstruction.

Chapter 6 details the target application of this thesis, namely the Bubble Tag™ identification in a “1 to many” authentication protocol and describes a family of suitable hashing functions, that could be used to optimise the research of the nearest match of a Bubble Tag™ in a large database.

Chapter 7 concludes the manuscript, pointing to some directions for future work.

1.4 Publications List

The work presented in this manuscript was (will be) published in francophone or international computer vision conferences:

1. V. Pătrăucean, P. Gurdjos, J. Conter, *Identification d'un code à bulles par une signature projectivement invariante*, RFIA2010, Caen.
2. V. Pătrăucean, P. Gurdjos, J. Conter, *"Bubble Tag"-based system for object authentication*, COMM2010, Bucharest.
3. V. Pătrăucean, P. Gurdjos, J. Conter, *Bubble tag identification using an invariant under perspective signature*, ICPR2010, Istanbul.
4. V. Pătrăucean, P. Gurdjos, G. Morin, J. Conter, *Détection de primitives linéaires et circulaires par une approche a contrario*, ORASIS2011, Praz-sur-Arly.
5. V. Pătrăucean, P. Gurdjos, R. Grompone von Gioi, G. Morin, *Parameterless line segment and elliptical arc detector with controlled number of false positives*, submitted.

Chapter 2

Basic Projective and Affine Geometry of Circles and Ellipses

Projective Geometry

This chapter is a brief reminder of some basic notions on the geometry of circles and ellipses, that will be used throughout this manuscript. We assume that the reader is familiar with the basics of projective geometry. A helpful introduction can be found in [Hartley 2004]; for a more formal and comprehensive presentation with proofs, see [Semple 1952, Mundy 1992]. Morin's thesis [Morin 1993] is a notable contribution to the formalisation of computer vision problems using projective geometry.

Let us denote by $P_n(\mathbb{R})$ the *real projective space of dimension* n, which is derived from the $(n+1)$-dimensional vector space $\mathbb{R}^{n+1}$. Basically $P_n(\mathbb{R})$ is the quotient space of $\mathbb{R}^{n+1} \setminus \{\mathbf{0}\}$ by the *parallelism relation* denoted by $\sim$; $P_n(\mathbb{C})$ refers to its complexification. The symbol π denotes the canonical surjection from $\mathbb{R}^{n+1} \setminus \{\mathbf{0}\}$ to $P_n(\mathbb{R})$ that maps a vector $\mathbf{p}$ to its equivalence class $\pi(\mathbf{p})$; notation-wise $\pi(\mathbf{p})$ will denote a (projective) point of $P_n(\mathbb{R})$.

Given a basis $\{\mathbf{a}_1, ..., \mathbf{a}_{n+1}\}$ of $\mathbb{R}^{n+1}$ and a point $\pi(\mathbf{p}) \in P_n(\mathbb{R})$, if $\mathbf{p} = \sum_{i=1}^{n+1} x_i \mathbf{a}_i$ then $(x_1, ..., x_{n+1})^\top$ is said to be a vector of *homogeneous coordinates* of $\pi(\mathbf{p})$ w.r.t. $\{\mathbf{a}_1, ..., \mathbf{a}_{n+1}\}$. *Abusing notation, such a point will be sometimes referred to as* $\mathbf{p}$, *without referring to a basis when the underlying base in* $\mathbb{R}^{n+1}$ *is the canonical one.*

The projective space $P_2(\mathbb{R})$ and its dual $P_2^*(\mathbb{R})$. This work focuses on the geometry of the projective plane $P_2(\mathbb{R})$. Lines of $P_2(\mathbb{R})$ (i.e., 1-dimensional projective subspaces derived from vector planes of $\mathbb{R}^3$) are in bijection with points of the *dual projective plane* $P_2^*(\mathbb{R})$ derived from the dual 3D vector space, so a projective line $l \subset P_2(\mathbb{R})$ can be identified to a "projective point" $\pi^*(\mathbf{l}) \in P_2^*(\mathbb{R})$, where π^* is the canonical surjection from the dual 3D vector space to $P_2^*(\mathbb{R})$. Note that if $\mathbf{l}$ is the vector of homogeneous coordinates of $l \subset P_2(\mathbb{R})$ with $\pi^*(\mathbf{l}) \in P_2^*(\mathbb{R})$, then $l = \pi(\ker \mathbf{l}^\top)$. Hence, we will say that a projective line l, whose "dual" is $\pi^*(\mathbf{l})$, has $\mathbf{l}$ as vector of homogeneous coordinates. A key property follows: if a point $\pi(\mathbf{p})$ lies on the line l, then $\mathbf{l}^\top \mathbf{p} = 0$.

Affine stratum of the projective plane. Given a projective line $l \subset P_2(\mathbb{R})$, an affine stratum of $P_2(\mathbb{R})$ is the subset $P_2(\mathbb{R}) \setminus l$, that is the complement of l w.r.t. $P_2(\mathbb{R})$. It is owing to the fact that $P_2(\mathbb{R}) \setminus l$ is endowed with a 2D affine structure. Such a projective line l is known as the *line at infinity* of the affine stratum of $P_2(\mathbb{R})$ and it is denoted by l_∞.

Embedding of the affine plane in the projective plane. We will embed the "ambient" affine plane by choosing a projective representation of $P_2(\mathbb{R})$ i.e., a system of homogeneous coordinates, in which the line at infinity l_∞ has

$$\mathbf{l}_\infty = (0, 0, 1)^\top$$

as vector of homogeneous coordinates, such a projective representation being called an *affine representation* of $P_2(\mathbb{R})$. Consequently, in any affine representation of $P_2(\mathbb{R})$

- affine lines with slope ϕ have homogeneous coordinate vectors in the form $(a, b, c)^\top \sim (-\sin\phi,\ \cos\phi,\ -\rho)^\top$ with $\rho \in \mathbb{R}^+$ being the orthogonal distance from the origin to the line. In other words, a, b, c are the coefficients of the line in its standard form.

- affine points $\pi(\mathbf{p}) \in P_2(\mathbb{R}) \setminus \pi(\ker \mathbf{l}_\infty^\top)$ can be derived from Cartesian coordinates, denoted as $\bar{\mathbf{p}} = \begin{pmatrix} x \\ y \end{pmatrix}$, by augmenting their vectors of homogeneous coordinates at $\mathbf{p} = \begin{pmatrix} \bar{\mathbf{p}} \\ 1 \end{pmatrix}$.

Affine Geometry of Conics in the Projective Plane

- In the affine plane, a conic $\mathcal{F}$ is the locus of points that satisfy the second degree polynomial equation
$$ax^2 + bxy + cy^2 + dx + ey + f = 0, \tag{2.1}$$
where (x, y) denote the Cartesian coordinates of a point and (a, b, c, d, e, f) the conic coefficients.

- The embedding of an affine conic in the projective space $P_2(\mathbb{R})$ is achieved by homogenisation of the polynomial equation (2.1), which yields the matrix form:
$$\mathbf{p}^\top \mathsf{C}_\mathcal{F} \mathbf{p} = 0, \tag{2.2}$$
with $\mathbf{p} = (x, y, 1)^\top$ being the vector of homogeneous coordinates of the point $\pi(\mathbf{p})$ and
$$\mathsf{C}_\mathcal{F} = \begin{pmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{pmatrix} \tag{2.3}$$
the homogeneous matrix of the conic $\mathcal{F}$. The obtained conic includes additional points compared to the affine one, namely some points at infinity, i.e. points whose third homogeneous coordinate is 0.

The affine classification of conics obeys the well-known algebraic rule about the leading principal order-2 minor of $\mathsf{C}_\mathcal{F}$, obtained by deleting its third row and column, that is

$$\mathtt{D}_2(\mathsf{C}_\mathcal{F}) \equiv \mathsf{C}_{11}\mathsf{C}_{22} - \mathsf{C}_{12}^2 = -\frac{1}{4}(b^2 - 4ac), \tag{2.4}$$

where C_{ij} denotes the element (i, j) of $\mathsf{C}_\mathcal{F}$. Hence

- for parabolas, $\mathtt{D}_2(\mathsf{C}_{\mathcal{F}}) = 0$,
- for ellipses, $\mathtt{D}_2(\mathsf{C}_{\mathcal{F}}) > 0$, and
- for hyperbolas, $\mathtt{D}_2(\mathsf{C}_{\mathcal{F}}) < 0$.

In order to exhibit the linearity of (2.2) with respect to the conic coefficients stacked as

$$\boldsymbol{\theta}_{\mathcal{F}} = (a, b, c, d, e, f)^\top, \tag{2.5}$$

we will rewrite (2.2) as:

$$F(\mathbf{p}, \boldsymbol{\theta}_{\mathcal{F}}) = (\mathbf{p} \otimes \mathbf{p})^\top \mathsf{J}\, \boldsymbol{\theta}_{\mathcal{F}} = 0, \tag{2.6}$$

where $\otimes$ denotes the Kronecker product and

$$\mathsf{J} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2.7}$$

- The dual of a conic seen as a locus of points is the envelope of lines $\pi^*(\mathbf{l}) \in P_2^*(\mathbb{R})$ satisfying the quadratic equation

  $$\mathbf{l}^\top \mathsf{C}^*_{\mathcal{F}} \mathbf{l} = 0,$$

  where the order-3 matrix $\mathsf{C}^*_{\mathcal{F}}$ plays the role dual to $\mathsf{C}_{\mathcal{F}}$ in $P_2(\mathbb{R})$. Such a "line conic" will be said a *dual conic* (see figure 2.1 right) and its matrix the *dual matrix.*

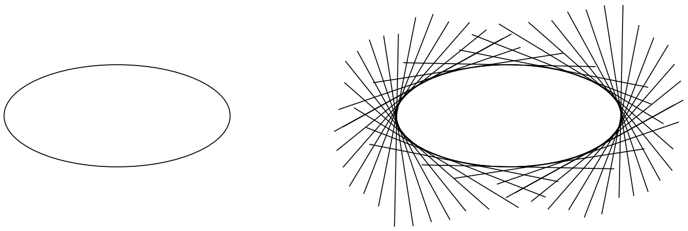


Figure 2.1: *Left: point conic. Right: dual conic (defined on lines).*

- A conic is *proper* or *degenerate*, according to whether its matrix is full rank or not. A proper conic is a self-dual figure (the relation between its matrix and dual matrix is $\mathsf{C}^*_{\mathcal{F}} \sim \mathsf{C}^{-1}_{\mathcal{F}}$), defining a bijection between $P_2(\mathbb{R})$ and $P_2^*(\mathbb{R})$, known as the *pole-polar relation*, which maps "tangent lines" $\pi^*(\mathbf{l})$ to "contact points" $\pi(\mathbf{p})$ by the projective equation

  $$\mathbf{l} \sim \mathsf{C}_{\mathcal{F}} \mathbf{p} \Leftrightarrow \mathbf{p} \sim \mathsf{C}^*_{\mathcal{F}} \mathbf{l}. \tag{2.8}$$

The (unique) line $\pi^*(\mathbf{l})$ with vector $\mathbf{l} \sim \mathsf{C}_{\mathcal{F}}\mathbf{p}$ is called the *polar of the point* $\pi(\mathbf{p})$ w.r.t. the conic locus, and dually, the point $\mathbf{p} \sim \mathsf{C}^*_{\mathcal{F}}\mathbf{l}$ is called the *pole of the line* $\pi^*(\mathbf{l})$ w.r.t. the conic envelope.

Planar homographies

Let $P_2(\mathbb{R})$ and $P'_2(\mathbb{R})$ be two projective planes.

- Any non-singular matrix $\mathsf{H} \in \mathbb{R}^{3\times 3}$ defines a one-to-one correspondence between $P_2(\mathbb{R})$ and $P'_2(\mathbb{R})$, called ***homography*** and denoted by $\mathcal{H}$, which is linear in homogeneous coordinates.
- Two matrices $\mathsf{G} \in \mathbb{R}^{3\times 3}$ and $\mathsf{H} \in \mathbb{R}^{3\times 3}$ define the same homography if and only if $\mathsf{G} \sim \mathsf{H}$, where $\sim$ is the projective equality extended to matrices.
- If points $\pi(\mathbf{p}) \in P_2(\mathbb{R})$ are mapped to points $\pi'(\mathbf{p}') \in P'_2(\mathbb{R})$ by $\mathcal{H}$ through
$$\mathbf{p}' \sim \mathsf{H}\mathbf{p},$$
then "lines" $\pi^*(\mathbf{l}) \in P^*_2(\mathbb{R})$, are mapped to "lines" of $\pi^{*\prime}(\mathbf{l}') \in P^{*\prime}_2(\mathbb{R})$ by $\mathcal{H}$
$$\mathbf{l}' \sim \mathsf{H}^{-\top}\mathbf{l},$$
where $\mathsf{H}^{-\top} = (\mathsf{H}^{-1})^\top$.
- A point conic (i.e., a locus of points) with matrix C is imaged under $\mathcal{H}$ via
$$\mathsf{C}' \sim \mathsf{H}^{-\top}\mathsf{C}\mathsf{H}^{-1},$$
whereas a line conic (i.e., an envelope of lines) with matrix C^* is imaged under $\mathcal{H}$ as
$$\mathsf{C}^{*\prime} \sim \mathsf{H}\mathsf{C}^*\mathsf{H}^\top.$$

Parametric Form of Circles and Ellipses

In (Cartesian) parametric form, the equation of a circle $\mathcal{C}$ centred in $\mathtt{O}(x_c,\ y_c)$, of radius r is:

$$\begin{cases} x(t) = x_c + r\cos(t) \\ y(t) = y_c + r\sin(t), \end{cases}$$

where $t \in [0, 2\pi]$ is a parametric variable, interpreted geometrically as the angle between the radius from the centre $\mathtt{O}$ to the point $\mathtt{P}$, with vector $\bar{\mathbf{p}} = (x(t), y(t))^\top$, and the x-axis, i.e. the angle $\angle(\overrightarrow{x}, \overrightarrow{\mathtt{OP}})$, where $\overrightarrow{x}$ is the unit vector of the x-axis.

An ellipse centred in $\mathtt{O}$, with a and b as major and minor axis respectively, and oriented under an angle θ, can be expressed in parametric form as the path of a point $\mathtt{P}$, with vector $\bar{\mathbf{p}} = (x(t), y(t))^\top$, where

$$\begin{cases} x(t) = x_e + a\ \cos t\ \cos\theta - b\ \sin t\ \sin\theta \\ y(t) = y_e + a\ \cos t\ \sin\theta + b\ \sin t\ \cos\theta, \end{cases}$$

as the parameter t varies from 0 to 2π. Note that the parameter t is **not** the angle between the line orthogonal to the ellipse in the point $\mathtt{P}$ and the x-axis, as it is for circles. The relation between the parameter t and the tangential angle $\phi(t)$ (also called the polar angle from the ellipse centre), is $\phi(t) = \frac{a}{b}\arctan(t)$. Geometrically, the polar angle can be computed by using the focal property of the ellipses, i.e. the tangent line in a point $\mathtt{P}$ belonging to the ellipse is orthogonal to the bisector of the angle formed by the lines passing through the ellipse foci, $\mathtt{F}$ and $\mathtt{F}'$, and the given point, as shown in figure 2.2.

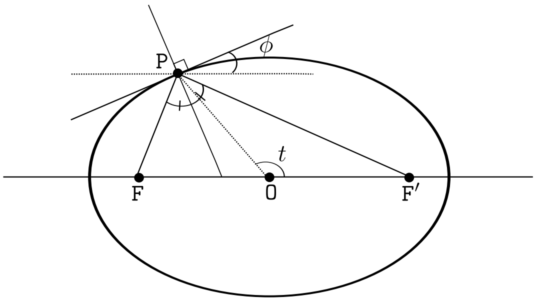


Figure 2.2: *The tangent line to an ellipse in a point* $\mathtt{P}$ *is orthogonal to the bisector of the angle formed by the lines* $\mathtt{FP}$ *and* $\mathtt{F'P}$.

Computing the Distance to a Conic

More generally, for any point of the plane, the quantity associated with the left member in (2.2) is referred to as the *algebraic distance of the point to the conic.*

The Euclidean distance between a point $\bar{\mathbf{p}} = (x, y)^\top$ and a circle $\mathcal{C}$ centred in $\bar{\mathbf{o}} = (x_c, y_c)^\top$, of radius r is $d = \sqrt{(x - x_c)^2 + (y - y_c)^2} - r$.

The power h of a point $\mathtt{P}$ with respect to the circle $\mathcal{C}$, reflects the relative distance of $\mathtt{P}$ to the circle $\mathcal{C}$, and is given by $h = s^2 - r^2$, where s is the (Euclidean) distance between the point $\mathtt{P}$ and the centre of the circle (figure 2.3 left). Points inside the circle have negative power, points outside have positive power, and points on the circle have zero power. Figure 2.3 right shows the absolute value $|h|$ of the power of a point, moving from the centre of a circle towards the exterior.

In the ellipse case, computing the exact distance between a point and an ellipse is cumbersome. We use two approximations throughout this manuscript: the *pole-polar* approximation and *Rosin distance.*

The *pole-polar* approximation [Hartley 2004, p. 31].

The pole-polar relation introduced above, can be used to obtain a linear approximation for the distance between a point and a conic. Indeed, for a given point $\pi(\mathbf{p})$, and a conic $\mathcal{F}$, if the point lies on the conic, the polar of $\pi(\mathbf{p})$, computed as $\mathbf{l} = \mathsf{C}_{\mathcal{F}}\mathbf{p}$, with $\sqrt{\mathbf{l}^\top \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \mathbf{l}} = 1$, will be tangent to the conic, thus the distance will be 0. Otherwise, the distance to the conic is

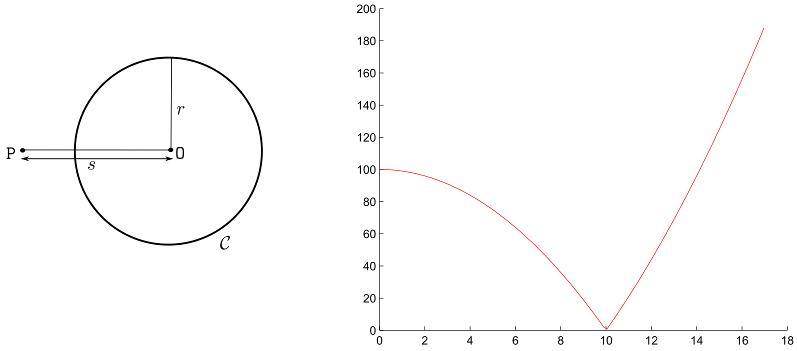


Figure 2.3: *Left: The power of a point to a circle* $h = s^2 - r^2$. *Right:* $|h|$ *for a point moving from the centre of a circle of radius* 10 *towards the exterior.*

approximated by the (Euclidean) distance between $\pi(\mathbf{p})$ and its polar $\pi^*(\mathbf{l})$, i.e. the Euclidean distance between the points $\mathbf{p}$ and $\mathbf{q}$, as illustrated in figure 2.4 left.

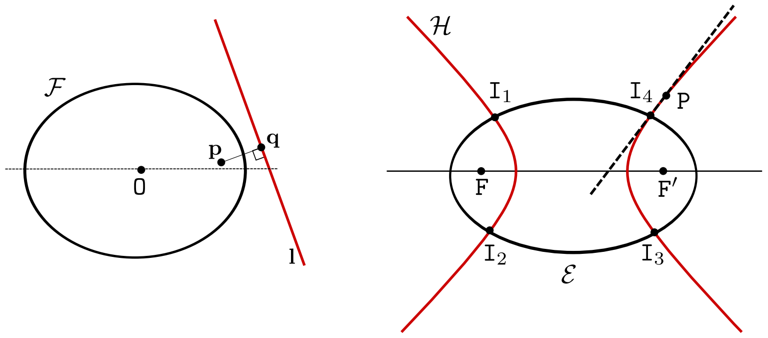


Figure 2.4: *Left: The pole-polar distance. Right: Rosin distance.*

***Rosin* distance** [Rosin 1998]. A more accurate approximation for the distance between a point P and an ellipse $\mathcal{E}$ can be computed using orthogonal conics, namely confocal families of ellipses and hyperbolae, which are known to be mutually orthogonal. Assuming that a hyperbola is piecewise linear, then the confocal hyperbola passing through P is a reasonable approximation of the straight line passing through P and normal to the ellipse, on the segment between P and the intersecting point of the ellipse with the hyperbola, as illustrated in figure 2.4 right.

To compute the Rosin distance, first the unique hyperbola $\mathcal{H}$, confocal with $\mathcal{E}$ and passing through P is determined. Next, the four intersection points $\mathtt{I}_i, i = \{1, .., 4\}$ are computed, and the Euclidean distances between P and $\mathtt{I}_i, i = \{1, .., 4\}$ are found. The minimal distance among the four is kept as the distance between the point P and the ellipse $\mathcal{E}$.

More precisely, we consider the ellipse $\mathcal{E}$ and the hyperbola $\mathcal{H}$ represented through their canonical equations:

$$\mathcal{E} : \tfrac{x^2}{a_e^2} + \tfrac{y^2}{b_e^2} = 1, \quad \mathcal{H} : \tfrac{x^2}{a_h^2} - \tfrac{y^2}{b_h^2} = 1, \tag{2.9}$$

where a_e, b_e (known) and a_h, b_h (unknown) are the major and the minor axes of the ellipse and hyperbola, respectively. The position of the foci along the x-axis for the ellipse $\mathcal{E}$ and the hyperbola $\mathcal{H}$ are given by:

$$\mathcal{E} : f_e = \pm\sqrt{a_e^2 - b_e^2} \quad \mathcal{H} : f_h = \pm\sqrt{a_h^2 + b_h^2}. \tag{2.10}$$

Since we are interested in confocal conics, $f_e = f_h$. To determine the parameters of the confocal hyperbola passing through P, the following substitutions are used:

$$\begin{array}{lclcl} A & = & a_h^2 & & \\ F & = & f_h^2 & = & f_e^2 \\ X & = & x^2 & & \\ Y & = & y^2. & & \end{array} \tag{2.11}$$

With (2.11), the equation of the hyperbola from (2.9) becomes:

$$\frac{X}{A} - \frac{Y}{F - A} = 1,$$

which produces a quadratic in A:

$$A^2 - A(X + Y + F) + XF = 0. \tag{2.12}$$

Solving (2.12) for A and re substituting in (2.11) and (2.10), yields the values of a_h and b_h. The intersection points $\mathtt{I}_i(x_i, y_i), i = \{1, .., 4\}$, are found by solving (2.9). We obtain:

$$\begin{array}{lcl} x_i & = & \pm a_h \sqrt{\frac{a_e^2(b_e^2+b_h^2)}{a_h^2 b_e^2 + a_e^2 b_h^2}} \\ y_i & = & \pm b_e b_h \sqrt{\frac{a_e^2 - a_h^2}{a_h^2 b_e^2 + a_e^2 b_h^2}}. \end{array} \tag{2.13}$$

The calculation of the (Euclidean) distance between P and $\mathtt{I}_i$ is now straightforward. Among the four solutions obtained, the shortest distance is kept.

Figures 2.5 and 2.6 give an idea on the precision of the two approximations, pole-polar and Rosin distance. It can be observed that the pole-polar approximation has a significant error and cannot be used when an accurate value of the distance between a point and an ellipse is needed. Nonetheless, being a convex function descending to its minimum as the point approaches the ellipse (figure 2.6), this approximation appears to be useful in minimisation problems, due to its simplicity. Rosin distance is accurate enough to be used as distance function, at a cost of increased complexity.

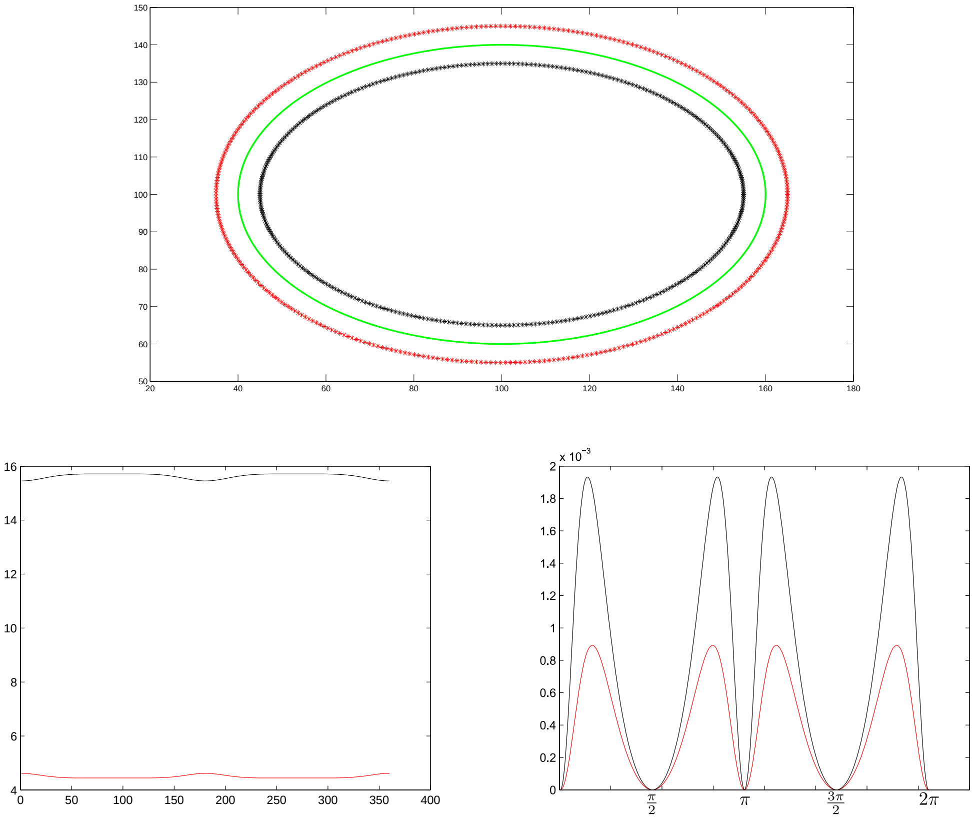


Figure 2.5: *Up: A point moving counterclockwise (starting from the x-axis) along an ellipse of semi axes* 60 *and* 40 *respectively, on the interior side (black), and on the exterior side (red) at a distance of* $d = 5$. *Down left: The pole-polar error distance. Down right: the Rosin error distance. In both cases, the error is more important when the point lies on the interior side of the ellipse.*

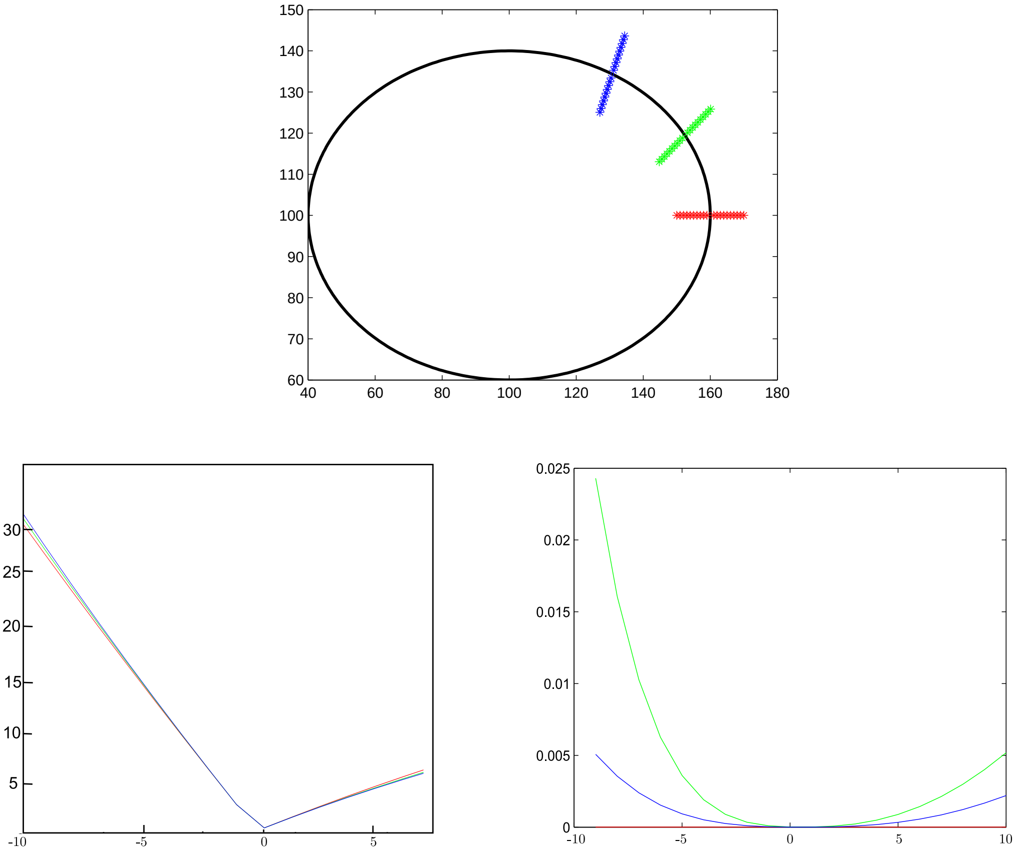


Figure 2.6: *Up: A point moving from the interior towards the exterior of an ellipse (semi-axes* 60 *and* 40 *respectively) under three different angles. Down left: The pole-polar error distance. Down right: the Rosin error distance. In both cases, the error is more significant when the point lies on the interior side of the ellipse.*

Chapter 3

Circle and Ellipse Fitting in Images Using Gradient Orientations

Contents

3.1 Introduction

The problem of fitting standard shapes or curves to measured data is a notorious issue in various fields of science and engineering (e.g. physics [Chernov 1984], biology [Paton 1970], metrology [Ahn 1997], computer graphics [Guennebaud 2007], biometrics [De Marsico 2010]). In computer vision, the conic fitting (especially ellipse and circle fitting) holds the researchers' interest as conics possess important features (like some projective invariant features), which make them suitable for use in solving particular problems like camera calibration [Wu 2006, Gurdjos 2006], object detection [Kanatani 2004, Zhang 2005], camera tracking [Calvet 2011], projective reconstruction of quadrics of revolution [Gurdjos 2009]. This chapter presents mostly a bibliographic work on circle and ellipse fitting, necessary in our context, as the bubble detector that we propose, *ELSD* (chapter 4), requires a circle/ellipse fitting step. With the constraints of the *ELSD* detector in sight, namely good performance in fitting partially occluded circles/ellipses and time efficiency, a special attention will be given to *conic fitting on incomplete data*.

Informally, *conic fitting* is the process of constructing a curve that fits well (or *best*), according to a predefined criterion, to a series of data. In image processing, these data are obtained from image *pixels*, which are endowed with integer Cartesian coordinates (relative to the image coordinate system), and possibly with some gradient information. As there exist methods that use a subpixel precision, or normalised pixel coordinates, we will use the generic term *points* when referring to pixel or subpixel measurements.

The task of fetching the points of interest is carried out by standard methods, like edge detection [Marr 1979, Canny 1986] or other heuristics that group together points with common features (e.g. gradient orientation[1], intensity, color) [Burns 1986, Grompone von Gioi 2010]. In either case, for the fitting procedure we assume that a set of input points is known and the goal is to provide the parameters of the curve.

After a good look on the fitting methods in the literature, and considering the needs of the *ELSD* detector in terms of execution time (in a Bubble Tag™ image, containing about $40 - 100$ bubbles, the fitting is performed a considerable number of times), and precision on partially occluded ellipses (the bubbles can form groupings, thus in 2D they will be approximated by elliptical arcs), we headed towards direct algebraic estimators, and we give here a means for improving their performance on incomplete data, by using additional information available in images, namely the gradient orientation.

The proposed circle/ellipse fitting technique was published as a step of a circle detector in [Pătrăucean 2011].

Chapter organisation. This chapter is organised as follows. Section 3.2 gives a classification of the most commonly used methods for conic fitting. In section 3.3, the problem is stated in a least-squares sense. Section 3.4 presents and compares briefly existing least-squares methods for circle and ellipse fitting, both geometric and algebraic procedures. A particular case of direct algebraic fitting, namely fitting using information on gradient orientation, is detailed in section 3.5, where an improved technique for circle and ellipse fitting is proposed. Section 3.6 shows the performance of the proposed method, compared to existing estimators, and section 3.7 concludes the chapter.

3.2 Classification of Methods for Conic Fitting

Many methods have been proposed in the literature to address the conic fitting problem. Two important issues need to be considered by the conic fitting techniques, namely *accuracy* and *complexity*. Usually, accurate methods are iterative, therefore costly, while the computationally efficient ones may lack precision to a certain extent. Depending on the application needs, a trade-off must be established. Figure 3.1 tries to give an image of the most commonly used.

First of all, the fitting methods can be grouped with respect to the model used to represent the input data, and we distinguish the methods based on deterministic models from those based on statistical models. The former combine voting procedures or different least-squares formulations, whereas the latter consider fitting as a statistical estimation in presence of noise.

a) Methods using deterministic models. In the class of methods considering deterministic models to represent the input data, we can further identify *deterministic* and *non-deterministic* operators. In the deterministic category, the simplest methods for conic fitting rely on some basic operations performed on the input data. For example, the moment-based method has been widely used as it provides a very simple, yet efficient technique for locating circular features when images are taken under some favorable conditions. It computes the centre of the circle as the barycentre of the input points, and afterwards, the radius is given, for example, by the

[1] Optimal methods for gradient computation are presented in [Fari 2004].

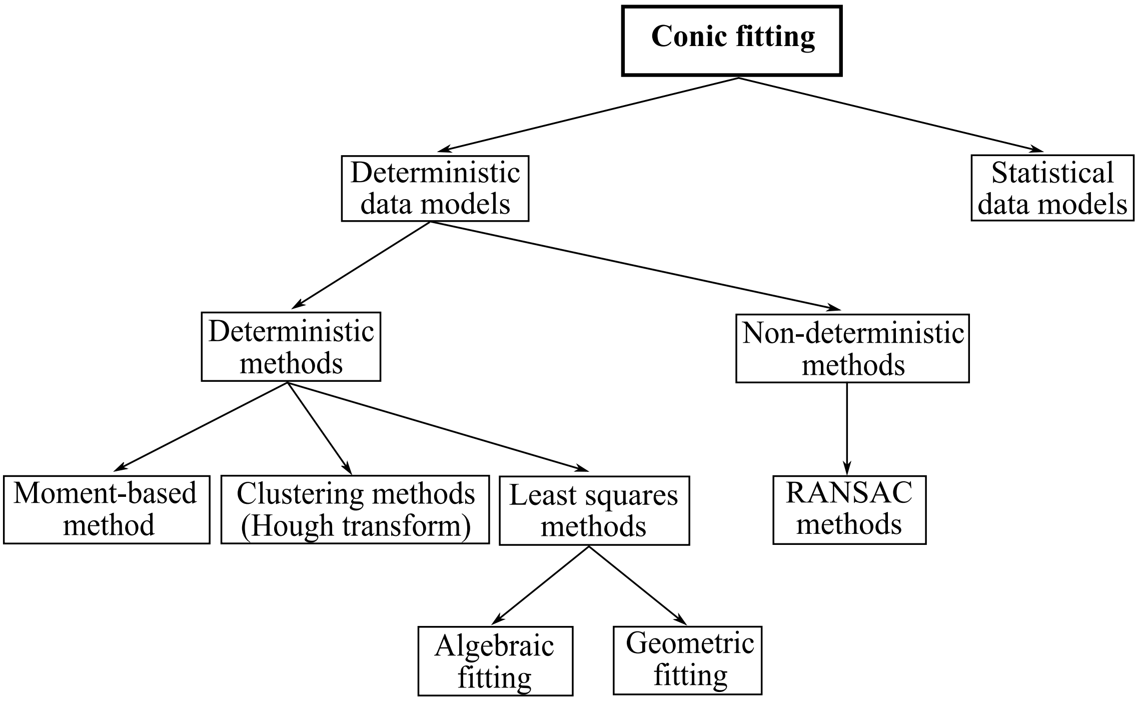


Figure 3.1: *Classification of conic fitting methods.*

mean distance between the estimated centre and the input points [Shafiq 2001]. This method is sensitive to illumination variations, and above all, to occlusions. As soon as the input points are not distributed all along the circle contour, the results can be highly erroneous. For this reason, these techniques are more likely to be used as initial estimates for some iterative, more accurate, procedures [Ahn 2001], rather than as fitting methods per se.

In the early sixties, a parametric approach for line detection (fitting) in images was proposed by Hough [Duda 1972]. Due to the generalisation proposed by Ballard [Ballard 1987], it soon became very popular in image processing for detecting arbitrary shapes and many versions have been published (see [Leavers 1993] for a survey). The Hough-based methods have the advantage of being robust to occlusions. In exchange, the high computational complexity and the non-uniqueness of the solution render them rather unsuitable for real-time applications. More details on this approach will be given in chapter 4, section 4.2.

With the least-squares *(LS)* formulation, one defines an error distance and tries to minimise in a least-squares sense this distance between the input points and the fitted curve. We identify two main classes of *LS* fitting techniques, namely *algebraic* and *geometric* fitting. They are differentiated through the definition of the error measure: the former account for methods that seek to minimise an error distance which has a geometric meaning (e.g. Euclidean distance) [Kasa 1962, Gander 1994, Ahn 2001, Sturm 2007], whereas the latter use an algebraic expression as error distance [Sampson 1982, Pratt 1987, Taubin 1991, Fitzgibbon 1999].

The non-deterministic class is represented by *RANSAC* (**RAN**dom **S**ample **C**onsensus) methods, which are iterative algorithms, consisting in two repeated steps: (1) randomly generating an hypothesis from a minimal subset of the input data (i.e. which contains enough points to compute exactly the sought model, e.g. 2 points for a line) and (2) verifying it against the whole input set. At each iteration, the equation of the curve is computed exactly using the chosen sample and afterwards, the rest of the input data (not belonging to the chosen subset), are split into *inliers* (points that belong to the curve with a given precision) and *outliers* (points that do not adapt well to the proposed model), using an error distance together with a threshold. The objective is to find the hypothesis that maximises the number of inliers [Frahm 2006], in a given number of iterations. The number of iterations needed to find the best subset can be computed adaptively, as a function of two quantities: the predicted proportion of outliers and the desired probability of finding the best subset [Hartley 2004, p. 119].

b) Methods based on statistical models. Some authors consider the conic fitting as statistical inference from noisy data. In this context, one needs to define a noise model and an optimality criterion. The standard noise model considered is the independent Gaussian noise. As optimality criterion, the *maximum likelihood (ML)* is chosen: the likelihood function obtained by substituting the observed data into the probability density of the noise model is maximised. The obtained solution achieves the theoretical accuracy bound, called the Cramér-Rao lower bound [Hartley 2004, Chernov 2005, Kanatani 2008]. It has been proven that some methods belonging to the deterministic class previously presented also reach the Cramér-Rao lower bound. Therefore, they can be considered statistically optimal. The geometric *LS* methods that seek to minimise the sum of the squares of the orthogonal distances are such an example and can be seen as direct methods that attempt to obtain the *ML* solution [Gander 1994, Ahn 2001, Sturm 2007]. *RANSAC* can equally be put in a statistical framework if instead of maximising the number of inliers, one tries to maximise the robust likelihood [Frahm 2006]. A particular statistical *RANSAC* is the variant proposed by Moisan and Stival [Moisan 2004], where the objective is to minimise the *number of false alarms*, a quantity which will be detailed in chapter 4.

Analysing the behaviour of the different fitting methods in terms of execution time and accuracy, we headed towards the deterministic *LS* formulation, and we investigated the behaviour of different algorithms when the input data are incomplete.

3.3 Problem Statement in the Least-Squares Sense

Denote by $\bar{\mathbf{p}} = (x, y)^\top$ the vector of Cartesian coordinates of a point in the Euclidean plane. Let $\mathcal{F}$ be an affine conic (ellipse, hyperbola or parabola) represented by an implicit second order polynomial equation, written in matrix form as:

$$\mathbf{p}^\top \mathsf{C}_\mathcal{F} \mathbf{p} = 0, \tag{3.1}$$

with $\mathbf{p} = \begin{pmatrix} \bar{\mathbf{p}} \\ 1 \end{pmatrix}$ being the vector of homogeneous coordinates of a point and

$$\mathsf{C}_\mathcal{F} = \begin{pmatrix} a & b/2 & d/2 \\ b/2 & c & e/2 \\ d/2 & e/2 & f \end{pmatrix} \tag{3.2}$$

being the matrix of the conic $\mathcal{F}$. The different elements of $\mathsf{C}_{\mathcal{F}}$, stacked in the 6-vector

$$\boldsymbol{\theta} = (a,\ b,\ c,\ d,\ e,\ f)^\top \tag{3.3}$$

are linked by a constraint denoted by

$$h(\boldsymbol{\theta}) = 0. \tag{3.4}$$

Given n points $\{\bar{\mathbf{p}}_i\}_{1\leq i\leq n}$, the problem of fitting to the data the conic $\mathcal{F}$ in the least-squares sense can be stated as:

$$\min_{\boldsymbol{\theta}} \sum_{i=1}^{n} (\delta(\bar{\mathbf{p}}_i, \mathsf{C}_{\mathcal{F}}))^2 \tag{3.5}$$

$$\text{subject to } h(\boldsymbol{\theta}) = 0, \tag{3.6}$$

where $\delta()$ is an *error distance* (called *error-of-fit*) between the input points $(x_i, y_i)^\top$ and the estimated curve, which is to be defined with respect to the curve type (e.g. for circle fitting, one can consider the Euclidean distance, whereas for ellipse fitting some approximations of the geometric distance are sometimes preferred, for complexity reasons).

In the context of our work, we consider an additional *term* for each point, namely a photometric consistency from which we expect a reasonable behaviour of the fitting when the input data are incomplete (see section 3.5).

3.4 Circle and Ellipse Fitting through Geometric and Algebraic Procedures

In the last decades, tremendous research work was concentrated on the *LS* formulation (3.5-3.6) of the circle and ellipse fitting in images, and the problem seems to be settled from a theoretical point of view. Nonetheless, when the input data is sampled along a small arc, many of the known estimators became instable. In this section we briefly present some reference works and try to point out their pros and cons.

Geometric Fitting of Circles and Ellipses

a) Geometric fitting of circles

Geometric fitting algorithms for circles commonly use the Euclidean distance as error-of-fit in the minimisation problem (3.5). When fitting a circle $\mathcal{C}$ of radius r and centred in $\mathsf{O}(x_c, y_c)$, a possible circle parametrisation is given by:

$$(x - x_c)^2 + (y - y_c)^2 = r^2, \tag{3.7}$$

and hence the error-of-fit is

$$\delta_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} - r. \tag{3.8}$$

The minimum obtained using this formulation is considered as a statistical optimal solution, as the problem is equivalent to the maximum likelihood estimation under the common assumption that the noise altering the input data has a Gaussian distribution [Chernov 2005]. The major drawback is its non linearity with respect to the sought-after parameters (x_c, y_c, r), requiring the use of an iterative procedure to reach its minimum. Various iterative procedures have been proposed. Gander et al. [Gander 1994] use a parametric representation of a circle in order to improve the performance of the classic non-linear least-squares problem solved using Gauss-Newton scheme. Späth introduces a set of dummy parameters along the set of parameters (x_c, y_c, r) and switches between the minimisation with respect to these dummy parameters and the real ones [Späth 1997]. This way, the function to be minimised is surely descending towards the solution. Ahn et al. [Ahn 2001] describe non-parametric algorithms for circle and ellipse fitting, with improved convergence speed compared to Gander and Späth. To lighten the computational complexity of the iterative method, Kasa [Kasa 1962] suggests using the *power of a point* (defined in chapter 2) as error distance in (3.5). So (3.8) becomes $\delta_i = (x_i - x_c)^2 + (y_i - y_c)^2 - r^2$.

In [Chernov 2005], Chernov and Lesort give an interesting study on the convergence of the iterative schemes when the estimation is carried out on incomplete data: only points sampled along a small arc (20 degrees or less) are known. The methods using the classic parametrisation (3.7) get stuck as soon as the points are sampled along a circular arc with low curvature or along a line. They recommend the use of an alternative parametrisation that accepts the equation of a line as a solution [Pratt 1987, Gander 1994, Guennebaud 2007] and describe an iterative algorithm suitable for the new parameter space. More details on this parametrisation will be given later.

b) Geometric fitting of ellipses

Solving a non-linear least-squares problem is costly, as in general, no closed-form solution exists, and therefore an iterative procedure needs to be used. In the case of ellipse fitting, another difficulty arises, namely the computation of the error-of-fit. Finding the orthogonal distance between a point and an ellipse requires solving a high degree polynomial. Safaee-Rad et al. solve a quartic equation and choose the minimal value among the four solutions [Safaee-Rad 1991]. Ahn et al. [Ahn 2001] solve simultaneously two quadratic equations through an iterative procedure. Gander et al. [Gander 1994] use a parametric representation of an ellipse, thus having a larger number of fitting parameters ($2m$ non-linear equations for $m + 5$ unknowns when m measurements are available). A similar approach was proposed in [Sturm 2007], where the authors describe a parametrisation applicable to the fitting of any type of conic, i.e. the method is not restricted to ellipse fitting. With this parametrisation, the geometric fitting of conics becomes more feasible from a computational point of view, but the procedure remains yet iterative.

More efficiently computable approximations for the orthogonal distance have been proposed. Rosin describes and compares several error-of-fit expressions, among which the most efficient are those using the focal property of ellipses [Rosin 1996] or confocal families of ellipses and hyperbolae [Rosin 1998]. The latter was described in chapter 2.

Algebraic Fitting of Circles and Ellipses

Algebraic methods substitute an algebraic distance to the geometric one used by the techniques previously mentioned, basically because it yields the cost function linear in $\boldsymbol{\theta}$, and —in the case where the constraint $h(\boldsymbol{\theta})$ is quadratic— a closed-form solution is available. Referring to (2.6), we first do some algebraic rewriting of (3.1) as

$$(\nu(\bar{\mathbf{p}}))^\top \boldsymbol{\theta} = 0, \tag{3.9}$$

where

$$\nu(\bar{\mathbf{p}}) = (\mathbf{p} \otimes \mathbf{p})^\top \mathsf{J} = [x^2,\ y^2,\ xy,\ x,\ y,\ 1], \tag{3.10}$$

denotes the Veronese embedding of the point $\bar{\mathbf{p}}$ in 5-dimensional projective space and J is the matrix given in (2.7).

The error $F(\bar{\mathbf{p}}, \boldsymbol{\theta}) \triangleq (\nu(\bar{\mathbf{p}}))^\top \boldsymbol{\theta}$ in (3.9) is called the *algebraic distance* of the point $\mathbf{p}$ to the conic $\mathcal{F}$. Under the algebraic formulation, one needs to solve (3.5-3.6) by minimising the sum of the squares of the algebraic distances. The objective function in (3.5) becomes:

$$\mathsf{F} = \sum_{i=1}^{n} (F(\bar{\mathbf{p}}_i, \boldsymbol{\theta}))^2 = \sum_{i=1}^{n} (ax_i^2 + bx_iy_i + cy_i^2 + dx_i + ey_i + f)^2,$$

and the optimisation problem becomes:

$$\min_{\boldsymbol{\theta}} \left\{ \sum_{i=1}^{n} F(\bar{\mathbf{p}}_i, \boldsymbol{\theta})^2 = \|\mathsf{D}\boldsymbol{\theta}\|^2 \right\} \tag{3.11}$$

$$\text{subject to } h(\boldsymbol{\theta}) = 0, \tag{3.12}$$

where

$$\mathsf{D} = \begin{pmatrix} (\mathbf{p}_1 \otimes \mathbf{p}_1)^\top \\ \vdots \\ (\mathbf{p}_n \otimes \mathbf{p}_n)^\top \end{pmatrix} \mathsf{J} \tag{3.13}$$

is the so-called $n \times 4$ design matrix.

a) Algebraic fitting of circles

As shown by [Chernov 2005], representing a circle using the standard equation (3.7) may introduce a significant loss in accuracy and convergence issues for the geometric fitting when the input data are sampled along circular arcs which can be fairly approximated by a line segment. An elegant way to avoid this is to consider the homogeneous parameterisation of a circle, used in [Pratt 1987, Gander 1994, Guennebaud 2007]:

$$a(x^2 + y^2) + dx + ey + f = 0. \tag{3.14}$$

The conversion between the geometric parameters of a circle —centre $(x_c,\ y_c)$ and radius r—

and the algebraic ones is given by:

$$x_c = -\frac{d}{2a}, y_c = -\frac{e}{2a}, r = \sqrt{\frac{d^2+e^2}{4a^2} - \frac{f}{a}}.$$

A key property of (3.14) is that it describes a circle $\mathcal{C}$ when $a \neq 0$ and a line when $a = 0$. We now give a projective interpretation of this property, albeit not mentioned by the authors, which explains why it holds. From the general matrix form (3.1), the equation (3.14) writes as

$$\mathbf{p}^\top \mathsf{C}_\mathcal{C} \mathbf{p} = 0, \tag{3.15}$$

where

$$\mathsf{C}_\mathcal{C} = \begin{pmatrix} a & 0 & d/2 \\ 0 & a & e/2 \\ d/2 & e/2 & f \end{pmatrix}$$

is the matrix of the circle $\mathcal{C}$. When $a = 0$ the matrix $\mathsf{C}_\mathcal{C}$ degenerates to a rank-2 matrix. If we consider now that $\mathbf{l} = (d, e, f)^\top$ is the vector of some affine line l, then it is easy to verify that when $a = 0$, we have

$$\begin{pmatrix} 0 & 0 & d/2 \\ 0 & 0 & e/2 \\ d/2 & e/2 & f \end{pmatrix} = \frac{1}{2}\left(\mathbf{l}\mathbf{l}_\infty^\top + \mathbf{l}_\infty\mathbf{l}^\top\right), \tag{3.16}$$

where $\mathbf{l}_\infty = (0, 0, 1)^\top$ denotes the vector of the line at infinity. Hence, an affine line l with vector $\mathbf{l}$ can be seen as a circle provided it is treated as a rank-2 (degenerate) conic consisting of the line-pair (l, l_∞) [Semple 1952, p. 117]. This is owing to the fact that, by definition, a conic is a circle if and only if it contains the (complex conjugate) circular point-pair at infinity [Semple 1952, p. 32] whose vectors are of the form $\mathbf{I}_\pm = (1, \pm\sqrt{-1}, 0)^\top$, *in any affine Euclidean representation of the projective plane* (i.e., obtained by embedding an affine Euclidean plane into the projective plane). In other words, if $\mathcal{C}$ is a conic with matrix $\mathsf{C}_\mathcal{C}$, then $\mathcal{C}$ is a circle if and only if

$$\mathbf{I}_\pm^\top \mathsf{C}_\mathcal{C} \mathbf{I}_\pm = 0. \tag{3.17}$$

It is easy to verify that the rank-2 conic with matrix (3.16) is a circle in the projective sense, as the circular points with vectors $\mathbf{I}_\pm$ lie on the line at infinity with vector $\mathbf{l}_\infty$ i.e., $\mathbf{l}_\infty^\top \mathbf{I}_\pm = 0$.

For the algebraic methods, this parameterisation is suitable as well as it provides the possibility to obtain a seamless algebraic estimator for circles and lines, useful in applications where both types of contours (lines and circles) are of interest (e.g. document/character analysis).

Thus, for n input points expressed in homogeneous coordinates $\mathbf{p}_i = (x_i, y_i, 1)^\top, i = \{1, .., n\}$, we obtain a linear system containing n equations of the form

$$\mathbf{p}_i^\top \mathsf{C}_\mathcal{C} \mathbf{p}_i = 0. \tag{3.18}$$

The algebraic optimisation problem is written as:

$$\min_{\boldsymbol{\beta}} ||\mathsf{D}\boldsymbol{\beta}||^2 \text{ subject to } h(\boldsymbol{\beta}) = 0, \tag{3.19}$$

where

$$\mathsf{D} = \begin{pmatrix} \bar{\mathbf{p}}_1^\top \bar{\mathbf{p}}_1 & \mathbf{p}_1^\top \\ \vdots & \vdots \\ \bar{\mathbf{p}}_n^\top \bar{\mathbf{p}}_n & \mathbf{p}_n^\top \end{pmatrix} = \begin{pmatrix} x_1^2 + y_1^2 & x_1 & y_1 & 1 \\ \vdots & & \vdots & \\ x_n^2 + y_n^2 & x_n & y_n & 1 \end{pmatrix} \tag{3.20}$$

is the so-called $n \times 4$ design matrix and

$$\boldsymbol{\beta} = (a, d, e, f)^\top$$

is a vector containing the unknown algebraic coefficients of the circle.

As for the constraint on $\boldsymbol{\beta}$, in order to avoid the trivial solution $\boldsymbol{\beta} = \mathbf{0}_4$, some authors suggest $h(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|^2 - 1$ [Paton 1970]. As reported by Bolles and Fischler in [Bolles 1981] under the constraint $h(\boldsymbol{\beta}) = a - 1$, it is easy to verify that what is minimised is the sum of the squares of the power of the n points to the circle [Kasa 1962]. When $h(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|^2 - 1$, providing the rank of the design matrix D is at least three, the (unique) "total least-squares" solution of problem (3.19) is given by the eigenvector corresponding to the smallest eigenvalue of the matrix $\mathsf{D}^\top\mathsf{D}$. This algorithm is very fast, but it is inaccurate when data are sampled along small circular arcs: it tends to grossly underestimate the radius.

An improved version is the gradient-weighted algebraic fit (*GRAF*), that was originally applied for ellipse and hyperbola estimation [Sampson 1982]. One minimises the sum of the squares of the *Sampson error*, which represents the algebraic distance normalised by the gradient norm. From the geometric point of view, *GRAF* works better than the simple algebraic scheme, as it can be shown that the new algebraic measure is a linear approximation of the orthogonal (Euclidean) distance. Unfortunately, there is no closed solution for this formulation. However, there are two approximations proposed by Pratt [Pratt 1987] and Taubin [Taubin 1991], that lead to non-iterative solutions. But, although efficient to solve, these approximations still act poorly on incomplete noisy data, as it will be shown in the experimental section 3.6.

b) Algebraic fitting of ellipses

For the ellipse algebraic fitting, one considers the general equation of a conic (3.1), and the optimisation problem (3.11-3.12). Different linear or quadratic constraints $h(\boldsymbol{\theta})$ have been proposed: $f = 1$ [Rosin 1996], $a + c = 1$ [Gander 1994, Rosin 1996], $a^2 + \frac{1}{2}b^2 + c^2 = 1$ [Bookstein 1979]. However, care must be taken when imposing such constraints, as they can affect the invariance of the estimator with respect to similarity transformations (translation, rotation, scale): for example, $f = 1$ is not invariant (see [Rosin 1996] for a review). The direct algebraic solution is easy to compute, but the results are inaccurate on incomplete data. Different weighting schemes have been proposed, that try to bring the algebraic solution closer to the geometric one [Sampson 1982, Gander 1994]. The error distance used by the *GRAF* method mentioned above, became a standard in computer vision for various estimation problems (2D homography estimation, conic fitting) as the solution is considered statistically optimal, in the sense that the covariance matrix of the parameter estimates satisfies the Rao-Cramér lower bound [Chernov 2005, Kanatani 2008]. This is true, however, only for small noise levels [Kanatani 2008]. Kanatani et al. give a deep analysis of this problem in [Kanatani 2010].

The problem statement previously given does not restrict the fitted curve to be an ellipse. As it uses the equation of a general conic, the result can be an hyperbola or a parabola, even when given elliptical data. A number of articles addressed this issue, and proposed different iterative schemes. In [Porrill 1990], a Kalman filter is used and at each iteration the discriminant $b^2 - 4ac < 0$ is tested and non-ellipse fits are rejected. Haralick and Shapiro [Haralick 1992] had a different approach: they considered particular expressions for the parameters (a, b, c) that ensure the discriminant to be negative. Indeed, if $a = p^2, b = 2pq, c = q^2 + r^2$, the discriminant becomes $-4p^2r^2$, which is surely negative.

Fitzgibbon et al. [Fitzgibbon 1999] were the first to propose a direct (non-iterative) ellipse-specific fitting. They show that the ellipticity constraint $4ac - b^2 > 0$ can be conveniently incorporated into the constraint $h(\boldsymbol{\theta})$ in the quadratic form

$$h(\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathsf{L}\boldsymbol{\theta} - 1,$$

where L is a 6×6 matrix, defined later in (3.27). Moreover, this constraint is covariant with affine transformations of the data points. All-in-one, the proposed estimator ensures ellipticity and invariance under Euclidean transformations. The same constraint was used in [Ouellet 2008], but in the dual form, in order to estimate a *dual ellipse*. The drawback of Fitzgibbon et al.' approach is its tendency to underestimate the eccentricity of the ellipse when the estimation is done on incomplete data. More details on this will be given in section 3.5.

Comparison between Fitting Methods

The absence of an objective criterion to decide which estimate is better makes the direct comparison between geometric and algebraic methods somehow irrelevant. We can however, show the advantages and the drawbacks of each approach.

At first glance, the computational complexity is at stake. From this point of view, the classification into geometric and algebraic methods should be dropped. The estimators are to be divided rather into iterative (geometric or algebraic) and non-iterative (direct) methods. The conic fitting being naturally a non-linear problem, only iterative methods achieve the statistical optimum. But this comes with all the computational cost and the instability that an iterative procedure involves. On the other hand, the direct algebraic operators are computationally very efficient, but the reliability of the estimated parameters cannot be assessed, as the error distance does not coincide with measurement guidelines.

In the past, drawbacks of the direct algebraic methods were the lack of affine invariance, or the fact that the fitting procedure can end in an unintended geometric feature (e.g. an hyperbola instead of an ellipse) [Ahn 2001]. These problems are solved through Fitzgibbon et al.' technique. Moreover, improving the conditioning of the equation system by means of data normalisation [Hartley 2004, p. 107] yields a less sensitive procedure to numerical error.

On these grounds, we conclude that the direct algebraic methods are a reasonable trade-off between efficiency and accuracy. In the following section we enquire into the possibility of improving the performance of direct algebraic estimators on incomplete data, by using additional information available in images, namely the gradient orientation.

3.5 Direct Algebraic Fitting of Circles and Ellipses Using Gradient Orientation

Estimating a conic from pixel coordinates identified as local maxima (given by a Canny edge detector for example) through a direct algebraic operator is inaccurate if the data are not sampled all along the contour. An improved estimation is obtained if the positions are known with a subpixel precision. This can be achieved by a parabolic interpolation of the gradient maximum along the edge direction or by considering regions centred at the maximum gradient location, together with a correction term based on the local curvature [Heikkila 1998]. The former is simple, but noise-sensitive, whereas the latter lowers the noise sensitivity at the cost of computational complexity. It is obvious that a method directly exploiting the gradient orientations of the input data could greatly simplify the process. Moreover, this idea is in agreement with the *a contrario* approach —the mathematical ground of the proposed detector *ELSD* (chapter 4)— which promotes the use of gradient orientations for alignment (line segment, ellipse) detection. Therefore, we will try to exploit the gradient orientation in order to improve the direct algebraic fitting on incomplete data.

A first step in this direction was made by Forstner's operator for circular feature fitting [Forstner 1987]. It computes the gradient field of the region encompassing the feature and then the centre of the circle is located as the intersection point (in a least-squares sense) of the support lines of the gradient vectors. The contribution of each line is weighted by the squared magnitude of the gradient. Forstner's operator is fast and simple. It yields though only the centre location, not the radius. The same idea, adapted for the 3D case, is used by Guennebaud et al. for spherical fitting in a computer graphics application [Guennebaud 2007]. Their procedure minimises separately two key constraints: positional and derivative. First, the derivative constraint is used to locate the centre, and subsequently, the positional constraint provides the radius information.

For ellipse fitting, the idea cannot be straightforwardly applied, as the support lines of the gradient vectors on the boundary of the ellipse do not converge towards its centre. To overcome this, Ouellet performs the ellipse fitting in the dual space, by exploiting the tangential constraints entailed by the gradient orientations of the given points [Ouellet 2008] (refer to chapter 2 for more details on the duality relation). Using the tangential information, the effect of the noise causing angular errors in the gradient orientation is reduced compared to Forstner's operator, which exploits directly the orientation information. When locating circle centres, the angular errors can have a leverage effect due to the distance between the edges and the intersection point. The effect is dimmed if the tangent lines are used instead.

Although these methods improve the direct algebraic fitting while preserving a low complexity, they do not exploit efficiently all the available information: Forstner and Guennebaud use successively the derivative and the positional constraints, while Ouellet uses only the tangential constraints. We suggest that the simultaneous usage of the positional and of the tangential information can further improve the results and we present a very simple direct algebraic operator, which takes into account simultaneously the positional and the tangential constraints.

Let us consider the one-to-one mapping between points $\mathbf{p}$ and lines $\mathbf{l}$ of the projective plane, given by the *pole-polar duality* w.r.t. a proper —i.e., non degenerate— conic $\mathcal{C}$, whose matrix is

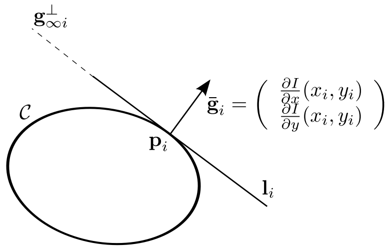

$\mathbf{g}_{\infty i}^{\perp}$
$\mathcal{C}$
$\bar{\mathbf{g}}_i = \begin{pmatrix} \frac{\partial I}{\partial x}(x_i, y_i) \\ \frac{\partial I}{\partial y}(x_i, y_i) \end{pmatrix}$
$\mathbf{p}_i$
$\mathbf{l}_i$

Figure 3.2: *Constraint entailed by the gradient orientation: the polar of the point* $\mathbf{p}_i$ *is orthogonal to the gradient orientation.*

C (cf. equation (2.8)):

$$\mathsf{C}\mathbf{p} \sim \mathbf{l}, \tag{3.21}$$

where $\sim$ denotes the projective equality. We remind that the point $\mathbf{p}$ is said to be the pole of the line $\mathbf{l}$ w.r.t. the conic $\mathcal{C}$, and conversely, the line $\mathbf{l}$ is said to be the polar (line) of $\mathbf{p}$.

As before, let $\mathbf{p}_i = (x_i, y_i, 1)^\top$ be a point given by its "embedded" Cartesian xy-coordinates. Denote by

$$\bar{\mathbf{g}}_i = \left(\frac{\partial I}{\partial x}(x_i, y_i), \quad \frac{\partial I}{\partial y}(x_i, y_i\right)^\top$$

the vector of the image gradient computed at (x_i, y_i) in the x and y directions of the image (affine) plane respectively. If $\mathbf{p}_i$ belongs to the conic $\mathcal{C}$, then its polar line $\mathbf{l}_i$ is the line tangent to $\mathcal{C}$ at $\mathbf{p}_i$, and corresponds in the affine plane to the line passing through $\mathbf{p}_i$, whose direction is orthogonal to the image gradient vector (see figure 3.2). In other words, projectively speaking, $\mathbf{l}_i$ is the line through $\mathbf{p}_i$ and the point at infinity associated with the direction orthogonal to the image gradient vector, whose vector of homogeneous coordinates is normalised as

$$\mathbf{g}_{\infty i}^{\perp} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \\ 0 & 0 \end{pmatrix} \frac{\bar{\mathbf{g}}_i}{\|\bar{\mathbf{g}}_i\|}.$$

This means that $\mathbf{g}_{\infty i}^{\perp}$ clearly lies on $\mathbf{l}_i$ i.e,

$$\mathbf{l}_i^\top \mathbf{g}_{\infty i}^{\perp} = 0,$$

from which, using the substitution given in (3.21), we obtain a new equation, linear in the elements of C (stacked in the vector $\boldsymbol{\theta}$):

$$\mathbf{p}_i^\top \mathsf{C} \mathbf{g}_{\infty i}^{\perp} = (\mathbf{g}_{\infty i}^{\perp} \otimes \mathbf{p}_i)^\top \mathsf{J} \boldsymbol{\theta} = 0. \tag{3.22}$$

Together with the positional constraints (3.18), the linear equation system using the gradient information writes as:

$$\begin{pmatrix} (\mathbf{p}_i \otimes \mathbf{p}_i)^\top \\ (\mathbf{g}_{\infty i}^{\perp} \otimes \mathbf{p}_i)^\top \end{pmatrix}_{2\times 9} \mathsf{J}_{9\times 6} \boldsymbol{\theta}_6 = 0. \tag{3.23}$$

If now we define a second term of algebraic error $G(\bar{\mathbf{p}}_i, \bar{\mathbf{g}}_i, \boldsymbol{\theta}) \triangleq (\mathbf{g}_{\infty i}^{\perp} \otimes \mathbf{p}_i)^\top \mathsf{J} \boldsymbol{\theta}$, then the problem becomes:

$$\min_{\boldsymbol{\theta}} \left\{ \sum_{i=1}^{n} F(\bar{\mathbf{p}}_i, \boldsymbol{\theta})^2 + G(\bar{\mathbf{p}}_i, \bar{\mathbf{g}}_i, \boldsymbol{\theta})^2 = \|\mathsf{Q}\boldsymbol{\theta}\|^2 \right\} \tag{3.24}$$

$$\text{subject to } h(\boldsymbol{\theta}) = 0, \tag{3.25}$$

where

$$\mathsf{Q} = \begin{pmatrix} \vdots \\ (\mathbf{p}_i \otimes \mathbf{p}_i)^\top \\ (\mathbf{g}_{\infty i}^{\perp} \otimes \mathbf{p}_i)^\top \\ \vdots \end{pmatrix} \mathsf{J} \tag{3.26}$$

is a $2n \times 6$ matrix.

Using the unit constraint $h(\boldsymbol{\theta}) = \|\boldsymbol{\theta}\|^2 - 1$, providing the rank of the design matrix D is at least five, the (unique) “total least-squares” solution of the problem is again given by the eigenvector corresponding to the smallest eigenvalue of the matrix $\mathsf{Q}^\top\mathsf{Q}$.

This fitting technique can be applied seamlessly for both circle and ellipse fitting. To improve the numerical stability, the input data are first normalised as described in [Hartley 2004, p. 107]. It should be noted that with this constraint, each point $\mathbf{p}_i$, expressed in homogeneous coordinates as $(x_i, y_i, 1)^\top$, enriched with the gradient information $\bar{\mathbf{g}}_i$, contributes with two linearly independent equations. In the classic methods, using only the positional constraint yields one independent equation per point. Moreover, the norm of the gradient of $\mathbf{p}_i$ could serve as weight for the constraints it entails. As the experiments showed no clear improvement, this weighting scheme was dropped.

Additionally, Fitzgibbon *et al.*’s constraint can be imposed to ensure the fitted curve is an ellipse. In matrix form, the proposed quadratic constraint $h(\boldsymbol{\theta}) = b^2 - 4ac - 1$ is written as $h(\boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathsf{L} \boldsymbol{\theta} - 1$, where L is:

$$\mathsf{L} = \begin{pmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3.27}$$

The problem can be solved by the method of Lagrange multipliers [Fitzgibbon 1999] and a closed-form solution exists. Section 3.6 will show to what extent and in which cases the proposed technique can improve the behaviour of the direct algebraic conic fitting operator.

3-Points (Exact) Ellipse Fitting

Some problems require fitting exactly an ellipse to a minimal set of points, e.g. in a *RANSAC*-like operator or in some methods based on variants of the Hough transform. In classic ellipse fitting methods, five independent equations are needed to determine completely the five parameters of an ellipse, thus five points is the minimum needed to exactly compute the ellipse. Nonetheless, using simultaneously the positional and the derivative constraints reduces the minimal number of points, as each point contributes with two independent equations. Within a *RANSAC* approach, using a reduced number of points diminishes significantly the number of trials needed to find with a high probability a sample free of outliers, that would provide a model explaining the input data[2]. With the proposed approach, three points (or more precisely two and a half points) are enough to exactly compute the ellipse parameters. The improvement in the execution time is significant, as the proportion of outliers grows: the search space can be reduced by an order of magnitude. However, the idea of using both positional and derivative constraints to exactly fit an ellipse was already present in the literature, and used in Hough-like voting procedures, such as the randomised Hough transform (*RHT*) [Xu 1990]. Figure 3.3 shows the technique used in *RHT* to compute the parameters of an ellipse using three points, endowed with Cartesian coordinates and with some information on the gradient orientation in the neighbourhood of the points. Although the two techniques use similar ideas, the proposed method has the advantage of being easily generalisable for a set of n input points, which can be put in a least-squares fitting formulation.

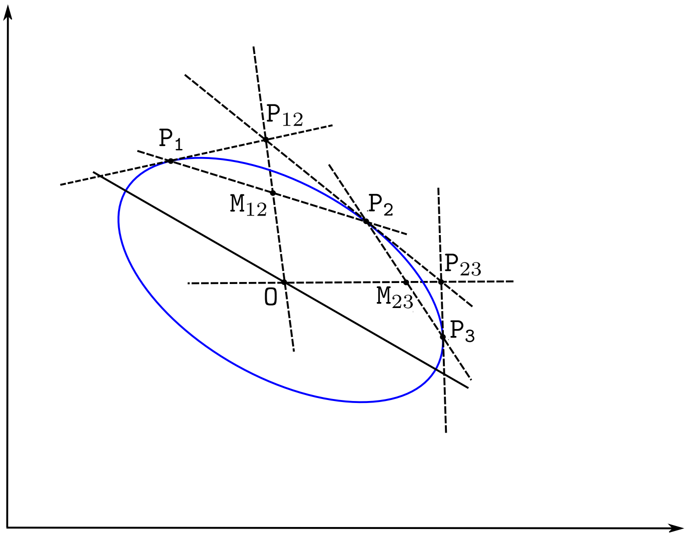


Figure 3.3: *3-points ellipse fitting used by RHT: Given three points* $\mathtt{P}_1$*,* $\mathtt{P}_2$*, and* $\mathtt{P}_3$ *belonging to an ellipse, first the tangents to the ellipse are computed using a least-square technique on the gradient orientations of the neighbourhood of each point. Then the intersection points* $\mathtt{P}_{12}$ *and* $\mathtt{P}_{23}$ *of each pair of tangents are computed, as well as the midpoints* $\mathtt{M}_{12}$ *and* $\mathtt{M}_{23}$ *of the segments* $\mathtt{P}_1\mathtt{P}_2$ *and* $\mathtt{P}_2\mathtt{P}_3$*. The intersection of the lines* $\mathtt{P}_{12}\mathtt{M}_{12}$ *and* $\mathtt{P}_{23}\mathtt{M}_{23}$ *gives the position of the centre* $\mathtt{O}$ *of the ellipse. The rest of the three parameters (axes and orientation) are found by solving the system obtained with the three equations showing that* $\mathtt{P}_1$*,* $\mathtt{P}_2$*, and* $\mathtt{P}_3$ *belong to the ellipse.*

[2] For a model defined by s parameters, if the desired probability of reaching an outlier-free sample is p and the predicted proportion of outliers is ε, then at least N selections (each of s points) are required, where $N = \frac{\log(1-p)}{\log(1-(1-\varepsilon)^s)}$ [Hartley 2004].

3.6 Results

The behaviour of the proposed operator was compared first against direct algebraic circle estimators. Taubin's operator [Taubin 1991] mentioned in section 3.4 and the Guennebaud et al.' method [Guennebaud 2007], described above, were considered. The former (called *Taubin* in figure 3.5) stands for one of the most efficient direct algebraic estimator among the classic operators, i.e. that use only the positional constraint, whereas the latter (called *GradDer*) uses the derivative and the positional constraints sequentially, not simultaneously as the proposed one (called *GradTan*).

Tests were carried out on real images to assess the performance of the proposed method on incomplete data: only points distributed along arcs of $\sim 45°$ are considered. A printed paper containing drawn circles of different radii (figure 3.4) was photographed in different lighting conditions, yielding 250 small images, each containing one circle. The radii values vary between 5 and 90 pixels.

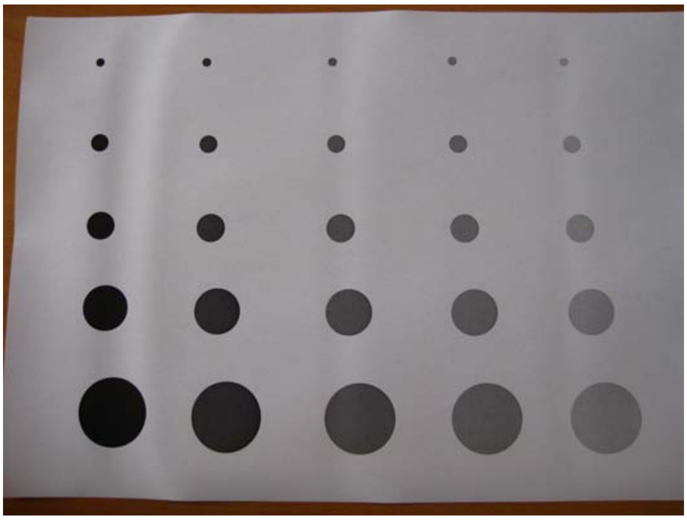


Figure 3.4: *Test image*

The images were divided into 10 classes, according to the radius value (first class: radius of $5 - 10$ pixels, second class: radius of $10 - 20$ pixels, third class: radius of $20 - 30$ pixels etc.).

In figure 3.5, each class is represented by the mean radius, and the corresponding mean relative error on the estimated radius is given. For radius values inferior to ~ 30 pixels, the proposed method *GradTan* is the most efficient, while *Taubin* has significant errors. The results show that the tangential constraint is more suitable than the derivative constraint, thus *GradDer* is less accurate than *GradTan*.

When the circles are relatively big (> 35 pixels), the accuracy of *Taubin* improves significantly, and even overtakes slightly *GradTan*. This shows that when few points are available, it is vital to exploit all the information they possess, whereas when enough data are provided, the gradient orientation does not bring significant improvement.

The proposed ellipse fitting method was compared against Ouellet's method, using simulation

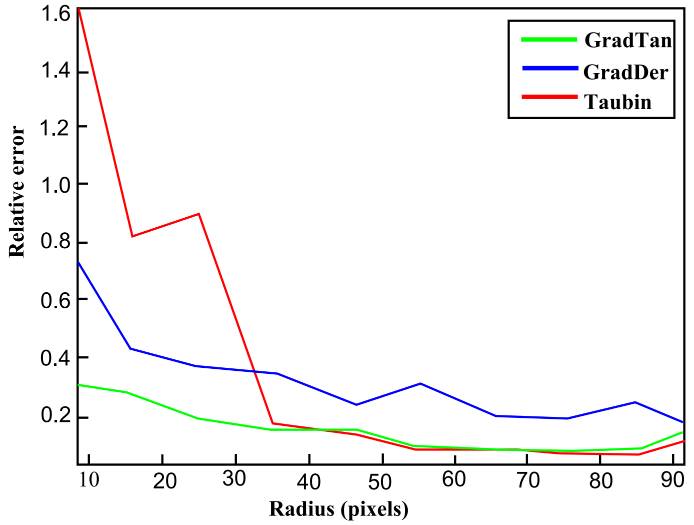


Figure 3.5: *Relative error on the estimated radius*

Mean (maximum) error on the estimated centre

Method	0%	2%	4%	6%	8%
GradTan	0.00628(0.0337)	0.0657(0.2175)	0.0989(0.3365)	0.1217(0.3397)	0.1381(0.4030)
Ouellet	0.00556(0.0315)	0.0721(0.2256)	0.1129(0.3727)	0.1418(0/4504)	0.1592(0.4177)

Mean (maximum) error on the estimated axes

Method	0%	2%	4%	6%	8%
GradTan	0.9646(1.6053)	0.9624(1.6542)	0.9488(1.6589)	0.9543(1.7351)	0.9566(1.7459)
Ouellet	1.1800(1.7557)	1.2191(1.8753)	1.2727(2.1181)	1.3410(2.6373)	1.3594(2.6373)

Table 3.1: *Estimation error on ellipse centre and axes.*

images and the results are given in tables 3.1 and 3.2. As expected, the results of *GradTan* are better, especially on incomplete and/or noisy data.

The obtained results reflect the improvement brought by using the additional constraint given by the gradient orientation in the least-squares formulation of the fitting problem.

3.7 Conclusion

In the context of our work, the problem of fitting a circle/an ellipse on incomplete data arose. The conic fitting problem has been extensively studied in the last decades and can be considered as settled from a theoretical point of view, but there are still unsolved issues when the data are incomplete. After investigating different directions in the literature, we focused on direct

Mean (maximum) error on the estimated centre for incomplete data (noise 2%)

Method	75% of contour	50% of contour
GradTan	0.1136(0.4465)	0.5972(4.8615)
Ouellet	0.2018(1.0942)	3.4944(8.3421)

Mean (maximum) error on the estimated axes for incomplete data (noise 2%)

Method	75% of contour	50% of contour
GradTan	0.5159(0.9927)	0.9068(5.1210)
Ouellet	0.8299(1.5326)	4.6549(12.0225)

Table 3.2: *Estimation error on ellipse centre and axes for incomplete data.*

algebraic estimators, for simplicity and efficiency reasons. To improve the accuracy of the direct algebraic fitting for circles and ellipses on incomplete data, we presented a simple technique that exploits simultaneously the positional and the tangential information of the given points. The tests carried out on simulated and real images show the efficiency of the proposed fitting operator, compared to similar works from the literature.

Chapter 4

A Contrario Ellipse Detection in Images

Contents

4.1 Introduction

The identification process of a Bubble Tag™ requires a feature extraction step in which the perceptual entities contained in a Bubble Tag™ image are captured and used subsequently to compute a discriminant signature. More generally, the first questions that arise naturally in the context of feature detection are:

1. *Which features are perceptually meaningful when analysing an image?*

2. *How to formalise the quality of a feature of being "perceptually meaningful"?*

3. *How to decide which features are "enough" perceptually meaningful to be detectable and, by extension, how to design automatic algorithms that detect them?*

Thus, the problem comes to detecting meaningful features in images. In this work, we are actually dealing with circle/ellipse detection, assuming that the bubbles can be modeled by such regular shapes. This problem is not new, and it has been intensively studied over the last decades, resulting in numerous detection procedures. Still, there remain some unsettled issues, namely the *parameter tuning* and the *control of false detections*. These issues are closely related and, regarding the latter, two types of detection error are distinguished: *false positives* and *false negatives* (referred to as Type I and Type II errors respectively, in the statistics vocabulary, cf. section 4.4). A *false positive* appears when the detection procedure reports a feature that does not exist (e.g., due to artifacts in measures). On the opposite side, a *false negative* detection is the case where the detection procedure does not find a feature that exists, i.e., that should be considered as "enough" perceptually meaningful.

The points mentioned above suggest that so far, the answer to the third question is flawed. The reason for this might be precisely the fact that classic detection methods focus only on this question, ignoring the first two. The meaningfulness of a feature is assessed by means of parameter tuning done empirically, relative to the specificity of an image. This results, obviously, in false detections when the algorithms are applied on a different (type of) image (as illustrated in figure 4.1).

An original and mathematically sound approach, which takes into account all the three questions above, was proposed by Desolneux et al. [Desolneux 2000, Desolneux 2007] and is now known as the *a contrario* approach. The answer to the first question was already given by the studies of the Gestalt school, which formulated the Gestalt theory [Metzger 1975], as a scientific attempt to state the laws of visual reconstruction. The *a contrario* approach assumes the mission of translating the Gestalt theory into a mathematics and computer vision program. To address the second question, the *a contrario* group formalises the *Helmholtz principle* which says that visual perceptions are generated only by spatial arrangements that are geometrically too regular to have occurred in noise. The equivalent form, i.e. *there is no perception in white noise*, allows defining general detection thresholds; this tackles the third question.

Thus, the *a contrario* approach answers the three questions by stating a theory of perception, which draws its inspiration from the Gestalt laws to describe what is *perceptible* in images, and by formalising the Helmholtz principle to establish general detection thresholds. These are the steps to follow in order to obtain parameter-free computer vision procedures. Several algorithms grounded on this approach have been proposed to tackle different essential problems in computer vision: line segment detection [Grompone von Gioi 2010], segmentation [Burrus 2009], computation of the fundamental matrix [Moisan 2004], image matching [Rabin 2009].

In our work, we use the *a contrario* approach to address the problem of bubble detection in images, by implementing a combined *a contrario* line segment and circle/ellipse detector. Being parameterless, the proposed algorithm is a general circle/ellipse detector that can be applied as is, on any kind of images, without the need of parameter tuning.

An initial version of the detector, which targeted only the detection of line segments and circular arcs, was published in [Pătrăucean 2011].

Notations

- Random variables will be denoted by majuscules, whereas the common variables will be denoted by minuscules.
- A digital image x of size $n \times m$ pixels, is a function defined on a grid $\Gamma = [1, n] \times [1, m] \subset \mathbb{N}^2$, with values in $\mathbb{R}$.
- N_t denotes the total number of possible candidates (*tests*) in an image x of size $n \times m$. It

depends on the geometrical structure we are interested in.

- The background model will be denoted by $\mathcal{H}_0$.

Chapter organisation. This chapter is organised as follows. In section 4.2 we glance through classic approaches used to address the primitive detection task. Section 4.3 introduces the core ideas of the *a contrario* approach, proposed by Desolneux et al. In section 4.4, we position this approach into a general statistical framework, namely the multiple hypothesis testing. In section 4.5 we detail the *a contrario* principles in the problem of line segment detection, as they apply seamlessly to the circle/ellipse detection. Section 4.6 presents the line segment detector proposed by Grompone von Gioi et al. [Grompone von Gioi 2010], which inspired our detector. The proposed circle/ellipse detector is introduced in section 4.7. In section 4.8, the main concept of the *a contrario* approach, namely the number of false alarms, is discussed within the model selection problem. Section 4.9 analyses the *a contrario* model proposed by Desolneux et al. Finally, experimental results and conclusion are given in sections 4.10 and 4.11 respectively.

4.2 A Glance on Primitive Detection Methods

A central problem in computer vision is the extraction (detection) of predefined *geometric primitives* from *geometric data*. A geometric primitive is a curve or surface that can be described by an equation with a number of free parameters; here, we are interested especially in ellipse detection. Geometric data are an unordered list of points in two- or three-dimensional Cartesian space. Such data are obtained through a variety of methods, either directly from images or by using an edge detector in the 2D case, or from stereovision in the 3D case. The detection of geometric primitives[1] could be a prerequisite to solving other higher-level problems, like pose determination, object recognition, or 3D scene reconstruction.

Existing geometric primitive detection algorithms can be roughly classified into two categories: *Hough-based* and *edge chaining* methods. Most of them operate on edge maps, obtained using an edge detector [Canny 1986].

Hough-Based Ellipse Detection Methods

Probably the most popular method for primitive detection is the generalisation of the Hough

[1] Although it might seem futile, we stress here the difference between *primitive detection* and *primitive fitting*, as the frontier is not clear in some cases, creating confusion. The previous chapter addressed the problem of primitive fitting in digital images, which consists in constructing a curve that fits well to a series of data. Thus, in primitive fitting problems, one assumes that the input data are known and the only concern is to find the parameters of the curve [Sampson 1982, Fitzgibbon 1999, Chernov 2005, Kanatani 2008]. A primitive detection problem, on the other hand, must take in charge several tasks, from extracting the geometric data and finding the parameters of the curve, to validating the fitted curve as a detection. Actually, in our work, we consider that a primitive detector implies two important steps: primitive candidate selection and primitive candidate validation. Thus, for the detectors using a primitive fitting stage, the latter is comprised in the candidate selection step of the primitive detection problem.

transform (HT) [Duda 1972], proposed by Ballard[2] (GHT) [Ballard 1987], used together with some detection thresholds. GHT is not directly applicable to images, but it can be applied to their edge map, obtained by means of an edge detector, e.g. Canny detector [Canny 1986]. Given a set of edge points, GHT computes the quantity of points on each presumed primitive. Then all primitives where an "excess" of points is observed, supposedly contain alignments. The basic idea of GHT is to quantise the transformation space into D-dimensional cells, obtaining the so-called (grid) *accumulators*. Each transformed image point T_i is quantised, and then votes for one of these cells. Cells whose number of votes yields a local maximum (*spike*) in the accumulator space and that is superior to some threshold, will be considered as valid detections.

A major advantage of GHT is its relative insensitivity to imperfect data, that is, noise and occlusions can be handled, up to a certain level. However, in practice, this approach is computationally inefficient, as the accumulators may become very large, and the execution time is important. E.g. in circle detection, one uses a 3D accumulator (a circle is determined by three parameters: centre coordinates and radius), whereas to detect ellipses, a 5D accumulator is needed (five parameters determine an ellipse: centre coordinates, axes and orientation). Another sensitive, yet crucial aspect of GHT is the choice of the quantisation precision. Most of the time, the cell size is fixed considering *ad-hoc*, problem specific arguments. The direct consequence of the quantisation choice is reflected in the number of false detections. Indeed, if quantisation is too fine, cells will not have enough votes, and correct instances will be missed (false negatives). On the other hand, a very coarse quantisation increases the likelihood of "excesses" occurring at random (false positives). This brings to light the importance of the validation phase, i.e. the importance of the detection thresholds used together with GHT in order to declare an "excess" of points as a valid detection.

Many methods have been proposed to overcome these drawbacks and two different reasonings can be identified: *one-to-many* and *many-to-one* Hough-based procedures. In a one-to-many approach, an edge point votes for several possible primitives, e.g. in a line detection problem, an edge point votes for all the 180 lines that could pass through that point, considering a precision of one degree for the line slope. GHT belongs to this class. In a many-to-one approach, several edge points vote for one primitive passing through those points. Either approach used, a few main improvement directions could be enunciated:

1. methods targeting exclusively the execution time and the memory consumption;

2. probabilistic methods addressing the problem of detection thresholds used to declare a detection as valid;

3. methods using additionally the gradient orientations to reduce the clutter in the voting space.

The exponent of the first category is the *randomised HT* (RHT) [Xu 1990, McLaughlin 1998], which is different from GHT in that it combines voting with geometric properties of analytical

[2] GHT was mentioned in chapter 3 also as a fitting technique. Indeed, within a Hough-based approach, the extraction of the geometric data from an image and the computation of the primitive parameters are indistinguishable. Thus, GHT carries out the fitting task, but accomplishes also the geometric data extraction, specific to detection methods. Hence, it is common to encounter references to GHT in both problems.

curves. *RHT* takes advantage of the fact that some analytical curves can be fully determined by a certain number of points on the curve. For example, an ellipse (or a circle) can be determined by three points, as detailed in section 3.6. *RHT* uses an iterative stochastic process, with a fixed number of iterations, and, at each iteration, the algorithm randomly chooses three edge points, which will vote for the ellipse passing through the three points. The accumulator array in this case is not anymore a D-dimensional (grid) accumulator, but rather a variable-length list, where each entry contains the parameters of an already voted ellipse, and a score. With a newly estimated ellipse, if a similar[3] ellipse is already present in the accumulator array, the corresponding entry is updated to the mean of the two ellipses and the score is incremented by one; otherwise, a new entry is added to the accumulator array, with the parameters of the new ellipse and score one. *RHT* is, thus, a many-to-one approach. Another approach in the same class is to decompose the parameter space into smaller subspaces, and the *fast HT* (*FHT*) [Li 1986] goes in this direction. It gives considerable speed-up and reduces memory requirement. Instead of dividing the parameter space uniformly into blocks, the *FHT* "homes in" on the solution, ignoring areas in parameter space relatively devoid of votes. This is achieved by considering a tree-like accumulator, where only the voted blocks get to have descendants. *FHT* is a variant of the one-to-many approach.

The methods in the second class are one-to-many *probabilistic HT* (*PHT*), that attempt to reduce the execution time and the memory requirements by cutting-down the amount of edge points used in the voting procedure [Kiryati 1991]. Intuitively, this works because a random subset of the edge map will fairly represent all features and noise present in an image, provided the chosen points are distributed conveniently in the image. Suitable detection thresholds need to be used in order to take into account the reduction of votes. Obviously, choosing a low percentage of edge points will lead to a faster algorithm, but clearly, it risks to fail in detecting smaller features. In [Kiryati 1991], the detection thresholds are fixed beforehand, requiring prior knowledge on the image content, e.g. number of primitives to be detected or their size. Therefore, each time the image changes, these thresholds need to be tuned. Matas et al. address this drawback, by proposing a *progressive PHT* (*PPHT*) [Matas 1998], where the authors model the edge map through a noise statistical model. Afterwards, a peak in the transformation space will be accepted as a valid detection only if it contains a number of votes that could not correspond to an accidental accumulation of votes. Although it improves *PHT* in that it does not require prior knowledge on the image content, the Achilles heel of this procedure is the lack of scalability: the detection thresholds are computed such as to guard against false detections in an image of 256×256, but the control of false detections is no longer ensured when the image size grows.

Finally, in the third class we could include all one-to-many methods mentioned above, with the specification that the voting procedure is done in a restricted manner. Assuming that the local gradient of the image intensity will necessarily be orthogonal to the edge, a point will vote only for the primitives orthogonal to its gradient direction, up to a certain precision. This idea was first proposed in [O'Gorman 1976]. It reduces the computation time and has the interesting effect of reducing the number of useless votes, thus enhancing the visibility of the spikes corresponding to real detections in the image.

Although many research works addressed the weaknesses of *GHT*, there remain some critical

[3] The similarity is measured using convenient metrics, e.g. the Euclidean distance, together with a threshold.

issues. First, *GHT* is applied on binary images, requiring the use of an edge detector to obtain them. The parameters of the edge detector need to be set systematically, when the image (type) changes. Second, the detection thresholds, closely related to the quantisation precision, are difficult to tune, and they provide no formal guarantee concerning the number of false detections. The proposed ellipse detector, based on the *a contrario* approach, is animated by the same idea as *PPHT*, namely it evaluates the probability that an event occurred by accident. The major difference, however, is that the detection thresholds are computed automatically, as a function of the image size, guaranteeing scalability and formal control of the number of false positives.

Edge Chaining Ellipse Detection Methods

A second important direction in primitive detection focuses on various techniques of pixels chaining. Most often than not, these detectors operate on edge maps. These algorithms usually extract the geometric data by starting with a point or a group of points, and subsequently, other points are progressively added, provided they obey some geometric properties of the sought primitive. This type of methods appear especially in applications that require a real-time execution, where the Hough-based methods are rather unsuitable. Eventually, some detection thresholds are used to accept or reject a given candidate.

Within this class, a first target could be the determination of polygonal approximations of planar curves, by considering chained lists of edge points, which are subsequently segmented using different techniques, such as segmenting at points of high curvature (vertices) or at points which are at maximum distance from a fixed representation (e.g. a line passing through the ends of the curve) [Urs 1972, Lowe 1987]. Possessing a polygonal approximation, the neighbour linear segments can be joined in order to fit higher-order representations: circular arcs [Etemadi 1992], conic sections or splines [West 1992, Kim 2002, Nguyen 2009]. This two-stage chaining technique proves to be useful in algorithms targeting the image vectorisation, used for example in applications of symbol recognition [Elliman 2002]. Once in possession of a set of chained pixels, the sought primitive is fitted using either some deterministic operators (e.g. based on least-squares) [Kim 2002, Chia 2011], or some randomised robust *RANSAC*-like approaches [Mai 2008]. Section 4.10 presents comparative results between the proposed algorithm and Etemadi's work [Etemadi 1992][4].

Other algorithms use iterative chaining techniques that seek directly higher order representations (circular or elliptical primitives especially). An initial set of points is used to fit the sought curve, and subsequent iterations will add other points that respect the geometric properties of the curve, while updating systematically the fitted curve [Kanatani 2004].

Although more efficient in execution time and memory consumption, the edge chaining detectors are very sensitive to the choice of the detection thresholds and of the edge detector parameters.

In conclusion, even if so many research works addressed the problem of primitive detection in images, it cannot be considered as closed until a sound mathematical background is settled.

[4] The source code of Etemadi's algorithm [Etemadi 1992] can be found at `http://hpux.connect.org.uk/hppd/hpux/Physics/ORT-2.3/`.

In the spirit of this conclusion, the present chapter tackles to some extent the reliability of the detected primitives and the handling of conflicting interpretations. The proposed detector uses the statistical fundamentals of the *a contrario* approach, to continue the formalisation work of the primitive detection in images pioneered by Desolneux et al. [Desolneux 2000]. We are interested namely in extending their work from line segment detection to circular/elliptical shapes, while attempting to handle the conflicts between different interpretations.

4.3 The *A Contrario* Approach

The goal of the *a contrario* approach, as intended by Desolneux et al. [Desolneux 2007], is to give a general framework that would allow designing computer vision algorithms free from parameter tuning and applicable per se to any kind of image, regardless of its source/content/size. Within this chapter, we will focus on the problem of structure detection in images, which can be stated as follows.

Problem 1 (*A Contrario* Structure Detection) *Given a set of features in a digital image, automatically compute a family of detection thresholds that allows reporting the meaningful feature arrangements.*

The parameter tuning is an essential feature of every computer vision algorithm, as the accuracy of the results is highly dependent on it. In particular, the detection thresholds have a direct influence on the number of false detections and their values need to be systematically adjusted each time the (type of the) image changes (see figure 4.1).

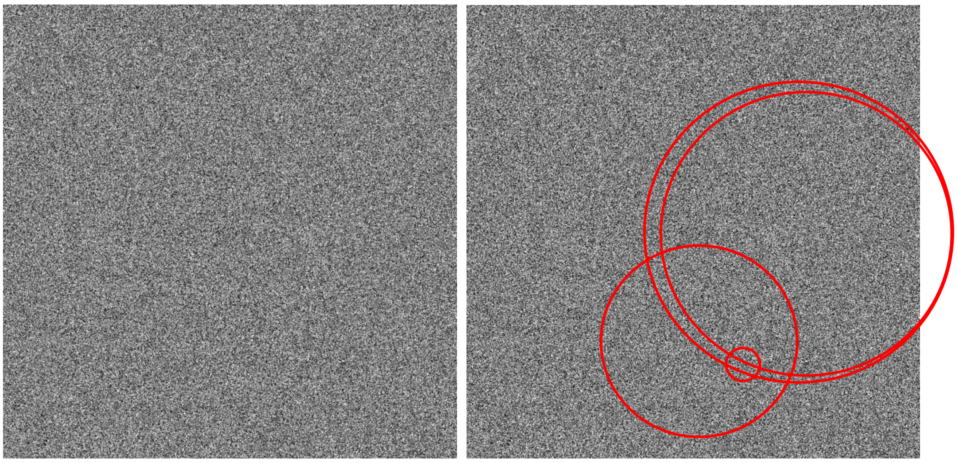


Figure 4.1: *Circle detection on a noise image (*512×512 *pixels) using the Hough-based algorithm available in the opencv library (*`www.opencv.com`*), with different parameter tuning: when the circle radii are limited to* $20 - 50$ *pixels, no detection is reported (left figure), whereas when this parameter is not set (the default range is between* 1 *and image_width), several false detections are reported (right figure).*

With regard to problem 1, the *a contrario* approach seeks to give an automatic technique for defining what is "enough structured" to be "meaningful", and hence for tuning the detection thresholds, by formalising the so-called *Helmholtz principle*. According to it, perceptual entities are feature arrangements, with a certain "degree of structuredness", that could not occur in *a (pure) noise image*[5]. Lowe was the first to formalise this perception principle in computer vision purposes [Lowe 1985][6]. Indeed, the equivalent form of this principle, i.e. *no structure should be perceived in noise*, gives a starting point in tackling the detection thresholds issue. In practice, it states that a computer vision algorithm applied on a noise image should report *no detection*, as such detections are false positives. Within such algorithms that control the number of false positives, two types of detection thresholds should be considered.

- First, the expected overall detection result on a noise image should be fixed: formally it controls the expected number of structure detections in a noise image. Its value should not be zero, as one must take into account that even in a noise image there exists the (small, but non-zero) probability of having accidental occurrences of structures.

- Second, if we call *candidate* a feature arrangement, the detection thresholds applied to each individual candidate must reject any candidate that is likely to occur in a noise image.

As proven by Desolneux et al. [Desolneux 2000], in the *a contrario* approach, the detection thresholds are linked in a way that relates the "degree of structuredness" of a candidate to the global expected result, through the number of possible candidates. More precisely, let $\mathcal{H}_0$ be the (pure) noise image model, also called the ***a contrario* model**. A function $k_x(s)$ is defined to measure the degree of structuredness of a candidate s in the image x. Eventually, this measure assigns a candidate score on which we assess the probability that the candidate s observed in the image x could appear accidentally in an image X (of the same size as x), drawn from the $\mathcal{H}_0$ model. The smaller the probability, the less probable the candidate is to be due to chance; thus it is *meaningful*.

Number of False Alarms

Let us call *candidate* a set s of pixels. We associate to the candidate s a degree of structuredness $k(s)$ which is defined as the number of pixels in s satisfying some given property, e.g. the number of pixels having the same gradient orientation. We will assume that the image geometry of a candidate can be modeled by a set of parameters which, once at least one parameter is fixed, defines a *type* (e.g., in the case of rectangles, we can identify as a type, all rectangles of a given length).

Let $k_X(s)$ be the random variable associated to the degree of structuredness of candidates in image X drawn from $\mathcal{H}_0$, which have the same type as some candidate s in image x. The probability that a candidate s in image x with quantity $k_x(s)$ as degree of structuredness is "due to chance" will be given by

$$\mathbb{P}_{\mathcal{H}_0}[k_X(s) \geq k_x(s)] \tag{4.1}$$

which is the probability of observing under $\mathcal{H}_0$ a candidate (in X) *at least* with the same degree of structuredness as the observed candidate s.

[5] The term "noise" here refers to "unstructured" in a general sense. Each particular problem gives its own precise definition of noise, depending on its goal.

[6] The Helmholtz principle, as described by Lowe: *"In other words, it is the degree to which some relation is unlikely to have arisen by accident which is the most important contributor to its significance"* [Lowe 1985, p. 27].

The essential concept of the *a contrario* approach is the so-called *number of false alarms*. Let s be a candidate in image x with $k_x(s)$ as degree of structuredness. The **number of false alarms** (*NFA*), predicted from the candidate s, is estimated as the quantity

$$NFA(s) = N_t \mathbb{P}_{\mathcal{H}_0}[k_X(s) \geq k_x(s)], \tag{4.2}$$

which is the probability (4.1) multiplied by the number of possible candidates N_t, of all types. Thus, $NFA(s)$ is the estimated number of candidates *of any type* at least as structured as s in X under $\mathcal{H}_0$.

By definition, one says that the candidate s is **ε-meaningful** if its *NFA* is less than ε:

$$NFA(s) = N_t \mathbb{P}_{\mathcal{H}_0}[k_X(s) \geq k_x(s)] \leq \varepsilon, \tag{4.3}$$

that is if the probability (4.1) of s being due to chance is less than ε/N_t. This equation presents the test to be performed in order to decide if the candidate s is valid or its occurrence is due to chance. The smaller the *NFA*, the more meaningful the candidate is.

Now, we will give the important result in the *a contrario* approach. We warn the reader that the notations might get somehow confusing, as in the following we will use the same denomination "number of false alarms", to denote a different quantity. The distinction is to be made through the parameter involved in the two notations: s for $NFA(s)$ defined in (4.2) and ε for $NFA(\varepsilon)$, which will be defined in the sequel.

Result 1 *Let $1_{NFA(s_i)\leq\varepsilon}$ ($i \in \{1,..,N_t\}$) denote the indicator function of the event "the candidate s_i is ε-meaningful". The expectation of the number of ε-meaningful candidates per image under the $\mathcal{H}_0$ model, denoted by*

$$NFA(\varepsilon) = \mathbb{E}_{\mathcal{H}_0}\left[\sum_{i=1}^{N_t} 1_{NFA(s_i)\leq\varepsilon}\right] \tag{4.4}$$

satisfies

$$NFA(\varepsilon) \leq \varepsilon. \tag{4.5}$$

This result reflects the setting up of the Helmholtz principle. The number of detections in a noise image is controlled by the threshold ε, which is the only parameter to tune and can be made as small as desired. The proof of the above result can be found in [Desolneux 2007, p. 71] or in [Grompone von Gioi 2008]. We recall it below because of its importance. We give the latter version for its compactness.

Proof.

$$NFA(\varepsilon) = \mathbb{E}_{\mathcal{H}_0}\left[\sum_{i=1}^{N_t} 1_{NFA(s_i)\leq\varepsilon}\right] = \sum_{i=1}^{N_t} \mathbb{P}_{\mathcal{H}_0}[NFA(s_i) \leq \varepsilon]. \tag{4.6}$$

To ease the writing, let us denote by $Y(s)$ the quantity $\mathbb{P}_{\mathcal{H}_0}[k_X(s) \geq k_x(s)]$ from (4.2). Thus

$$NFA(s_i) = N_t Y(s_i). \tag{4.7}$$

As we are interested in the expected number of ε-meaningful detections, the candidates s_i in (4.7) are ε-meaningful, thus (4.3) stands. We have:

$$Y(s_i) \leq \frac{\varepsilon}{N_t}. \tag{4.8}$$

With this notation and knowing that, for every $\alpha > 0$,

$$\mathbb{P}_{\mathcal{H}_0}[Y(s_i) \leq \alpha] \leq \alpha, \tag{4.9}$$

we can rewrite (4.6) as:

$$NFA(\varepsilon) \overset{(4.7,4.8)}{=} \sum_{i=1}^{N_t} \mathbb{P}_{\mathcal{H}_0}\left[Y(s_i) \leq \frac{\varepsilon}{N_t}\right] \overset{(4.9)}{\leq} N_t \frac{\varepsilon}{N_t} = \varepsilon.$$

■

Ergo, in the *a contrario* literature, two quantities are attached to the same concept of the *number of false alarms*, and care must be taken to distinguish them. On the one hand, the *NFA* is used at the individual candidate level —$NFA(s)$ (4.2)— yielding a measure of meaningfulness of a candidate, and on the other hand, at the image level, it represents the overall detection result —$NFA(\varepsilon)$ (4.4). Both quantities are linked to ε, which is the unique detection threshold of the *a contrario* approach. It is the cut-off value on which the meaningfulness of a candidate is assessed. Moreover, as proven in [Desolneux 2007, p. 77], the dependance of the results on ε is logarithmic, thus weak. So a convenient value can be chosen once and for all. Many *a contrario* algorithms use the simple value $\varepsilon = 1$, which is suitable in our problem as well. As the same threshold ε controls the expected number of false positives, with the choice $\varepsilon = 1$, we assume the risk of accepting in average one false positive per image. This is a major difference with respect to classic detection methods: the detection threshold imposed on each candidate *says* something about the overall detection result that one can expect, whereas when using classic methods, (e.g. the Hough transform [Duda 1972]), no such information is available. In particular, here is the main difference between *PPHT*, mentioned in the previous section, and the *a contrario* approach: both methods are interested in the probability (expectation) that a candidate occurred by accident, but *PPHT* keeps this analysis only at the candidate level, missing the essence of the problem, by ignoring the link with the overall detection result.

In conclusion, the *a contrario* approach gives a (nearly) parameterless procedure, that formally guarantees the control of the number of false positives.

A Toy Example: Dot Alignments

Figure 4.2 illustrates a toy example, taken up from [Desolneux 2007, p. 41]. It presents the Helmholtz principle applied to a problem of dot alignments, which can be stated as follows. Given a set of points in a digital image, automatically compute a family of detection thresholds that allows reporting the meaningful dot alignments. According to the Helmholtz principle, any candidate that is likely to occur in a noise image should be rejected.

A dot alignment s will be considered as meaningful if it contains a sufficient number of dots that all fall into a strip of a given width d. The *a contrario* model $\mathcal{H}_0$ in this case is that dot positions are uniform, independent random variables, namely a uniform spatial distribution. A candidate alignment will be accepted as valid when the test (4.3) stands. The average number

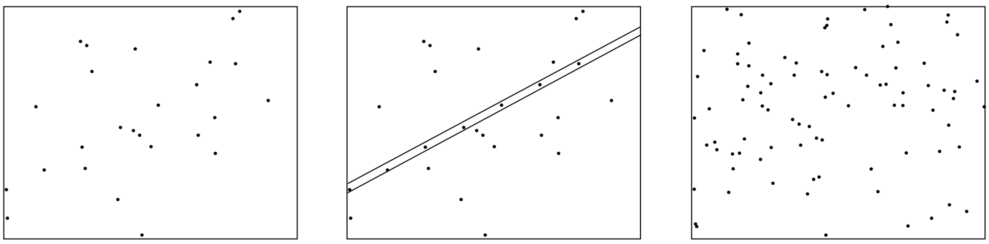


Figure 4.2: *Dot alignment example, illustrating the Helmholtz principle.*

of accepted false detections will be considered $\varepsilon = 1$. The quantities involved in the test, namely the number of possible candidates N_t, and the probability of observing a candidate at least as structured as s in a noise image, must be computed. The number of candidates is given by the total number of stripes of all possible widths whose endpoints are observed dots. If M denotes the number of points, then $N_t = \alpha\frac{M(M-1)}{2}$, with α being the number of considered widths (we choose $\alpha = 10$). The probability that a dot falls into a strip of width d in a $N \times N$ image is estimated as $p_d = \frac{d\sqrt{2}}{N}$, which is the ratio of a strip whose length is the image diagonal and width is d, over the image area. If an observed alignment s contains k dots, the probability of observing, in a noise image, a candidate that contains at least k dots is given by the binomial tail

$$\mathcal{B}(M, k, p_d) = \sum_{i=k}^{M} \left(\begin{array}{c} M \\ i \end{array} \right) p_d^i (1 - p_d)^{M-i}. \tag{4.10}$$

In the given example, the left image of 100 × 100 pixels contains 20 randomly distributed dots to which an alignment of 7 dots was added, yielding $M = 27$ points. If strips of minimum 3 pixels width are considered, then $N_t \simeq 3 \times 10^3$ and $\mathcal{B}(27, 7, 3\sqrt{2}/100) \simeq 10^{-5}$. According to (4.2), the *NFA* associated to this candidate is $\simeq 3 \times 10^{-2}$, which is smaller than 1, therefore the candidate is reported as valid (middle image). The same alignment is not meaningful anymore in the right image, which contains the same 7 dot alignment, along with a random configuration of 80 dots. In this case, the number of tests becomes $N_t \simeq 4 \times 10^5$, thus $NFA(s) \simeq 4$, greater than 1.

This procedure is a mathematical formalisation of the Helmholtz principle, and it attempts to reproduce the human perception. Indeed, in the left image, the candidate is likely to attract attention, i.e. to generate a perception, whereas in the right image, it integrates seamlessly the background, thus no valid detection is reported.

4.4 *A Contrario* Methods and Multiple Hypothesis Testing

Several works on the *a contrario* approach make references to the well-known statistical framework of multiple hypothesis testing (*MHT*) and compare it with their approach [Desolneux 2007, p. 253], [Grompone von Gioi 2008], [Cao 2008, *Preface*]. It is the purpose of this section to further investigate this idea. We will briefly analyse the behaviour and the limitations of the procedures commonly used in *MHT* problems, when applied to problems similar with those targeted by the *a contrario* approach. Eventually, an analogy with the microarray data analysis (*MDA*) problem, a classic case of *MHT* practical application, will be established.

Definitions

We first define some basic notions used in the hypothesis testing domain, partially taken up from [Lehmann 2005]:

- **statistical hypothesis** = a statement about the parameters describing a population (not a sample).
- **test statistic** = a value calculated from a sample, often to summarise the sample for comparison purposes.
- **null hypothesis** = a simple hypothesis associated with a contradiction to a theory one would like to prove.
- **statistical test** = a decision function that takes its values in the set of hypotheses.
- **significance level of a test** (α) = the probability of incorrectly rejecting the null hypothesis. The false positive rate.
- **p-value** = the probability of observing a result at least as extreme as the test statistic, assuming the null hypothesis is true.
- **E-value** = the average number of times in multiple testing that one expects to obtain a test statistic at least as extreme as the one that was actually observed, assuming that the null hypothesis is true. The E-value is the product of the number of tests and the p-value.
- **Type I error** (false positive) = it occurs when a statistical test rejects a true null hypothesis.
- **Type II error** (false negative) = it occurs when the test fails to reject a false null hypothesis.

P-values and Multiple Hypothesis Testing (*MHT*) Problem

The *a contrario* approach proposes a new vision on the detection problem, based on the Helmholtz perception principle. Usually, existing (statistical) methods propose several statistical models for the objects to be detected and for the background. The *a contrario* approach simplifies this scheme, by proposing to use only a background model, and afterwards, the objects of interest are detected as unexpected deviations (outliers) from this model. Assessing the probability that a candidate belongs to the background model comes to deciding if its configuration is reasonably explained by this model, i.e. it is due to chance, or it is an outlier.

Statistically, the background model $\mathcal{H}_0$ represents the null hypothesis. In an image x, there are N_t statistical tests $s_i, i \in \{1, .., N_t\}$, that evaluate the validity of the null hypothesis $\mathcal{H}_0$, having the form:

$$\boxed{\begin{array}{l}\text{reject } \mathcal{H}_0 \text{ if } k_x(s_i) \geq \kappa_{s_i},\\ \text{accept } \mathcal{H}_0 \text{ otherwise,}\end{array}}$$

where $k_x(s_i)$ is a function applied to the candidate s_i from image x to compute its test statistic, and κ_{s_i} is a detection threshold, whose value depends on s_i.

Thus, the null hypothesis should be rejected whenever the probability of observing under $\mathcal{H}_0$ a candidate as structured as s_i (in the sense of $k(\cdot)$), $\mathbb{P}_{\mathcal{H}_0}[k_X(s_i) \geq k_x(s_i)]$, is *very small*, i.e., the p-value associated to the test s_i is less than a chosen significance level $\alpha \in [0, 1]$. The choice of α is crucial, as the overall control of the false positives relies on this threshold.

If there were only one test s, the threshold κ_s should be chosen at the value

$$\kappa_s = \min_{\kappa}\{\mathbb{P}\left[k_X(s) \geq \kappa\right] \leq \alpha\}.$$

With this choice, the probability of accepting a false positive is less than α. As an example, let us consider the simple problem of coin flipping where one wants to decide if a coin is fair or not. The null hypothesis in this case would be that the coin is fair. One might declare that the coin was biased if in 10 flips it landed heads at least 8 times, i.e., $\kappa_s = 8$. It can be shown that this is a relevant choice since the probability that a fair coin would come up heads at least 8 out of 10 times is given by the tail of the binomial distribution $\mathcal{B}(10, 8, 0.5) = 0.0547$. This value can be considered small enough to conclude that the null hypothesis should be rejected whenever a pattern at least as unusual as this one is observed. With this significance level, we have a probability less than 0.0547 to misclassify a given coin, i.e. to reject a true null hypothesis, thus to accept a false positive.

As we are dealing with multiple tests, the *a contrario* approach corresponds to a **multiple hypothesis testing (*MHT*)** problem. In this context, the following quantities are of interest:

1. N_t – the number of tests;
2. V – the number of times $\mathcal{H}_0$ was rejected;
3. S – the number of times $\mathcal{H}_0$ was erroneously rejected;
4. $FWER = \mathbb{P}_{\mathcal{H}_0}[V \geq 1]$ (Family Wise Error Rate) – the probability of accepting at least one false positive;
5. $PFER = \mathbb{E}[V]$ (Per-Family Error Rate) – the expectation of the number of false positives.
6. $FDR = \mathbb{E}[\frac{V}{V+S}]$ (False Discovery Rate) – the proportion of false positives among the validated candidates.

When testing N_t hypothesis on a set of data, the significance level attached to each test has to take into account the number of performed tests, in order to keep under control the false positives. Let us continue the coin flipping example from above to illustrate this idea. A multiple-comparisons problem arises if one wanted to use the aforementioned significance level (which is appropriate for testing the fairness of a single coin), to test the fairness of many coins. Suppose one was to test $N = 100$ fair coins by this method. Given that the probability of a fair coin coming up 8, 9, or 10 heads in 10 flips is 0.0547, the likelihood that all 100 fair coins are identified as fair by this criterion is $p = (1-0.0547)^{100} \approx 0.0036$, and it tends to 0 as N grows. Therefore applying the single-test coin-fairness criterion to multiple comparisons would more likely falsely identify at least one fair coin as unfair, producing false positives. To guard against Type I errors (false positives), the significance level of each test must take into account the number of tests.

Usually, the *MHT* problems focus on controlling the *FWER* value, by computing its upper bound. According to the Šidác correction, if the significance level of each test is set to $1-(1-\alpha)^{1/N_t}$, then it can be proven that the *FWER* is up-bounded by α [Abdi 2007]. This procedure cannot be applied in the *a contrario* context, as the Šidác correction makes an assumption on the independence of the N_t tests, which does not necessarily hold in the problems targeted by the *a contrario* approach, e.g. for line segment detection, a candidate line segment can be contained in another larger candidate line segment. For the general case (the tests are independent or, by the contrary, a high correlation between tests is in evidence), the Bonferroni correction applies, and it states that choosing the significance level of each test at the value α/N_t, leads to a *FWER* up-bounded[7] by α.

Nevertheless, this usage of the Bonferroni correction can become unsuitable when N_t is large and the tests are correlated. The significance level becomes *weaker* when N_t grows. This can result in failing to reject the null hypothesis as often as it should, i.e. producing false negatives (Type II errors). The family wise error rate is up-bounded by α, but in reality, it can be much smaller.

This issue appears for example in bioinformatics, namely in the gene expression analysis problem, which involves monitoring the expression levels of thousands of genes simultaneously under a particular condition. The microarray technology and the *MHT* statistical framework make possible handling the enormous amount of data, while guarding against Type I errors. In the following, we will briefly describe this technology and the underlying statistics in order to establish an analogy with the *a contrario* approach. For further details on this problem, we refer the reader to [Babu 2004, Yang 2003b].

Microarray data analysis (*MDA*) is a relatively recent domain (developed after 1996), that attracted quickly the interest of an important number of biologists and statisticians. A microarray experiment may have various goals, like the study of host genomic responses to bacterial infections [Alizadeh 2000] or identifying tumor subclasses [Golub 1999]. Regardless of the goal, a common question in the microarray experiments is the identification of differentially expressed genes, that is, genes whose expression levels are associated with a factor of interest, e.g. a certain treatment.

"A microarray is typically a glass slide on to which DNA molecules are fixed by a robot in an orderly manner at specific locations called spots (or features). A microarray may contain thousands of spots and each spot may contain a few million copies of identical DNA molecules that uniquely correspond to a gene (figure 4.3A). The most popular application is to compare the expression of a set of genes from a cell maintained in a particular condition (condition A) to the same set of genes from a reference cell maintained under normal conditions (condition B). Figure 4.3B gives a general picture of the experimental steps involved. First, RNA is extracted from the cells and reverse transcribed into cDNA, which is then labelled with different fluorescent dyes.

[7] The proof is obtained by using the Boole's inequality $\mathbb{P}[A_1 \cup A_2 \ldots \cup A_n] \leq \sum \mathbb{P}[A_i]$.

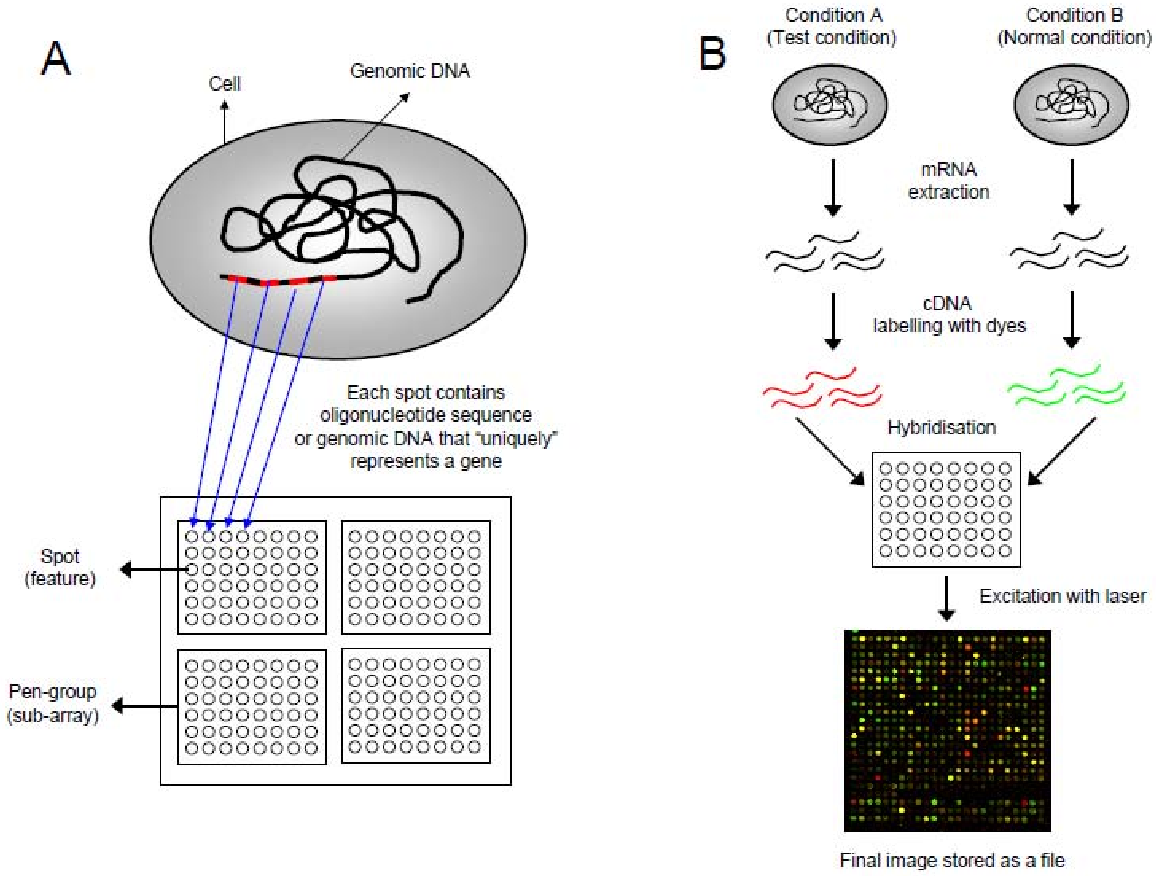


Figure 4.3: *The microarray technology (image reproduced by courtesy of M. Madan Babu).*

For example, cDNA from cells grown in condition A may be labelled with a red dye and from cells grown in condition B with a green dye. Once the samples have been differentially labelled, they are allowed to hybridize onto the same glass slide. At this point, any cDNA sequence in the sample will hybridise to specific spots on the glass slide containing its complementary sequence. Following the hybridisation step, the spots in the hybridised microarray are excited by a laser and scanned at suitable wavelengths to detect the red and green dyes. The amount of fluorescence emitted upon excitation corresponds to the amount of bound nucleic acid. For instance, if cDNA from condition A for a particular gene was in greater abundance than that from condition B, one would find the spot to be red. If it was the other way, the spot would be green. If the gene was expressed to the same extent in both conditions, one would find the spot to be yellow, and if the gene was not expressed in both conditions, the spot would be black. Thus, what is seen at the end of the experimental stage is an image of the microarray, in which each spot that corresponds to a gene has an associated fluorescence value representing the relative expression level of that gene" [Babu 2004].

Once the experimental part is completed, the statistics tools come into play. Their role is to classify the genes associated to spots into two sets: non-differentially expressed genes and differentially expressed genes. This step is similar to the *a contrario* reasoning. To estimate the two sets, an *MHT* framework is settled. First, a set of null hypothesis $\mathcal{H}_{0j}$ is defined, where $j = \{1, .., N_s\}$, N_s being the number of spots. Usually, the null hypothesis is that a gene is non-differentially expressed. Rejection decisions of the null hypothesis are based on appropriate test statistics (mean intensity of the spot, variance, etc.). E.g. Reject $\mathcal{H}_{0j}$, i.e., declare gene j differentially expressed, if $T(j) > c_j$, where $c = (c_j), j = \{1, .., N_s\}$ denotes an N_s-vector of cut-off values. Finally, a Type I error rate that corresponds to a suitable form of control of false positives for the particular application (e.g. *FWER*) is chosen.

The analogy between the *MDA* problem and the *a contrario* typical problems resides in the large number of tests [Dudoit 2003, Gordon 2007], that are potentially highly correlated. Indeed, in the *MDA* domain as in the *a contrario* case, no independence hypothesis on the tests can be made, as the spots can contain replicated genes or co-regulated genes [Yang 2003b, p. 64].

In the *MDA* domain, the Bonferroni correction was declared repeatedly as too conservative for practical purposes, because of the considerable gap between the value of α and the actual *FWER* [Yang 2003b, p. 64], [Dudoit 2003]. Thus, its use has highly diminished. Several less conservative procedures that improve the Bonferroni correction were considered (see [Gordon 2007] for a survey) and even new concepts of error rate were studied, such as the *FDR*. These new procedures prove to bring the *FWER* value closer to its upper-bound, with the cost of an increased complexity.

The *a contrario* approach, pioneered by Desolneux, Moisan and Morel in 2000 [Desolneux 2000], could be considered as a novel practical usage of the Bonferroni correction. Even if it was well known that the Bonferroni correction leads to controlling both *FWER* and *PFER*, this procedure had always been perceived only as a *FWER*-controlling scheme in practical applications. The *a contrario* approach focuses on controlling the expectation (*PFER*), and not the probability (*FWER*), as the upper-bound α in this case gains a direct practical significance and can be easily fixed[8]. Namely, it represents the average number of false detections that one could expect (accept) when the *a contrario* framework is in use. Up-bounding an expectation and not a probability, lets to the user the possibility of choosing a convenient, simple value, which is no longer restricted to the interval $[0, 1]$. For example, a value like $\alpha = 2$ means that we assume the risk of accepting in average two false positives per image. The α value equipped with this new significance will be denoted by ε, as in [Desolneux 2000].

Later, in 2007, Gordon et al., analysing the *MHT* framework applied to the *MDA* problem, dedicate an article [Gordon 2007] to clear the Bonferroni correction from the accusation of exceeding conservatism: *"the notorious conservatism of the Bonferroni procedure is a misconception stemming from the traditionally conservative choice of its parameter rather than from any solid evidence of its conservative nature per se".* They show that the results obtained with a *PFER*-controlling Bonferroni correction are similar with those obtained by a more complex *FDR*-controlling procedure, called *BH* [Benjamini 1995]. They stress though the fact that the two approaches serve different purposes: *"If, for example, the practitioner decides that, on the average, he/she can afford two false positives per experiment, then it is natural to use the Bonferroni procedure with the nominal level of the PFER equaling 2. If, on the other hand, the*

[8] If the tests have continuous distributions, the PFER equals α. Otherwise, for discrete distributions, the PFER is up-bounded by α [Grompone von Gioi 2009, Gordon 2007].

practitioner wants, on the average, the proportion of false positives among all positives not to exceed 10%*, he/she can use the BH procedure with the nominal level of the FDR equaling 0.1."* The former case corresponds to the *a contrario* approach, which can be seen as a precursor of Gordon et al.'s conclusion.

Indeed, Desolneux et al. built their decisional framework on the *PFER* control, which is a more appropriate approach for problems with a high number of correlated tests. The *number of false alarms*, defined as the average number of meaningful candidates that one can observe in an unstructured model (4.4) corresponds to the per family error rate, which is controlled at the ε level. The proof of this statement can be given either by using the *a contrario* reasoning, based on the Helmholtz principle, or by using the Bonferroni correction, based on the Boole's inequality.

According to the *PFER*-controlling Bonferroni correction, the control of the number of false positives at the ε level is ensured if each test $s_i, i \in \{1, .., N_t\}$ rejects the $\mathcal{H}_0$ hypothesis whenever the p-value of the test is less than ε/N_t, i.e.

$$\mathbb{P}_{\mathcal{H}_0}[k_X(s_i) \geq k_x(s_i)] \leq \varepsilon/N_t. \tag{4.11}$$

The number of false alarms of a given candidate defined in (4.2) results immediately from (4.11) and it represents the p-value of the test, multiplied by the number of tests. Thus, it corresponds to the E-value of the test; it was also called a *corrected p-value* in [Grompone von Gioi 2008].

The parallel between the *a contrario* approach and the *MHT* framework stands when comparable target problems are addressed, namely problems that involve a large number of tests and interdependence between the p-values of the tests, like the *MDA* problem, which is a very active research domain. Currently, the interest of statisticians working in this domain concerns *complete* procedures, which allow guarding against Type I errors and Type II errors [Rice 2010]. The parallel might be useful inasmuch as the advancements of one domain might represent future research leads for the other.

A contrario vs *MINPRAN*

In the same context of statistical hypothesis testing, is worth mentioning the link between the *a contrario* approach and the *MINPRAN* method, proposed by Stewart [Stewart 1995], which is discussed in more details in [Desolneux 2007, p. 87]. *MINPRAN*'s author introduced the method as a new paradigm, and he applied it to the 3D alignment problem. More precisely, Stewart tackled the problem of detecting 3D point alignments by considering the probability of the event: "at least k points among the N, randomly fall at a distance less than r from the plane P". Essentially, the reasoning is the same as in the *a contrario* approach: we are interested in measuring how probable is an event to be due to chance. However, there are some *technical* differences which deserve attention. *MINPRAN* focuses on the probability that the aforementioned event occurs by chance, and makes a rather unrealistic hypothesis on the independence between the considered tests. Moreover, it requires a particular parameter setting, e.g. the estimated maximum fraction of outliers, the minimum number of points to be allowed in a fit and the estimated maximum number of correct fits. The *a contrario* approach overcomes these drawbacks, by using the expectation of the event instead of its probability, and by using the Shannon's principle in order to bypass the parameter tuning.

4.5 Line Segment Detection

The previous sections introduced the general *a contrario* framework. In this section, we will particularise it for the line segment detection problem, as the same principles stand for any kind of feature, including circular/elliptical arc detection, the object of our study.

Two main ingredients are required by the *a contrario* technique: a measure function giving a score to each candidate which enables assessing its degree of structuredness, and a background model. The former is used to decide if a candidate fits well the defined background model, or it is an outlier. To this end, Desolneux et al. decided to give importance to the orientation of the gradient, rather than to its magnitude, following Wertheimer's contrast invariance principle: *"Image interpretation does not depend on actual values of the gray levels, but only on their relative values."* [Desolneux 2007, p. 5].

Background model

The background model $\mathcal{H}_0$ proposed by Desolneux et al. [Desolneux 2000] for the problem of primitive detection (line segments, ellipses etc.) is a gradient field where the gradient orientations associated to each pixel are i.i.d. random variables. Thus, a random image X of $n \times m$ pixels, $X : \Gamma = [1, n] \times [1, m] \rightarrow \mathbb{R}$, drawn from $\mathcal{H}_0$, is a random image such that:

1. $\forall p \in \Gamma, Angle(\bigtriangledown X(p))$ is uniformly distributed over $[0, 2\pi]$.
2. $\{Angle(\bigtriangledown X(p))\}_{p\in\Gamma}$ are independent random variables.

These two assertions hold in a Gaussian white noise image, i.e. $X(i,j), i = \{1,..,n\}, j = \{1,..,m\}$ are Gaussian random variables with mean μ and variance σ^2, under some condition on the sampling [Desolneux 2007, p. 67], discussed in section 4.9. Figure 4.4 presents an example of a Gaussian white noise image, together with its gradient field. We use the same background model in our work.

Meaningfulness assertion – *NFA* discrete formulation

A pixel $p \in \Gamma$ is said to be δ-*aligned*, or simply *aligned*, with an oriented line segment s up to a precision δ if

$$Angle(\bigtriangledown x(p), dir_\perp(s)) \leq \delta\pi, \tag{4.12}$$

where $\bigtriangledown x(p)$ is the gradient of the image x in p and $dir_\perp(s)$ is a vector orthogonal to s (see figure 4.5).

The equation (4.12) is necessary to introduce the function used to measure the degree of structuredness of a candidate. The probability that a candidate belongs to the background model is assessed by counting the number of aligned pixels that it contains, as shown in figure 4.5. This corresponds to the discrete formulation of the *NFA* attached to a candidate. A

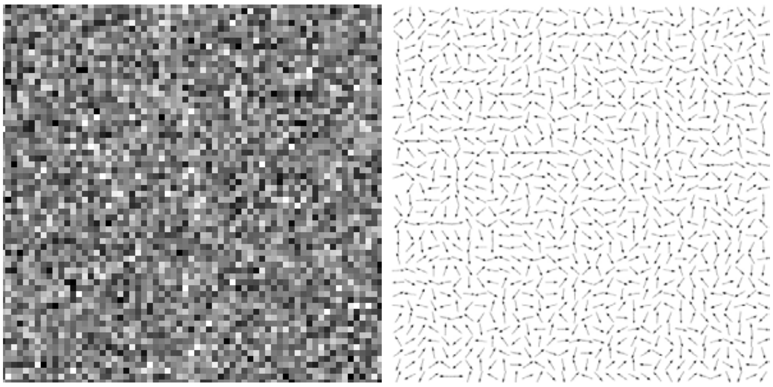


Figure 4.4: *Example of Gaussian noise image and its gradient orientations*

continuous formulation, as suggested by Grompone von Gioi et al. in [Grompone von Gioi 2009] will also be experimented in section 4.8.

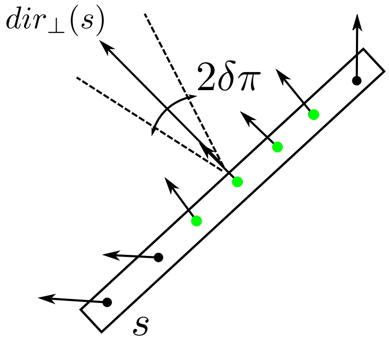


Figure 4.5: *A candidate line segment* s *of length* 7*, containing* 4 *aligned pixels, marked in green.*

Formally, let s be a straight line segment of length l, containing l independent points $x_1, x_2, \ldots, x_l$. To each point $x_i, i = \{1, .., l\}$, a random variable X_i will be attached such as

$$\begin{cases} X_i = 1, & \text{if } x_i - aligned \\ X_i = 0, & \text{otherwise.} \end{cases} \tag{4.13}$$

The probability that a point x_i is δ - aligned with the line segment s is $\frac{2\delta\pi}{2\pi} = \delta$. Thus, the random variables X_i have a Bernoulli distribution of parameter δ: $\mathbb{P}[X_i = 1] = \delta$ and $\mathbb{P}[X_i = 0] = 1 - \delta$.

Let S_l be the random variable representing the number of aligned pixels that the candidate s of length l contains:

$$S_l = X_1 + X_2 + \ldots + X_l.$$

If the independence hypothesis on x_i holds, the law of S_l is given by the binomial distribution

$$\mathbb{P}[S_l = k] = \left(\begin{array}{c} l \\ k \end{array} \right) \delta^k (1-\delta)^{l-k}.$$

As the number of false alarms attached to a line segment s counts for the average number of candidates at least as structured as s under the null hypothesis $\mathcal{H}_0$, the *NFA* computation comes to estimating $\mathbb{P}[S_l \geq k]$, which is given by the binomial tail of parameters l, k, δ:

$$\mathcal{B}(l,k,\delta) = \sum_{i=k}^{l} \left(\begin{array}{c} l \\ i \end{array} \right) \delta^i (1-\delta)^{l-i}. \tag{4.14}$$

Thus, the *NFA* of a candidate line segment s of length l, containing k δ-aligned pixels, belonging to an image x of size $n \times m$ pixels, is:

$$\mathit{NFA}(s) = N_t \mathcal{B}(l,k,\delta). \tag{4.15}$$

where $N_t \sim O(nm)^2$, as each pair of pixels $(p_1, p_2) \in x$ can represent the ends of a line segment.

A candidate line segment s will be called ε*-meaningful* as soon as

$$\mathit{NFA}(s) \leq \varepsilon. \tag{4.16}$$

As mentioned in the previous section, a convenient value can be fixed once and for all for ε, as the results have a logarithmic ε-dependence. In practice, $\varepsilon = 1$ is satisfactory. Thus, a candidate line segment s will be accepted if

$$\mathit{NFA}(s) \leq 1. \tag{4.17}$$

4.6 The *LSD* Algorithm

The problem of feature detection in images should be considered as a two-stage process: first some feature candidates are identified using different heuristics, and afterwards, each candidate has to pass a validation phase. Obviously, in the first phase it is important to avoid introducing *false negatives* (an object is present, but it is not detected). Afterwards, considering the application needs, the validation phase has to minimise either the number of *false positives* (no object is present, but the algorithm detects one), or the number of false negatives. Although for the former step there exist techniques widely accepted (e.g. Hough transform [Duda 1972]), the validation step continues to represent a challenge for the computer vision community, as it usually depends on detection thresholds, which are no longer valid as soon as the (type of the) image changes.

For the line segment detection problem, the *a contrario* approach was originally proposed by Desolneux et al. as a complete detection procedure: the candidate selection was in fact an exhaustive step where each of the N_t possible candidates were considered, and afterwards, the validation was carried out using the test (4.17) [Desolneux 2000]. This ensured that no false negative is introduced by the candidate selection procedure, but in exchange, redundant detections (e.g. on a blurry edge, a bunch of line segments were detected instead of a single one) and misinterpretations (two collinear line segments were interpreted as a single longer line segment containing both) harmed the execution time and the precision of the method.

An efficient line segment detector, called *LSD*, was proposed by Grompone von Gioi et al. [Grompone von Gioi 2010]: it uses a greedy heuristic for candidate selection, followed by an *a contrario* validation. We will briefly describe their algorithm, as we will take up the core ideas in our detector.

Line Segment Candidate Selection

The candidate selection procedure of the *LSD* algorithm shares some ideas with the detection method proposed by Burns et al. [Burns 1986]. In opposition to classic methods, *LSD* makes use of the gradient orientations instead of the edge points computed based on gradient magnitude (e.g. Canny points). Candidate line segments are represented in fact by the so-called *support regions*, which cover zones of the image containing pixels whose gradient orientations are similar up to a given precision δ. The support regions are computed in a greedy-like manner: first a seed pixel is chosen, and afterwards, the algorithm searches recursively the neighbours that have the same gradient orientation up to a given precision. An 8-connected neighbourhood is considered. If a neighbour pixel is found as *aligned* in the sense of (4.12), it will be added to the current support region and marked as visited, in order to avoid visiting it again.

The main steps of the selection procedure are:

1. Compute the image gradient and keep both the magnitude and the orientation.

2. Create a *pseudo-ordered*[9] list $\mathcal{L}$ of pixels, so that pixels with a higher magnitude, which are more probable to belong to edges, are considered first as seed pixels.

3. Create and systematically update a status image, indicating which pixels have already been visited.

4. Scan the list $\mathcal{L}$ and propose as pixel seed the subsequent non-visited pixel.

5. Launch a RegionGrow procedure (algorithm 1) with the chosen pixel seed.

6. Compute the rectangle that covers the obtained region.

In algorithm 1, the function *isAligned* (line 5) uses (4.12) to decide if a neighbour pixel is δ-aligned within the current rectangle candidate.

Each time a new pixel is added, the rectangle orientation is updated (line 8) to a pseudo-mean of the orientations of the pixels p_i currently in the region R:

$$\theta_R = \arctan \frac{\sum_i \sin(Angle(p_i))}{\sum_i \cos(Angle(p_i))}. \tag{4.18}$$

[9] For efficiency reasons, the pixels are not effectively sorted, but rather classified, depending on the magnitude of their gradient, into a predefined number of bins [Grompone von Gioi 2010].

This expression is chosen for its efficiency in terms of computational complexity and precision, especially considering the fact that an incremental approach is required i.e., θ_R must be updated at a very low computational cost for each newly accepted pixel.

When no more neighbour pixels can be added, the obtained region is approximated by a rectangle, described through its length, orientation, width and centre (figure 4.6), which will be proposed as candidate for validation.

The parameters that appear in the candidate selection phase, namely ρ and δ are to be considered as *internal parameters* [Grompone von Gioi 2010]: their values can be fixed empirically as they do not influence significantly the detection results. The parameter ρ is a threshold on the gradient magnitude, so that pixels with small gradient are not considered, in order to avoid the errors introduced by the quantisation noise. Desolneux et al. [Desolneux 2002] showed that gray-level quantisation produces errors in the gradient orientation angle. This error is negligible when gradient magnitude is large, but can be dominant for a small gradient magnitude. It can *invent* patterns, producing thus false detections. A positive side effect of this threshold is a speed improvement, by reducing the number of considered pixels. More details on the computation of ρ are given in [Grompone von Gioi 2010], where it is shown that ρ should be considered a function of δ, the precision used to declare a pixel as aligned. For example, if δ is set to the value 1/8, then $\rho = 5.2$. We consider this value in our algorithm.

As for the parameter δ, which appears also later in the *NFA* computation, its influence was addressed by Desolneux et al. [Desolneux 2007, p. 86], and they showed that fixing values too stringent for δ yields no significant improvement on the detection results. By the contrary, it can introduce spurious meaningful alignments, due to the quantisation of gradient orientations. The value $\delta = 1/8$, which corresponds to 8 angle bins of 22.5 degrees each, gives satisfactory results. It was intensively tested by Grompone et al. and used also by Burns et al. Nevertheless, Grompone von Gioi et al. use multiple precision values in their algorithm, for refinement purposes.

```
Input: Seed pixel $p_s$, gradient orientations image $Angle$, status image indicating used
       pixels $used$.
Output: List of connected pixels $R$ sharing the same gradient orientation up to a
        precision $\delta$.
$R \leftarrow p_s$;
$\theta_R \leftarrow Angle(p_s)$;
foreach $Pixel\ p\ in\ R$ do
    foreach $Neighbour\ \overline{p}\ of\ p\ not\ used\ and\ whose\ magnitude\ is \geq \rho$ do
        if $isAligned(\overline{p},\theta_R,\delta)$ then
            Add $\overline{p}$ to $R$;
            Mark $\overline{p}$ as used;
            Update $\theta_R$ according to (4.18);
        end
    end
end
```

Algorithm 1: RegionGrow

A contrario Candidate Validation

The *a contrario* validation has to consider the particularity of the candidate selection procedure, namely a greedy procedure which seeks for rectangular candidates, and not linear segment candidates. As explained in [Grompone von Gioi 2010], in practice, only the support regions found in the candidate selection phase of the algorithm are tested as candidates for a line segment detection. However, this does not imply that one can set N_t equal to the number of support regions effectively tested. These support regions have complex statistics, different from white noise statistics, and we do not have a closed formula for it. Thus, we must be content with the estimate given by counting the number of potential rectangles in the image. Considering a one pixel precision, there are $(nm)^2$ potential oriented line segments in an $n \times m$ image (starting and ending on a point of the grid Γ. If we set to $(nm)^{1/2}$ the number of possible width values for the rectangle, the number of tests becomes $N_t = (nm)^{5/2}$. Similarly, it can be considered that the number of tests has $O((nm)^{5/2})$, as one rectangle has five degrees of freedom[10] (df), illustrated in figure 4.6: rectangle centre coordinates (2 df), rectangle length (1 df), rectangle orientation (1 df), and rectangle width (1 df). This estimation is rough, the exact number depending on the exact precision considered. What is important here is the order of magnitude, which enables *NFA* to adapt to different image sizes.

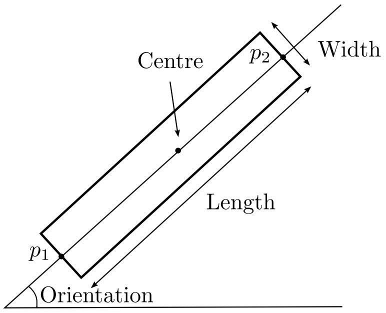


Figure 4.6: *A support region is determined by its centre, length, orientation, and width.*

Thus, the test from (4.17) that each candidate has to pass becomes:

$$NFA_{rectangle} = (nm)^{5/2}\mathcal{B}(l, k, \delta) \leq 1, \tag{4.19}$$

where l represents the area of the rectangle and k is the number of δ-aligned pixels that the rectangle contains. Refer to section 4.9 for details on the distance between the considered points.

4.7 Ellipse Detection

Computer vision literature is rich in ellipse detectors[11]. More often than not, the detection is carried out on binary images, issued from an edge detector, e.g. the Canny detector [Canny 1986].

[10] Each df increases N_t by multiplying it with $(nm)^{\frac{1}{2}}$.

[11] As circles are particular ellipses, their detection can be carried out seamlessly by ellipse detectors. We will use the syntagm "ellipse detector" to refer to the general case (ellipse/circle), and when needed, we will make explicitly the distinction between circle and ellipse detectors.

The declared objectives of the existing procedures target the execution time (real-time execution is desired), and/or the precision of the detected features, especially in cases of partially occluded ellipses [Kim 2002, Kanatani 2004, Zhang 2005, Nguyen 2009].

In general, the existing circle/ellipse detectors do not address explicitly the problem of the false detections (false positives and false negatives). Moreover, in order to obtain a reasonable detection result, in both time and precision terms, a prior parameter tuning is required, to set up some thresholds on the gradient magnitude for the edge detector, together with some limits on the parameters of the expected detections. E.g. the circle detection procedure based on the Hough transform available online in the opencv library[12] requires as input parameters the range of the circle radii, the minimal distance between two circle centres and a minimal percentage of the detected circle arc length, used as validation threshold. Other, non-public ellipse detectors, as the ones cited above, impose by default some relatively permissive values for the parameters of the expected detections and for the detection thresholds, but to reach the desired performance, tighter values need to be considered. According to the tests and results published by their authors, the real-time execution is achieved for images of maximum 400×600 pixels, where only few ellipses are present. Obviously, the parameter tuning needs to be updated when the type of the image changes. This operation is to be carried out somehow *blindly*, as the chosen values give no clue on what one could expect in terms of final result.

In this section we will detail the proposed detector, called *ELSD*. Its goal is to carry out a parameterless feature detection, while formally guaranteeing a control of the number of false positives. *ELSD* can be seen as a generalisation of the *LSD* detector described in the previous section. We aim at simultaneously detecting line segments, circular and elliptical arcs, by implementing a greedy-like candidate selection step, which ensures a reasonable execution time, followed by an *a contrario* validation, which yields an efficient control of the false positives. As it targets several feature types (line segments, circular/elliptical arcs), *ELSD* requires an additional step, namely a model selection phase. Section 4.8 details this problem. The *number of false alarms* will be used as a model selection criterion to decide between possible interpretations: line segment, circular or elliptical arc.

ELSD Candidate Selection

The intended goal of our work is to obtain a parameterless[13] feature detector that minimises the number of false detections. The *a contrario* validation guards effectively against false positives (Type I errors). Concerning the false negatives (Type II errors), if they appear, they are strictly due to an over-restrictive candidate selection procedure, i.e. there exist candidates which do not even get the chance to arrive to the validation phase. Any of the existing ellipse detectors could carry out the candidate selection, if, empirically, their behaviour proves to be not over-restrictive. But, nothing other than the fact that classic detectors operate on binary images (obtained applying an edge detector which needs a rather *critical* parameter tuning), where a significant part of the information contained in the original image is (potentially erroneously) discarded, indicates a constraining behaviour. Another issue, specific to Hough-based

[12] http://www.opencv.com

[13] The parameterless quality refers to the absence of detection thresholds. The parameters that will appear are used only in the candidate selection phase, and their values are chosen as loose (non-constraining) as possible.

detectors, is linked to the quantisation precision, which can generate also false detections. In [Grompone von Gioi 2008], the authors showed that a detector using a Hough candidate selection, followed by an *a contrario* validation, can fail in reporting a satisfying result, due to the improperly set quantisation precision in the candidate selection phase. In this context, *the main feature of the proposed candidate selection is to be free of critical parameters and as permissive as possible, in order to avoid introducing false negatives.*

The candidate selection step in *ELSD* adds a second level of regroupment compared to the *LSD* approach, by simply alternating region growing and region chaining operations (algorithm 2).

1. Region growing: like in *LSD*, rectangular regions are first constructed by aligning pixels in the sense of the definition given in (4.12), cf. algorithm 1.

2. Region chaining: detected regions are chained whenever they obey some loose, elementary constraints, which characterise the elliptical shapes.

Regarding the region chaining, the ends of one detected region are used as seed pixels for subsequent region growing procedures. Thus, the region growing yields a polygonal approximation of a curve through a recursive scheme. Regions that satisfy elementary constraints on elliptical shapes are chained: namely, we impose that the contour described by the chained regions be convex and roughly smooth. Of course, other non-elliptical shapes fulfill these constraints and will be proposed thus as candidates for validation. It is the task of the *a contrario* validation or/and of the *a contrario* model selection to reject them as valid circle/ellipse detections.

We give now more details about the *ELSD* selection procedure. As it extends in a natural way the *LSD* candidate selection, we start with the steps of the *LSD* candidate selection and give subsequently the additional steps, specific to *ELSD*.

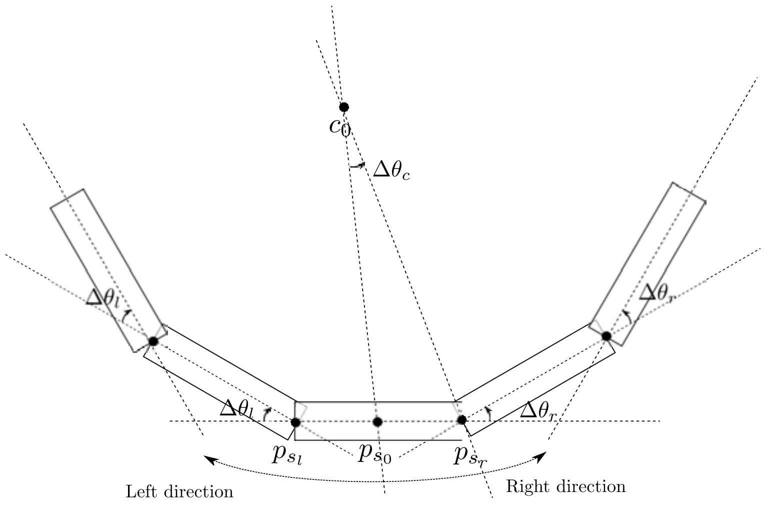


Figure 4.7: *Chaining procedure used by the ELSD candidate selection.*

1. Compute the image gradient and keep both the magnitude and the orientation.

2. Create a pseudo-ordered list $\mathcal{L}$ of pixels, so that pixels with a higher magnitude, which are more probable to belong to edges, are considered first as seed pixels.

3. Create and systematically update a status image, indicating which pixels have already been visited.

4. Scan the list $\mathcal{L}$ and propose as pixel seed p_{s_0} the subsequent non-visited pixel.

5. Launch a RegionGrow procedure (algorithm 1) with the current pixel seed.

6. Compute the rectangle that covers the obtained region and keep it as line segment candidate for the subsequent validation and model selection phases.

   6.1. Compute p_{s_r}, the pixel seed for the right side of the curve (figure 4.8) allowing the subsequent region growing:

   > Let the pixel p_{e_r} represent the right end of the current rectangle[14]. The pixel p_{s_r} is the pixel possessing the maximum gradient magnitude in the neighbourhood of p_{e_r} and whose orientation differs with no more than $\pi/2$ with respect to $\theta_\perp$, the direction orthogonal to the current rectangle. The orientation of the pixel seed is constrained in this way to ensure a loose smoothness of the curve. The neighbourhood size is given by the width of the current rectangle, i.e. the pixel seed will be sought in the square centred at p_{e_r} and of side equaling *width* (figure 4.8).

   6.2. Compute the left pixel seed p_{s_l} in the same manner as above, using the other end of the rectangle.

7. Scan the right direction, by repeating the steps 5, 6 and 6.1., as long as the chained regions describe a convex shape. The convexity constraint is imposed by considering the sign of the variation $\Delta\theta_r$ between the orientation of succeeding rectangles (figure 4.7). This variation is positive when moving in the right direction and negative for the left direction.

8. Scan the left direction, by repeating the steps 5, 6 and 6.2., as long as the chained regions fulfill the convexity constraint, i.e. the variation $\Delta\theta_l$ between the orientation of succeeding rectangles is negative.

9. Compute the circle that fits the gathered pixels, using the variant for circle fitting of the algebraic technique proposed in section 3.5.

10. Compute the ellipse that fits the gathered pixels, using the variant for ellipse fitting of the same algebraic technique.

The steps enumerated above bring together pixels aligned with respect to a roughly smooth convex shape. Eventually, the curve growing procedure returns a list of pixels, together with the two extreme points p_{s_r} and p_{s_l}, and the parameters of the fitted curves. These data are used to delimit the support regions, which will be proposed as candidates in the validation phase. Now, a couple of technical remarks about the procedure described above.

[14] p_{e_r} belongs to the region which generated the current rectangle, i.e. is aligned with the current rectangle, and is the closest to the right side of the rectangle.

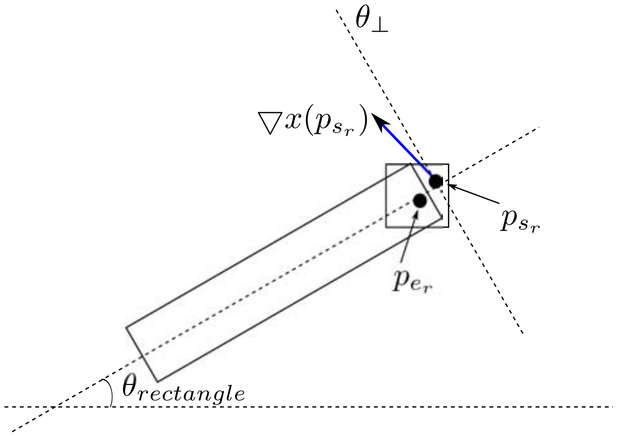


Figure 4.8: *In the neighbourhood of the right rectangle end, represented here by p_{e_r}, the pixel possessing the maximal gradient magnitude and whose orientation differs with no more than $\pi/2$ with respect to $\theta_\perp$ is chosen as new pixel seed p_{s_r}.*

- To carry out steps 6.1. and 6.2. we need to define the right and the left directions for the chaining procedure (figure 4.7); it will also indicate the sign of the angle variation that one should get in case of convex shapes. Actually, it must be done only once at the beginning, when the curve is reduced to a simple (first) rectangle. We perform a rough initial estimation of the centre c_0 of the circle/ellipse, using Forstner's operator [Forstner 1987], described in section 3.5. We are not interested in obtaining a precise location of the centre as we need only the side on which the centre lays, with respect to the first rectangle. Forstner's operator proved to be very efficient in this matter, at a very low computational cost. Hence, the end of the rectangle for which the angular difference θ_c is positive[15] will serve to compute pixels at the right side of the curve, and conversely, the other end will serve to compute pixels at the left side of the curve.

- There are rare cases when the initial estimation of the centre (cf. previous item), fails to return the correct side where the centre lays. This may happen when the initial region is small and/or the orientations of the gradients are highly altered by noise. It is obvious that in this case, the curve growing will stop before succeeding to add any neighbour region. To recover from this situation, we only need to launch again the curve growing procedure, by imposing that the centre is on the other side of the initial rectangle, which leads to interchanging the positions of p_{s_r} and p_{s_l}.

- It is worthy of note that in the case of closed shapes, there is a certain redundancy between steps 7 and 8. Normally, scanning the curve in the right direction should bring us to the initial left pixel seed (the final extreme points coincide), so the step 8 would not be mandatory. However, the pixels are not visited twice, as they are marked as *used* the first time they are visited. This redundancy is useful when a portion of the shape is occluded,

[15] All angles used in this section are oriented angles with respect to the x-axis, measured in counterclockwise sense.

thus the scanning cannot continue on one side. In this case, the scanning on the opposite side will try to continue, if possible, the curve growing.

- Eventually, an additional constraint is considered in order to avoid the spiral shaped contours. In the case of a spiral, the successive regions describe a smooth convex contour, but we choose to stop the curve growing after a complete tour. This is done by imposing that the angle $\Delta\varphi$ between the two extreme points p_{s_r} and p_{s_l} and the initial pixel seed p_{s_0} always lowers (figure 4.9).

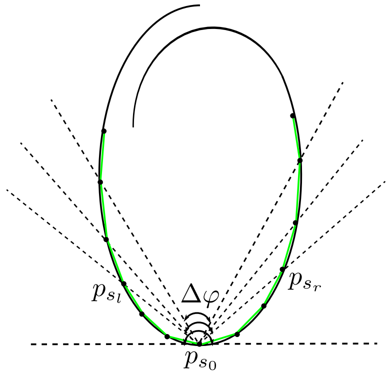


Figure 4.9: *The spiral shapes are avoided by imposing that the angle $\Delta\varphi$ between the extreme points (p_{s_r}, p_{s_l}) and the initial pixel seed p_{s_0} lowers with each new region added to the curve. The initial value of $\Delta\varphi$ is set to π.*

Considering that the circle is a particular ellipse, one may question the usefulness of the two distinct steps 9 and 10, i.e. why the two features (circle, ellipse) are not treated seamlessly at this stage, by fitting only an ellipse and defining its support region. Afterwards, the decision between circle and ellipse could be carried out by the model selection phase. The reason for making the distinction in this early stage resides in the poor performance of the ellipse fitting operators on incomplete data, i.e. when only small arcs[16] ($< \pi/2$) are present. We choose to perform separately the ellipse and the circle fitting, in order to *save* these small arcs. When a small arc is present, it may be either a true circular arc, or an elliptical arc. Most often than not, only the circle fitting will yield a coherent result. This is not very disturbing though, if we accept that an ellipse may be reasonably approximated by circular arcs [Rosin 1998]. The choice of the fitting operator is based on performance and computational complexity reasons. We avoid the iterative (geometric) operators and we use the direct algebraic operator proposed in section 3.5, as it yields the best performance on incomplete data among the existing non-iterative circle/ellipse fitting operators.

A similar chaining procedure is used in Etemadi's work [Etemadi 1992], which targets the segmentation of planar curves into line segments and circular arcs. The main steps of Etemadi's algorithm are: 1) edge detection; 2) edge pixel chaining [West 1992] 3) segmentation of pixel chain into straight segments 4) circle fitting using a least-square technique (section 3.4); 5) model

[16] By "small arc" we refer either to arcs that belong to circles/ellipses with small radius/axes, or to arcs that belong to very large shapes, thus having a small curvature.

selection (line segment or circular arc) using a distance error as criterion [West 1992]. Thus, Etemadi's algorithm cannot be used as a detector per se. A validation step is required. Hence, the main differences between the proposed detector and Etemadi's work are:

- we carry out the pixel chaining directly on the original image, in order to avoid the critical parameters of the edge detector.
- *ELSD* targets also the elliptical shapes, which are of high interest in shape detection.
- *ELSD* uses a validation phase, formalising the control of the false positives (see next section).
- the model selection in *ELSD* case is carried out using a more appropriate criterion – the number of false alarms (section 4.8).

```
Input: List of pixels aligned with the initial rectangle $R_0$, gradient orientations image
       $Angle$, status image indicating used pixels $used$.
Output: List of connected pixels $R$, aligned up to a precision $\delta$, on a roughly smooth,
        convex shape.
$R \leftarrow R_0$;
Compute $p_{s_r}$;
Compute $p_{s_l}$;
$FlagConvex \leftarrow True$;
while $FlagConvex$ do
    $R_r$ = RegionGrow($p_{s_r}$,$Angle$,$used$);
    if convex($R, R_r$) then
        Add $R_r$ to $R$;
        Compute $p_{s_r}$;
    end
    else
        $FlagConvex \leftarrow False$;
    end
end
$FlagConvex \leftarrow True$;
while $FlagConvex$ do
    $R_l$ = RegionGrow($p_{s_l}$,$Angle$,$used$);
    if convex($R, R_l$) then
        Add $R_l$ to $R$;
        Compute $p_{s_l}$;
    end
    else
        $FlagConvex \leftarrow False$;
    end
end
```

Algorithm 2: CurveGrow

A Contrario Circle/Ellipse Validation

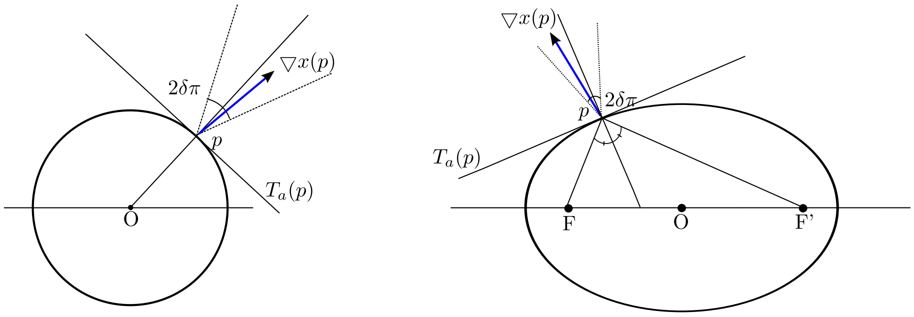


Figure 4.10: *Left: A pixel p is aligned with a circular arc if the support line of its gradient vector $\bigtriangledown x(p)$ has the same direction as the line passing through the centre of the circle O and the pixel p, up to a given precision δ. Right: A pixel p is aligned with an elliptical arc if the support line of its gradient vector $\bigtriangledown x(p)$ has the same direction, up to a precision δ, with the bisector of the angle formed by the lines passing through the focus F and the pixel p, and respectively through the focus F' and the pixel p.*

The reasoning for the *a contrario* validation from *LSD* applies to circular/elliptical arcs, with some specific adjustments, namely the number of candidates and the definition of an aligned pixel. Desolneux et al. anticipated it in [Desolneux 2007, p. 245], but to the best of our knowledge, no efficient implementation of a circle or ellipse detector is reported in the *a contrario* literature.

The goal of this section is to provide a thorough investigation of the *a contrario* validation used in *ELSD*. Under the same background model as in section 4.5, a pixel $p \in \Gamma$ is said to be *aligned* with a circular or an elliptical arc a up to a precision δ if

$$Angle(\bigtriangledown x(p), dir_{\perp}(T_a(p))) \leq \delta\pi, \tag{4.20}$$

where $\bigtriangledown x(p)$ is the gradient of the image x in p and $dir_{\perp}(T_a(p)))$ is a vector orthogonal to the tangent $T_a(p)$ of the circle/ellipse supporting a at p (see figure 4.10).

In fact, in our algorithm implementation, we will use a strictly equivalent definition, which distinguishes the case of a general ellipse and that of a circle, in order to prepare the circular and the elliptical candidates for the model selection phase.

- For a circular arc, a pixel is considered *aligned* if the support line of its gradient vector has the same direction as the line passing through the pixel and the centre of the circle, up to a given precision δ (figure 4.10 left).

- For an elliptical arc, the *aligned* quality of a pixel is assessed by using the focal property of an ellipse, i.e. a pixel is *aligned* if the support line of the gradient vector has the same direction, up to a given precision δ, as the bisector of the angle formed by the lines passing through the pixel and the foci respectively, as shown in figure 4.10 right.

ELSD candidate selection yields circular/elliptical arc candidates which correspond to regions containing *aligned* pixels (in the sense of figure 4.10) up to a given precision, that we will call *circular support ring* and *elliptical support ring* respectively. For circular arcs, the number of possible candidates N_t has an order of $O((nm)^3)$, for an image of $n \times m$ pixels, as for a circular arc we consider its circular support ring, which has 6 degrees of freedom, namely: centre coordinates (2 df), radius (1 df), delimiting angles (2 df) and width (1 df), as illustrated in figure 4.11 left. Similarly, for an elliptical arc, N_t has $O((nm)^4)$. An elliptical support ring has 8 degrees of freedom, being determined by its centre (2 df), axes (2 df), orientation (1 df), delimiting angles (2 df), and width (1 df) (figure 4.11 right).

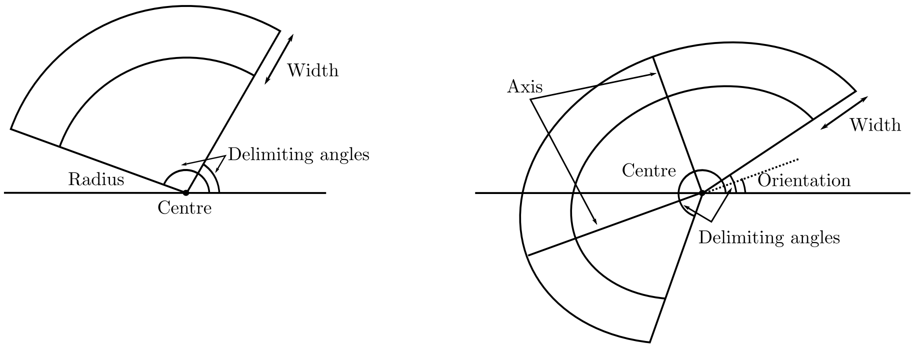


Figure 4.11: *Left: A circular support ring is determined by its centre, radius, delimiting angles, and width. Right: An elliptical support ring is determined by its centre, axes, orientation, delimiting angles, and width.*

The validation of a circular/elliptical arc comes to evaluating the *NFA* of the candidate. If

$$\begin{cases} NFA_{circle} = (nm)^3 \mathcal{B}(l,k,\delta) \leq 1, & \text{for circular arcs} \\ NFA_{ellipse} = (nm)^4 \mathcal{B}(l,k,\delta) \leq 1, & \text{for elliptical arcs} \end{cases} \tag{4.21}$$

the candidate will be declared meaningful and accepted as a valid detection.

In (A.15), l stands for the total number of pixels in the support ring, whereas k denotes the number of *aligned* pixels.

In the sequel we will give some technical details on the validation of the circular/elliptical arcs. *ELSD* candidate selection returns a list R of pixels which are aligned on a roughly smooth, convex shape, and the parameters of the fitted circle $\mathcal{C}$ and of the fitted ellipse $\mathcal{E}$. In the validation phase, we are interested in the number k of pixels that are aligned on the fitted curves (in the sense of figure 4.10), among the total number l of pixels contained in the circular/elliptical support rings that cover the pixels in R. Eventually, the validation phase must return the *NFA* of each candidate, together with its parameters: centre coordinates, radius/axes, delimiting angles, orientation and width. The validation procedure of the circular candidate follows the same steps as the one for the elliptical candidate, but in practice, we have different validation procedures, imposed by the different techniques used to compute the distance point ↔ circle/point ↔ ellipse and the angle of a pixel on a circle/ellipse (figure 4.10). We will give the main steps of the validation procedure, and the differences that appear in each step between the circular and the elliptical validation (algorithms 3,4).

1. Sort pixels in R according to their angular (polar) coordinate, with respect to the centre of the fitted circle $\mathcal{C}$/ellipse $\mathcal{E}$, and compute their delimiting angles α_{start} and α_{end}.

2. Find the maximal distances (towards the exterior d_{ext} and the interior d_{int} of the circle/ellipse) between the pixels in R and the curve, in order to compute the width of the circular/elliptical ring: $width = d_{ext} + |d_{int}|$. For the circular ring, the distances are computed using the Euclidean distance. In the elliptical case, we use the approximation given by Rosin distance, introduced in chapter 2.

3. Perform a new region growing procedure similar with algorithm 1. The pixel seed is the first pixel in R. The goal here is to gather all pixels whose angular positions are in the range $[\alpha_{start}, \alpha_{end}]$, and which are at a distance d with respect to the curve, such that $d_{int} \leq d \leq d_{ext}$. l is incremented each time a pixel fulfilling the two conditions is found. If, moreover, the pixel is aligned with respect to the circle $\mathcal{C}$/ellipse $\mathcal{E}$ (in the sense of figure 4.10), k is incremented as well.

4. Given l and k for each candidate, compute NFA_{circle} and $NFA_{ellipse}$ using (A.15).

```
Input: List $R$ of pixels, circle parameters $\mathcal{C}[c_x, c_y, r]$, gradient orientations image $Angle$,
       $N_{t_c}$.
Output: $NFA_{circle}$, $\mathbf{c}[c_x, c_y, r, \alpha_{start_c}, \alpha_{end_c}, wdth_c]$.
$ang\_coord_c$ = angular_coordinates($R, \mathcal{C}$);
$[\alpha_{start_c}, \alpha_{end_c}]$ = delim_angles($ang\_coord_c$);
$[d_{int_c}, d_{ext_c}]$ = Euclidean_distance($R, \mathcal{C}$);
$pixel_{seed} \leftarrow R[first]$;
$[l_c, k_c]$ = region_grow($pixel_{seed}, \alpha_{start_c}, \alpha_{end_c}, d_{int_c}, d_{ext_c}, Angle$);
$NFA_{circle}$ =nfa($l_c, k_c, N_{t_c}$) ;
```

Algorithm 3: Circular Ring Validation

```
Input: List $R$ of pixels, ellipse parameters $\mathcal{E}[e_x, e_y, a, b, \theta]$, gradient orientations image
       $Angle$, $N_{t_e}$.
Output: $NFA_{ellipse}$, $\mathbf{e}[e_x, e_y, a, b, \alpha_{start_e}, \alpha_{end_e}, \theta, wdth_e]$.
$ang\_coord_e$ = angular_coordinates($R, \mathcal{E}$);
$[\alpha_{start_e}, \alpha_{end_e}]$ = delim_angles($ang\_coord_e$);
$[d_{int_e}, d_{ext_e}]$ = Rosin_distance($R, \mathcal{E}$);
$pixel_{seed} \leftarrow R[first]$;
$[l_e, k_e]$ = region_grow($pixel_{seed}, \alpha_{start_e}, \alpha_{end_e}, d_{int_e}, d_{ext_e}, Angle$);
$NFA_{ellipse}$ =nfa($l_e, k_e, N_{t_e}$) ;
```

Algorithm 4: Elliptical Ring Validation

Figure 4.12 shows the minimum number k_{min} of aligned pixels that a length[17] l circular/elliptical arc belonging to an image of $n \times m$ pixels must contain in order to be meaningful. Here, we assume that the circular/elliptical support rings have fixed width, $width = 1$, thus

[17] Here, the length l of a circular/elliptical arc is the number of points counted considering a Bresenham circle/ellipse representation.

the number of tests becomes $(nm)^{5/2}$ for the circular candidate, and $(nm)^{7/2}$ for the elliptical candidate.

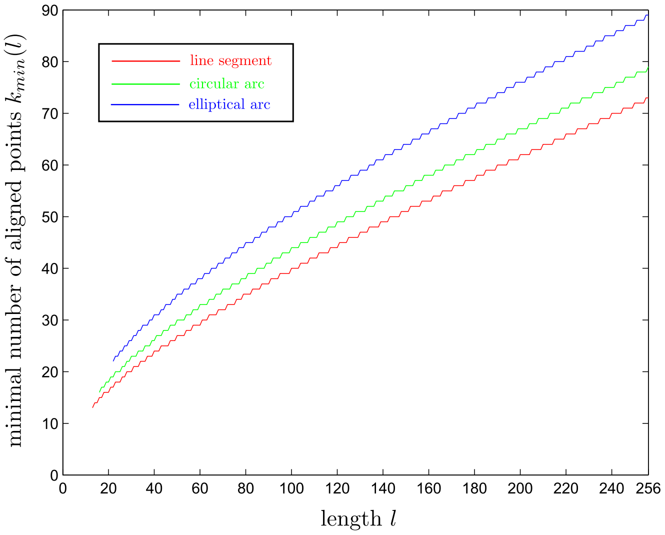


Figure 4.12: *The minimum number of aligned points that a line segment and a circular/elliptical arc must contain to be considered* ε*-meaningful. This result is obtained for an image of* 512×512 *pixels, with* $\varepsilon = 1$ *and* $\delta = 1/8$.

The proof that the number of meaningful circular/elliptical arcs in a noise image is controlled by $\varepsilon = 1$ is straightforward; only the number of candidates changed with respect to line segment validation, in order to take into account the increased complexity of the newly addressed feature types.

The three candidates supplied by *ELSD* candidate selection step (one rectangle, one circular and one elliptical ring) will be proposed to the validation phase, which consists in verifying the test (4.19) for the rectangular candidate and the tests (A.15) for the circular/elliptical candidates. The candidates declared meaningful at this stage will compete subsequently for the position of the *most meaningful* interpretation for the given data and the winner will be kept as final valid detection.

4.8 *NFA* - Model Selection Criterion

The usage of the main *a contrario* concept, the number of false alarms, evolved from its initial meaning. In the first *a contrario* works, the *NFA* of a candidate served as validation criterion, to identify *meaningful* candidates [Desolneux 2000, Grompone von Gioi 2010]. Soon thereafter, it became also a selection criterion, designed to choose the *most meaningful* candidate among several meaningful candidates belonging to the same family: Moisan and Stival [Moisan 2004] use the *NFA* to select the fundamental matrix that justifies *best* a set of point correspondences extracted from a pair of images, among several possible models which yield meaningful interpretations, by using a *RANSAC* approach. In [Desolneux 2007] (Chapter 6, *Maximal meaningfulness and the exclusion principle*), the authors choose the most meaningful line segment, when multiple meaningful line segments can be fitted on the same set of pixels. Later, *NFA* carried out the model selection between different types of candidates, belonging to the same family [Cao 2007, Grompone von Gioi 2007]. The former deals with a clustering problem, where the topic of merging two neighbour clusters C_1 and C_2 into a larger one C appears. The authors suggest that comparing the meaningfulness of C_1 (or of C_2) against the meaningfulness of C is not *fair*, and they proposed a solution to measure the meaningfulness of a new type of candidate, namely the *pair of clusters* (C_1, C_2), by using a trinomial coefficient in the computation of the *NFA*. The procedure was *generalised* in [Grompone von Gioi 2007], where the authors search for the best interpretation between a large line segment and multiple (2 or more) collinear shorter line *subsegments*, by computing the *NFA* for a new candidate type, the so-called *multisegment*.

In the present work, we try to make a step forward and use the *NFA* as a model selection criterion to decide between interpretations belonging to different families: line segments or circular/elliptical arcs.

The model selection theory is a central subject in statistical learning [Hastie 2001, Duda 2001]. The proposed ellipse detector *ELSD* entails a model selection step within a linear regression problem and uses the *NFA* of a candidate as model selection criterion. Hence, a formal positioning of the *NFA* in the classic model selection framework deserves a deeper analysis. Because a thorough investigation of this issue falls beyond the scope of our work, we will give here only a sketch study, in order to assess the pertinence of using *NFA* as model selection criterion. A rigourous study is foreseen as perspective work.

Model Selection Setup

Consider the general regression model

$$Y_i = f(\mathbf{x}_i) + \nu_i, i = \{1, .., l\},$$

where $\mathbf{x}_i = (x_{i1}, \dots, x_{id})$ is the value of a d-dimensional variable at the i-th observation among l, Y_i is the response, f is the true regression function, and the random errors ν_i are assumed to be i.i.d. normally distributed with mean zero and variance σ^2. For simplicity, we assume that the input values are fixed in advance (non-random), thus we denote them by minuscules, whilst the response is denoted by majuscule, being a random variable.

Usually, the true regression function f is unknown and only the prediction model $\hat{f}$, estimated from a training sample, is available.

Definitions

We define a *loss function* to measure the errors between Y and $\hat{f}(\mathbf{x})$, that we denote by $L(Y, \hat{f}(\mathbf{x}))$. A typical choice for L is the squared error [Hastie 2001, p. 194]:

$$L(Y, \hat{f}(\mathbf{x})) = (Y - \hat{f}(\mathbf{x}))^2.$$

The *test error* (also called the *generalisation error*), is the *expected prediction error* over an independent test sample (which was not used in the training phase):

$$EPE = \mathbb{E}[L(Y, \hat{f}(\mathbf{x}))].$$

The *training error* is the average loss over the training sample:

$$\overline{\text{err}} = \frac{1}{l} \sum_{i=1}^{l} L(Y_i, \hat{f}(\mathbf{x}_i)).$$

Model Selection – Tradeoff between Bias and Variance

Within the statistical model selection problem, a finite number of plausible linear models are considered:

$$Y = f_j(\mathbf{x}, \theta_j) + \nu,$$

where for each j, $\mathcal{F}_j = \{f_j(\mathbf{x}, \theta_j), \theta_j \in \Theta_j\}$ is a linear family of regression functions with θ_j being the parameter of a finite dimension t_j. Thus, t_j accounts for the number of parameters of the model j, which reflects the *model complexity*. We suppose that t_j grows with j.

For an input point $\mathbf{x}_0$, the *EPE* for a squared error loss associated to a candidate model $\hat{f}_j$ can be decomposed as [Hastie 2001, p. 197]:

$$\begin{aligned} EPE_j(x_0) &= \mathbb{E}[(Y - \hat{f}_j(\mathbf{x}_0))^2] \\ &= \text{Irreducible Error} + \text{Bias}^2 + \text{Variance}. \end{aligned}$$

This decomposition reflects that the model selection problem is grounded on a *bias-variance tradeoff*. Indeed, the first term – Irreducible Error – represents the variance of the new test target and is beyond our control, even if we knew the true $f(\mathbf{x}_0)$. The second and the third terms are under our control, and represent the mean squared error of $\hat{f}_j(\mathbf{x}_0)$ in approximating $f(\mathbf{x}_0)$. It sums up a squared bias component and a variance component. The bias term is the amount by which the average of our estimate differs from the true mean. The variance term is the expected squared deviation of $\hat{f}_j(\mathbf{x}_0)$ around its mean. In general, as the model complexity is increased, the variance tends to increase and the squared bias tends to decrease. The opposite behaviour occurs as the model complexity is decreased.

Figure 4.13 illustrates more intuitively this behaviour. The true model f is the curve in green. If the input data are highly non linear, a low-complexity model (a linear model in figure 4.13 left) will not have the flexibility needed to capture the global shape of the distribution. The line will be most of the time far from the data points, leading to large errors. The model is then said to have a large bias because the mean distance of its predictions for a given x_0 (blue dot) with respect to the true model is high. On the other hand, because of this *rigidity*, the predictions of the model will depend only little on the particular sample that was used for building the model, and will therefore have a low variance. We talk about *underfitting* in this case. But a too complex model (figure 4.13 right) will make the curve very sensitive to the details of the sample. Another sample would have lead to a completely different model, with completely different predictions. The model is then said to have a large variance because the variance of its predictions (for a given x_0) is large. The increased complexity confers *flexibility*, so the estimated curve can now get close to the points (low bias). This is an example of *overfitting*.

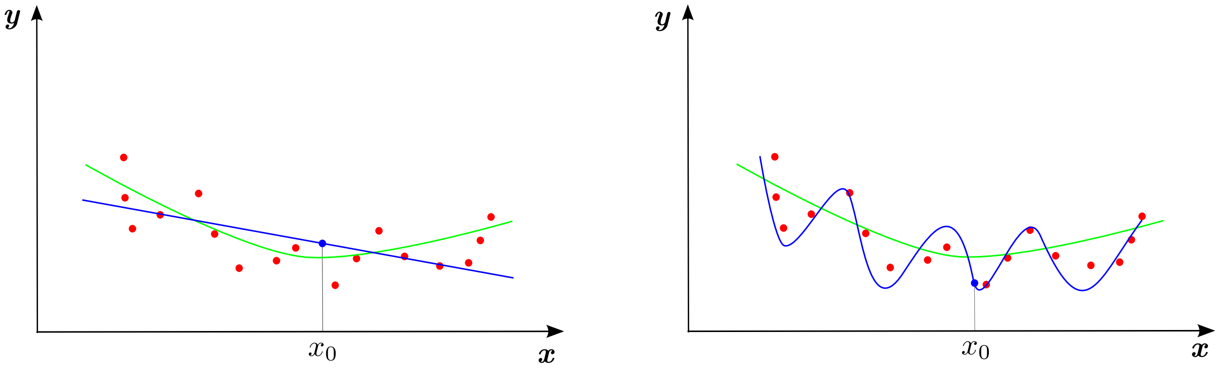


Figure 4.13: *The bias-variance tradeoff. In green: the true model. Left: example of data underfitting. Right: example of data overfitting.*

Figure 4.14 shows the behaviour of the test and training error, as the model complexity varies. The training error tends to decrease whenever the model complexity is increased, i.e. the data are fitted harder. However, with overfitting, the model adapts itself too closely to the training data, and will not generalise well, yielding an increased test error, because of the large variance. In contrast, if the model is not complex enough, it will underfit the input data, having an important bias. Again, the generalisation will be poor.

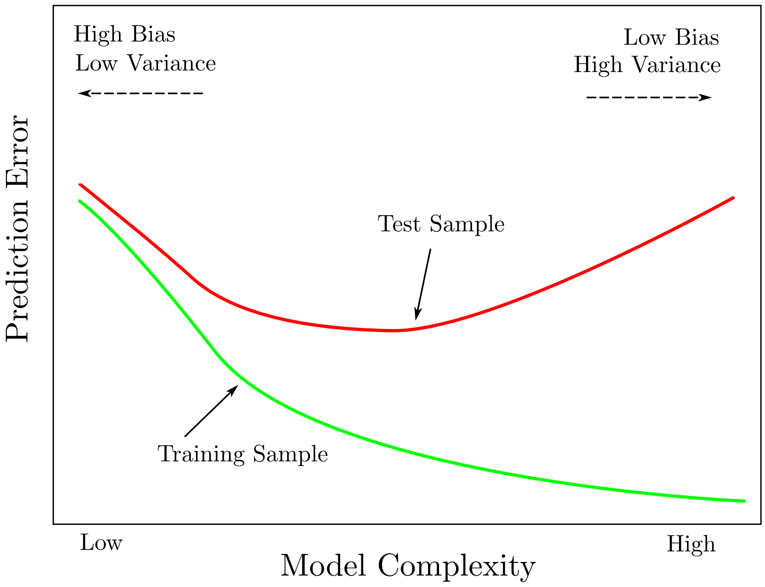


Figure 4.14: *Behaviour of test sample and training sample error as the model complexity is varied.*

The goal of the model selection would be to trade bias off with variance in such a way as to choose the model minimising the test error, and hence to estimate the optimal amount of model complexity. Given the candidate models, we need to select the one that captures best the underlying distribution of the data, while penalising the complexity. This statement is a formalisation of the Ockham's razor principle (*law of parsimony*): *"plurality should not be posited without necessity"* [18]. Other researchers tackled to a certain extent the *simplicity* principle and concluded that *"people seem to perceive the simplest possible interpretation for any given data"* [Lowe 1985, p. 21].

Classification of Model Selection Criteria

Up to now, there is a rather large literature on model selection methods. A simple classification of the existing criteria is given in [Rust 1995], where the authors distinguish *split sample*, *jackknife*, and *full sample* criteria. *Split sample* criteria require that the empirical data be divided into two parts: an *estimation sample* and a *validation sample*. Model parameters are estimated on the estimation sample, and then model performance is tested on the validation sample. *Jackknife* criteria do a similar cross-validation, one observation at a time. Each data point is held out in turn, the model parameters are estimated on the rest of the points, and the likelihood of the holdout point is evaluated. A *pseudo-likelihood*, the product of the individual point likelihoods, is then computed and used to compare model performance. *Full sample* criteria calculate the maximum likelihood and then adjust for parsimony, by subtracting a penalty term, which is an increasing function of the number of estimated parameters. These criteria are the easiest computationally, and hence gained widespread popularity.

In order to assess the pertinence of using *NFA* as model selection criterion, we will compare its selection capabilities against a classic criterion. One of the most representative classic model selection criterion is Akaike's information criterion (*AIC*) [Akaike 1973], which belongs to the full sample class. Our study will focus on the comparison between *NFA* and *AIC*.

Akaike's Information Criterion (*AIC*)

AIC was proposed within the field of Information Theory and it offers a relative measure of the information loss when a given model is used to describe reality. Thus, it reflects the Kullback-Leibler divergence[19] between the true distribution and its estimate [Yang 2003a]. *AIC* is astonishingly simple. This criterion uses a fine term to penalise the log maximum likelihood for lack of parsimony. For an input sample containing l points and an estimated model $\hat{f}_j$, having t_j parameters,

$$AIC_j = -\frac{2}{l} \cdot \log \mathsf{L}_j + 2 \cdot \frac{t_j}{l}, \tag{4.22}$$

where L is the maximized value of the likelihood function for the estimated model j [Hastie 2001].

[18] *Ockham's razor.* Encyclopædia Britannica. Encyclopædia Britannica Online. 2010.

[19] Kullback-Leibler divergence (*KL*) [Kullback 1951] is a non-symmetric measure of the difference between two probability distributions P and Q. *KL* measures the expected number of extra bits required to code samples from P when using a code based on Q, rather than using a code based on P. Typically P represents the "true" distribution of data and the measure Q represents an approximation (estimate) of P.

Given a data set and several candidate models, they are ranked according to their *AIC* values. *AIC* is, par excellence, a tool for model selection. It does not provide a test of a model in the usual sense of testing a null hypothesis, i.e. it does not measure how well a model fits the data in an absolute sense. Hence, if all the candidate models fit poorly, *AIC* will give no warning of that.

NFA Continuous Formulation

Thus far, we have introduced and used only the *discrete formulation* for the *NFA* of a feature candidate (4.15, A.15). With this formulation, the pixels of a candidate are considered as a sequence of Bernoulli variables, taking the value 1 when a pixel is *aligned*, and 0 otherwise (4.13). We suggest that, especially for model selection purposes, the *continuous formulation* of the *NFA* could be of interest. In this section we will denote the continuous *NFA* by NFA_c and the discrete formulation by NFA_d.

A continuous *NFA* was proposed by Igual et al. in [Igual 2007]. They use a continuous distribution in the *NFA* computation, which serves afterwards as merging criterion for two adjacent regions in a satellite segmentation application [Igual 2007]. Following their idea, Grompone von Gioi et al. proposed a continuous *NFA* formulation for the line segment detection [Grompone von Gioi 2009], that we will experiment in the model selection problem.

When using a continuous distribution, a feature candidate s of length l is not anymore a binary sequence of random variables $(X_1, X_2, \ldots, X_l)$, encoding the fact that the pixels are δ-aligned or not. Instead, the values of the gradient orientations are normalised between 0 and 1 and kept as is. Thus, let $s = (x_1, x_2, \ldots, x_l)$, where $x_i \in (0, 1]$. A function $t(s)$ needs to be defined to measure the degree of structuredness of the candidate s, similarly with the discrete case. The meaningfulness of s is then given by

$$NFA_c(s) = N_t \mathbb{P}[T(s) \geq t_s], t_s = t(s). \tag{4.23}$$

In [Grompone von Gioi 2009], Grompone von Gioi et al. proposed the function $t(s) = -\sum_{i=1}^{l} \log(x_i)$. In this case, $T(s)$ follows a Γ distribution of parameters $(l, 1)$, for which the cumulative function is given by:

$$\mathbb{P}[T(s) \geq t_s] = \exp(-t_s) e_{l-1}(t_s),$$

where $e_{l-1}(t_s) = 1 + t_s + \frac{t_s^2}{2} + \ldots + \frac{t_s^{l-1}}{(l-1)!}$. Eventually, it can be shown that if a candidate s is accepted as ε-meaningful when

$$NFA_c(s) = N_t \left(\prod_{i=1}^{l} x_i \right) e_{l-1} \left(- \sum_{i=1}^{l} log(x_i) \right) \leq \varepsilon, \tag{4.24}$$

then the number of expected ε-meaningful candidates in a noise image is up-bounded by ε [Grompone von Gioi 2009]. The continuous formulation has the major advantage of no longer needing the precision parameter δ, used in the discrete formulation.

We add the continuous formulation in our study on the suitability of the *NFA* as model selection criterion.

NFA vs *AIC*

Within the *ELSD* model selection problem, we have three competing models: line segment, circular arc, and elliptical arc, and we choose to keep the one possessing the smallest *NFA*. We are interested in comparing qualitatively and quantitatively *NFA* (both discrete and continuous formulations) and *AIC* in the context of model selection, to assess the pertinence of using *NFA* as model selection criterion.

Let us consider an image of $n \times n$ pixels. Given an alignment candidate containing l pixels, among which k are δ-aligned, the scores assigned by the three criteria – NFA_d, NFA_c, and *AIC* – to a candidate model j, are:

$$\begin{array}{rcl} \log NFA_{dj} & = & \log \mathcal{B}(l,k,\delta) + t_j \cdot \log n \\ \log NFA_{cj} & = & \log \mathbb{P}[T(s) \geq t_s] + t_j \cdot \log n \\ AIC_j & = & -\frac{2}{l} \cdot \log \mathsf{L}_j + 2 \cdot \frac{t_j}{l}. \end{array} \tag{4.25}$$

In a trivial sense, we could say that the three criteria manifest a somehow similar behaviour. The first right-hand terms in all expressions indicate the goodness (structuredness) of the alignment with respect to the considered model j, whereas the second term penalises for complexity. The differences however, e.g. the two *NFA* expressions incorporate the image size in the term penalising for complexity, come from the different purposes that the criteria serve: NFA_d and NFA_c serve as validation criteria as well, whereas *AIC* is only a model selection tool, focusing exclusively on the given candidate feature and ignoring the context in which the candidate occurred.

From a quantitative point of view, the analysis requires a different setup for *ELSD*, in order to yield the three criteria comparable. From the expression (4.22), it is obvious that *AIC* is suitable as selection criterion in problems where the competing candidate models are fitted using maximum likelihood (*ML*) estimators. But, for efficiency reasons, the candidate models compared in *ELSD* are obtained through direct algebraic estimators (estimators maximising the likelihood are iterative, thus we avoided them). For testing purposes, we choose to handle this conflict by replacing in *ELSD* the direct algebraic estimators through iterative estimators based on likelihood maximisation. In this way, the three criteria will have equal grounds in promoting their model selection capabilities. For ellipse fitting, we use Kanatani's "compact algorithm for strictly *ML* ellipse fitting"[20]. The circle fitting is performed using Chernov's estimator[21].

The tests were carried out on computer-generated images, containing linear shapes or circular/elliptical shapes, affected by a small amount of Gaussian noise. For each image, a support region containing pixels with a strong gradient (which are possibly located on an edge) is selected, and the three competing models – line segment, circle, ellipse – are fitted using the aforementioned *ML* estimators. For the images containing circular/elliptical shapes, the support regions were chosen as to cover different portions of the circle/ellipse contour, e.g. only a quarter of the ellipse contour or a half, or the entire ellipse. Afterwards, for each fitted model, we compute its *AIC*, its NFA_d and its NFA_c. Possessing the ground truth, i.e. the real shape

[20] Code available online at `http://www.iim.ics.tut.ac.jp/~sugaya/public-e.html`.

[21] Code available online at `http://www.math.uab.edu/~chernov/cl/LMA.m`.

contained in the selected support region, we are interested to evaluate which criterion designates the correct model. The results are shown in tables 4.1, 4.2, and 4.3. The explanations following each table are important for their interpretation.

Ellipse	*AIC*	NFA_d	NFA_c
1/4 of contour	23%	13%	18%
1/2 of contour	100 %	84 %	99%
entire contour	100%	100%	100%

Table 4.1: *The proportion of elliptical models correctly designated by the three model selection criteria: AIC, NFA_d, and NFA_c on* 200 *tests.*

Images of elliptical shapes. The first line of the table 4.1 indicates a poor behaviour for all the three criteria. In the failure cases, all the three criteria designate the circular candidate as winner. The reason is that the *ML* ellipse estimator fails in proposing a coherent candidate. Most of the time, the proposed candidate is a general conic (hyperbole, parabola), but not an ellipse. Thus, the circular candidate is chosen instead. Nonetheless, *AIC* indicates a slightly better performance. The explanation comes from the different pixel features that the three criteria take into account to measure the goodness of a candidate model: *AIC* considers the distance between the input points and the fitted ellipse, whereas the two *NFA* criteria consider the gradient orientations. In general, the cases where *AIC* gives the correct result and the two *NFA* fail are those where the fitted conic is an ellipse, but the precision is very poor (the fitted ellipse is much bigger than the real one). The points are close to the fitted ellipse, thus *AIC* gives a good score, but the pixel directions are not aligned with the theoretical gradient directions that the points on the fitted ellipse should have, yielding an *NFA* selection failure. In these cases, *AIC* chooses the correct feature family (even if the fitted model is far from the true model), whilst the two *NFA* criteria are prevented from choosing the elliptical model, by the validation characteristic of the *NFA*: the proposed elliptical model is too far from the true model to be considered as meaningful or as more meaningful than the circular model candidate. Line 2 shows that the discrete *NFA* has the tendency to vote for simplicity, choosing the circular model. The continuous *NFA* is very close to *AIC*.

Circle	*AIC*	NFA_d	NFA_c
1/16 of contour	0%	0%	0%
1/8 of contour	83%	77%	88%
1/4 of contour	100%	85%	95%
1/2 of contour	100 %	100 %	100%
entire contour	100%	100%	100%

Table 4.2: *The proportion of circular models correctly designated by the three model selection criteria: AIC, NFA_d, and NFA_c on* 200 *tests.*

Images of circular shapes. When only about 1/16 from the circle contour is covered by the support region, all the three criteria choose the simplest model, i.e. the linear model. For the case where 1/8 of the contour is available, the NFA_c shows the best performance. Dealing with relatively small arcs, the positions of the pixels may indicate rather a linear structure, causing *AIC* and NFA_d to fail. The continuous *NFA*, which captures faithfully the gradient directions indicating a circular structure, behaves slightly better. When a quart of the contour is given, the two *NFA* criteria yield some rather surprising failures, voting for the elliptical model, thus for overfitting. We hold responsible for these cases the approximations made in computing the areas of the circular/elliptical rings, involved in the *NFA* computation.

Line	*AIC*	NFA_d	NFA_c
entire contour	100%	100%	100%

Table 4.3: *The proportion of linear models correctly designated by the three model selection criteria: AIC,* NFA_d*, and* NFA_c *on* 200 *tests.*

Images of linear shapes. All the three criteria choose the correct model, i.e. the linear one. Most of the time, the candidate model proposed by the ellipse *ML* estimator is not a valid ellipse, thus it is successfully rejected. It worths mentioning though that the frontier between the linear model and the circular model is quite fragile in some cases for all the three criteria, as on the support region containing a line segment, a very large circle will also fit reasonably well the input data. In these cases, the term penalising for complexity has its word.

To conclude this section, we suggest that the performance of the *number of false alarms* as model selection criterion, and in particular the performance of the continuous *NFA*, is quite *similar*[22] with the performance of a classic representative criterion, the *AIC*. Moreover, the use of *NFA* is convenient because it carries out also the task of candidate validation, a quality which *AIC* does not possess. Nonetheless, a deeper and more accurate study is foreseen as perspective work.

4.9 Distance between Points

Within the *a contrario* approach, the meaningfulness of a candidate s, belonging to an image x of size $n \times m$, is assessed by computing its *NFA*, which is the expected number of candidates at least as structured as the observed one s, under the null hypothesis $\mathcal{H}_0$ (4.2). When using the discrete formulation for the *NFA*, this comes to evaluating the probability of having at least k successes among l, i.e. at least k δ-aligned pixels in a length l line segment. If the gradient orientations

[22] One may argue that this comparison is not completely sound, as the three criteria use different pixel features in attributing scores to the candidate models: *AIC* considers the positional error with respect to the fitted model, whereas the two *NFA* consider the derivative error. A correct comparison would require either circle/ellipse estimators that maximise the likelihood considering the distribution of the gradient orientations, or an *NFA* computed considering a positional noise model. Because of time limitation, this study will be left as perspective work. Here we are interested in a grossly comparison between *NFA* and a classic criterion, that could reflect the pertinence of using *NFA* as model selection criterion.

are considered as Bernoulli independent variables, the p-value is given by the binomial tail of parameters (l, k, δ) (4.15). For the independence hypothesis to stand, Desolneux et al. compute the gradient using a discrete operator having the smallest support possible, i.e. 2×2 mask as given in (4.26), and then 1 pixel out of 2 is dropped [Desolneux 2000]:

$$\nabla x(i,j) = \begin{pmatrix} \frac{x(i+1,j)-x(i,j)+x(i+1,j+1)-x(i,j+1)}{2} \\ \frac{x(i,j+1)-x(i,j)+x(i+1,j+1)-x(i+1,j)}{2} \end{pmatrix}. \tag{4.26}$$

As experimented in [Grompone von Gioi 2008], dropping pixels results in rejecting small, but valid candidates. To cope with this issue, Grompone von Gioi et al. tested the influence of the correlation on the overall detection result by considering all the points under a false hypothesis of independence. By running exhaustive detection tests on noise images, they showed that the number of false positives does not *explode*, i.e. the number of ε-meaningful line segments is still up-bounded by ε in noise images. They concluded thus that the correlation is too weak (a correlation coefficient of about 0.22 can be observed on noise images) to deteriorate significantly the results, and proposed to use all the points.

To join their idea, we seek to formalise the computation of the discrete *NFA* with dependent points. We show that the influence of the correlation on the binomial tail has a rather unexpected effect in this context: considering all the points under the false hypothesis of independence does not make the number of false positives to grow. By the contrary, it can introduce false negatives.

The problem can be formulated as follows.

Problem 2 *If s is a length l line segment*[23]*, containing l pixels $p_i, i \in \{1, .., l\}$, which can be unambiguously classified as a success (p_i is aligned with the line segment s at a precision δ) or a failure (p_i is not aligned), compute the probability of observing k aligned pixels among l, i.e. the probability of having k successes in l trials, assuming the existence of correlation between trials, which prohibits the use of conventional techniques, like the binomial distribution.*

This problem was tackled in the statistical literature, at first to test for the existence of correlation and to study its impact on the conventional techniques. Several authors [Walsh 1951, Anderson 1954] concluded that the tests may be highly misleading if the observations are even slightly correlated. Afterwards, different procedures to obtain the distributions of certain correlated random variables were proposed: Gringorten uses Monte Carlo techniques to compute approximate distributions for random variables of interest to meteorologists [Gringorten 1966], whereas Gabriel [Gabriel 1959] and Helgert [Helgert 1970] use Markov chain-based models to compute exact probability distributions involved in hypothesis testing problems. Using the same theoretical ground, i.e. Markov chain modelling, Ladd published an efficient algorithm for computing the probability of having k successes in l correlated (dependent) trials [Ladd 1975]. His goal was to estimate the impact of the one stage dependence (the outcome of one trial affects the outcome of the next, and no others) in problems like developing confidence intervals and hypothesis testing. His results show that for particular types of correlated samples, correlations with coefficients less than 0.4 do not deteriorate significantly the results. In the following, we detail Ladd's algorithm, based on a first-order Markov chain

[23] We are talking about discrete line segments here, i.e. a fine set of connected pixels.

model, that we implemented in order to study the effect of the dependence between the gradient orientations in the *a contrario* line segment detection problem.

Markov Chain Modelling of Gradient Orientations

Let $X_i, i \in \{1,..,l\}$ be a sequence of random variables, associated to the length l line segment s. X_i can take binary values: 1 (success, p_i is aligned) or 0 (failure, p_i is not aligned). Hence, they form a sequence of Bernoulli trials (4.13). Let $P = \mathbb{P}[X_i = 1]$, for every $i \in \{1,..,l\}$.

In order to take into account the correlation between the gradient orientations of the neighbour pixels, we approximate the sequence of random variables by a first-order Markov chain. The dependence level is suggested by the 2×2 size of the gradient operator support.

The proposed first-order Markov chain representation involves the following conditional probabilities:

$$\begin{array}{rcccccccc} & & P_{11} & = & \mathbb{P}(X_i & = & 1|X_{i-1} & = & 1), \\ 1-P_{11} & = & P_{01} & = & \mathbb{P}(X_i & = & 0|X_{i-1} & = & 1), \\ & & P_{10} & = & \mathbb{P}(X_i & = & 1|X_{i-1} & = & 0), \\ 1-P_{10} & = & P_{00} & = & \mathbb{P}(X_i & = & 0|X_{i-1} & = & 0). \end{array} \tag{4.27}$$

Thus, P_{11} is the probability of the pixel i to be aligned, knowing that the preceding pixel p_{i-1} was aligned and so forth.

For a particular $i = n$, let $P_n = \mathbb{P}(X_n = 1)$. For the Markov model, one has $P_n = P_{n-1}P_{11} + (1 - P_{n-1})P_{10}$. Since $P_{n-1} = P = P_n$, one also has $P = PP_{11} + (1 - P)P_{10}$; hence

$$P_{11} = 1 - \frac{1-P}{P}P_{10}. \tag{4.28}$$

Thus, all four of the conditional probabilities defined in (4.27), can be directly determined as soon as P and either P_{11} or P_{10} are given.

Ladd's Algorithm

Ladd's algorithm [Ladd 1975] can be a useful tool in problems that require the distribution of the probability of observing k successes in l dependent trials, when the trials can be modelled through a first-order Markov chain. Let $\phi_{k,l}(P, P_{11}) \equiv \mathbb{P}[k \text{ successes in } l \text{ trials } |P, P_{11}]$. For ease in writing, we denote this as $\phi_{k,l}$.

An intuitive representation of the various outcomes of a series of trials is given in figure 4.15. Each vertical line represents one trial. The nodes on the lines represent the possible number of successes among a certain number of trials, given by the depth of the node in the tree. A going-up is considered a *miss*, whereas a going-down is a *hit*. Thus, on each line, the nodes correspond to values of k, running from 0 (uppermost) to l (lowermost). Finally, $\phi_{k,l}$ represents the probability of reaching the node (k, l). Because of the two-stage dependence, the probability of reaching the point (k, l) depends on whether the two preceding points $(k-1, l-1)$ and $(k, l-1)$ were reached by a *success* or by a *failure*. Let

$$\phi_{k,l} = \eta_{k,l} + \xi_{k,l},$$

where

$$\eta_{k,l} \equiv \mathbb{P}[k \text{ successes in } l \text{ trials}, l^{th} \text{ trial fails}],$$

and

$$\xi_{k,l} \equiv \mathbb{P}[k \text{ successes in } l \text{ trials}, l^{th} \text{ trial succeeds}].$$

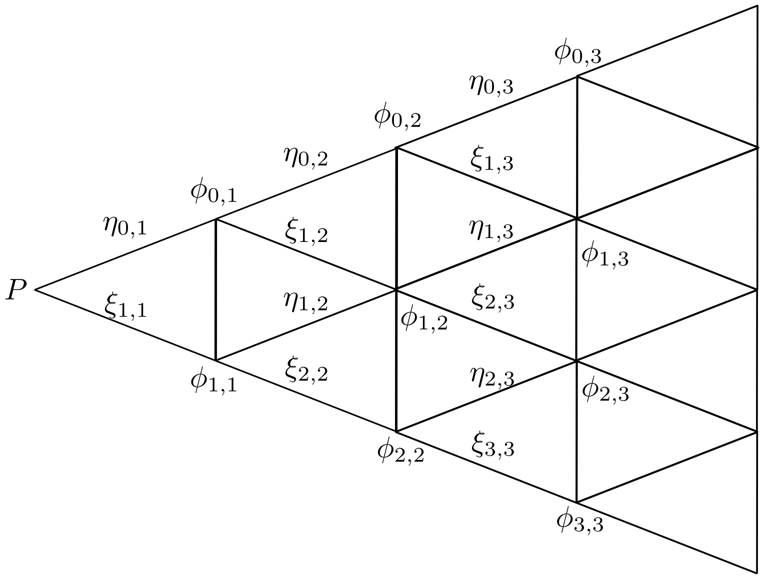


Figure 4.15: *A sequence of trial outcomes*

To implement the dependence tree from figure 4.15, the next recursive algorithm is used:

1. for the first trial:

$$\begin{array}{lll} \eta_{0,1} = 1 - P, & \eta_{1,1} = 0; & \phi_{0,1} = 1 - P \\ \xi_{0,1} = 0, & \xi_{1,1} = P; & \phi_{1,1} = P \end{array}$$

2. for succeeding trials:

$$\begin{array}{lll} \eta_{0,l} = \eta_{0,l-1} P_{00} = (1-\delta) P_{00}^{l-1} & \xi_{0,l} = 0 & \\ \eta_{k,l} = \xi_{k,l-1} P_{01} + \eta_{k,l-1} P_{00} & \xi_{k,l} = \eta_{k-1,l-1} P_{10} + \xi_{k-1,l-1} P_{11} & k = \{1, .., l-1\} \\ \eta_{l,l} = 0 & \xi_{l,l} = \xi_{l-1,l-1} P_{11} = P P_{11}^{l-1} & \end{array}$$

Once the ϕ values are computed, it is easy to obtain the probability of having k or fewer successes in l trials, as:

$$S(l, k, P) = \sum_{i=0}^{k} \phi_{i,l},$$

which will be used subsequently to compute the probability of having at least k successes in l

trials, which we denote by $\mathcal{M}(l, k, P)$:

$$\mathcal{M}(l, k, P) = 1 - \sum_{i=0}^{k-1} \phi_{i,l}.$$

Finally, the expression of the *NFA* associated to a candidate s_i, of length l, containing k dependent aligned pixels is:

$$NFA(s_i) = N_t \mathcal{M}(l, k, P).$$

As said before, the computation of the conditional probabilities from (4.27), involved in the algorithm, require the values of P and P_{11} (or P_{01}). For P we consider the probability of success used in the independent model, $P = \delta$. The probability of having an aligned pixel, knowing that the preceding pixel was aligned P_{11}, was learned on noise images. The obtained value is $P_{11} = 0.0267$. As a sanity check, it should be observed that for $P_{11} = P$, the output of the algorithm is similar with the binomial distribution.

The purpose of our study is to enforce (formalise) the proposition of Grompone von Gioi et al. to use all the points under the false hypothesis of independence. To this end, we have compared the probability distribution obtained using the algorithm presented above, with the binomial distribution used in the definition of the *NFA* (4.15). Figure 4.16 left shows the behaviour of the probability distribution with respect to the learnt probability P_{11}. It can be observed that the probability distribution has the same aspect as the binomial one (in green); only the amplitude of the principal mode depends on the value of P_{11}: for values $P_{11} < \delta$, the curve is above the binomial (in blue), whereas for $P_{11} > \delta$, the curve passes beneath (in red). The consequence of this behaviour is reflected in figure 4.16 right, which illustrates the minimum number of dependent aligned points that a line segment must contain in order to become meaningful, using the two methods: binomial with false independence hypothesis (that we call *binomial NFA*) and dependent points modelled by a Markov chain (*Markov NFA*). It can be observed that with the presented algorithm, fewer points are needed for a candidate to be meaningful. Thus, Grompone von Gioi et al.'s proposition will not introduce false positives; by the contrary, their method is too restrictive, as it requires more aligned points to accept a candidate as valid.

The difference is not, however, crucial. We have tested the *LSD* algorithm using the two procedures for the *NFA* computation. An example can be seen in figure 4.17, where one can observe that in the left image, where the *Markov NFA* is used, there are small line segments which were discarded by the *binomial NFA* (right image). The choice between the two procedures must be made considering the application needs, precision vs execution time: if one wishes a very precise detection, the *Markov NFA* should be used, whereas in applications where execution time is critical, the *binomial NFA* should be kept instead.

4.10 Experiments

ELSD targets the simultaneous detection of line segments, circular and elliptical arcs in digital images. Here, we will present some comparative results, in order to stress the advantages of *ELSD* in front of other detectors. There are very few detectors that address simultaneously

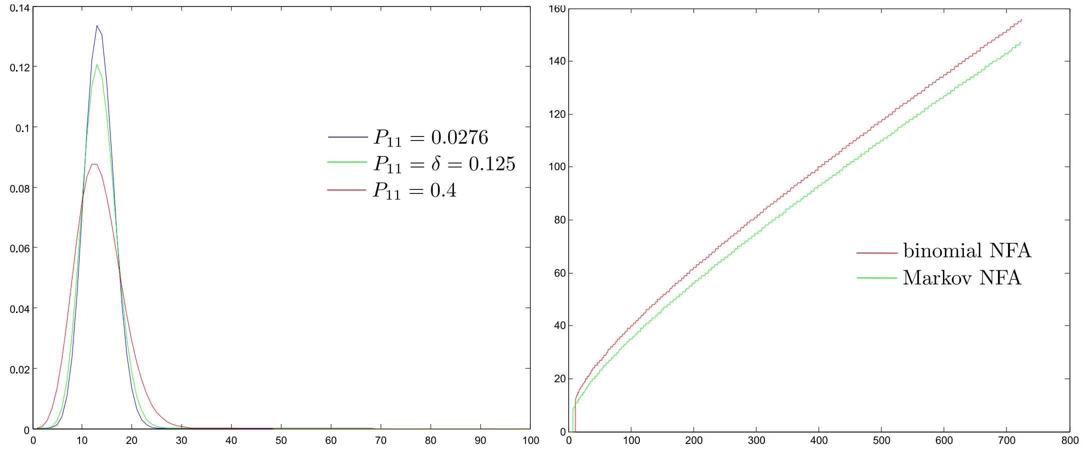


Figure 4.16: *Left image: The probability distribution computed with Ladd's algorithm, for different values of* P_{11}. *For* $P_{11} = \delta$, *the distribution computed with Ladd's algorithm is identical with the binomial distribution (green). Right image: The minimum number* k_{min} *of aligned points, observed in a* 1*-meaningful line segment in an image of* 512×512, *for a precision* $\delta = 1/8$.

the detection of several primitive types. We will test Etemadi's algorithm [Etemadi 1992], that targets line segments and circular arcs and some Hough-based algorithms that address only a primitive family at a time (lines, circles, ellipses).

More precisely, the algorithms involved in our study are:

- Etemadi's algorithm for curve segmentation into line segments and circular arcs (that we will call *Etemadi*);
- the Hough line detector (*Hline*) proposed by Barinova et al., which uses a parameterless non-maxima suppression [Barinova 2010];
- the Hough circle detector available in the opencv library (*RHTcircle*); it implements an improved variant of *RHT*, that uses also the gradient direction;
- the ellipse detector (*RHTellipse*) implemented by Andrew Schuler, based on *RHT* (Matlab implementation)[24]. The same approach is available online in the *MIPAV* library (Java implementation)[25].

The detectors tested here, except the proposed one, operate on edge maps, and their results are highly influenced by the quality of the edge map. They exhibit an important number of false positives in presence of noise in the edge map, as it will be shown in the sequel. Equally, Hough-based methods do not handle well images containing several primitive families (line segments and arcs), producing false positives, even when the edge map is relatively noise free. Or, in other cases, no detection is reported (high number of false negatives), as the different primitives interfere in a destructive way.

[24] Andrew Schuler's Matlab code `http://lars.mec.ua.pt/public/LARProjects/Laser3D/2003_MiguelMatos/TransformadasdeHough/Agorithm/TheRandomizedHoughTransformusedforellipsedetection.htm`.

[25] *MIPAV* (Medical Image Processing, Analysis and Visualisation) library `http://mipav.cit.nih.gov/`.

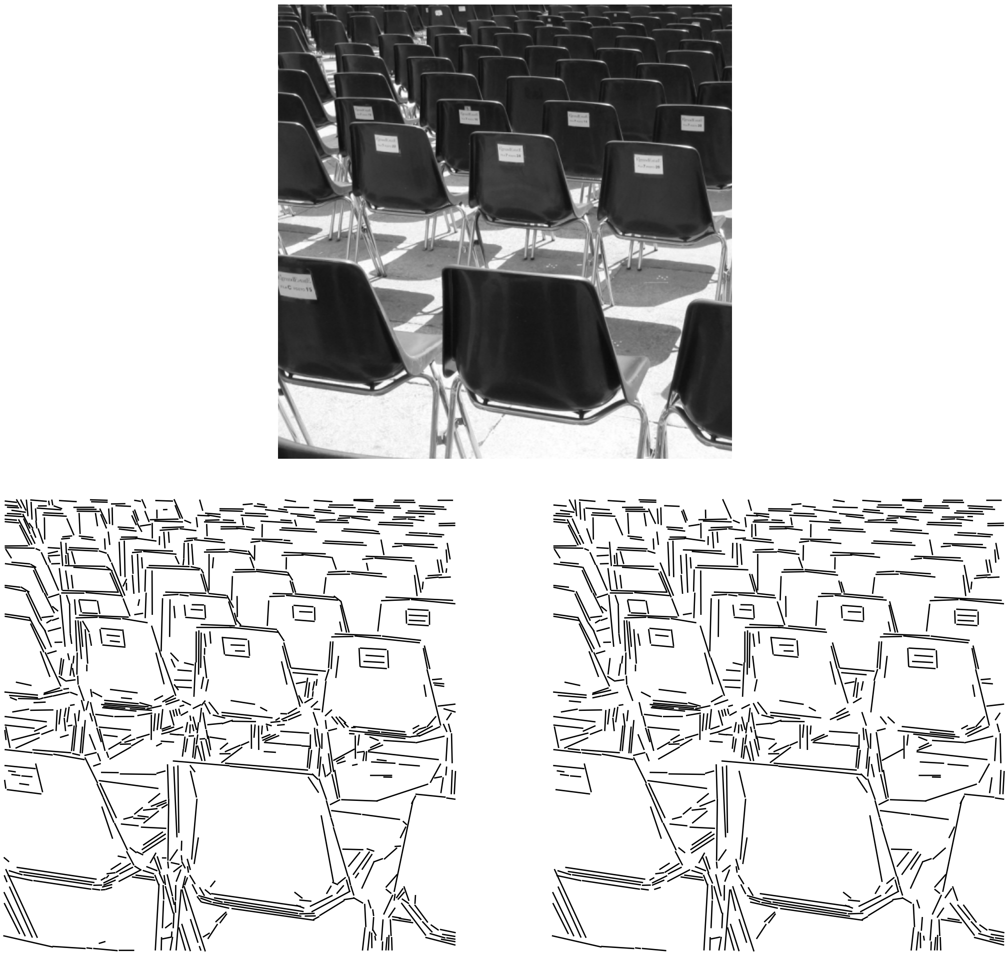


Figure 4.17: *First row: original image. Second row: comparison between the detection results obtained with Markov NFA (left) and binomial NFA (right).*

We stress the fact that *ELSD* is directly applied on gray level images; no prior edge detection is performed. Equally, the gradient is computed using a 2×2 window, and no classic denoising is applied, in order to avoid violating the independence hypothesis made on the gradient orientations. However, we use all the points, thus the independence hypothesis is not completely valid (see previous section). There are cases where the primitive detection is more satisfactory if a smoothing is performed beforehand. A Gaussian filtering can be applied, followed by subsampling. The most important aspect is that *ELSD* requires no parameter tuning, and the results are coherent, even when the image type/source/size/content changes. All detectors were tested with their default parameters. The edge maps were obtained using Canny's edge detector available in Matlab, with default parameters.

Figure 4.18 evaluates the robustness to noise of these algorithms. Even for small Gaussian noise levels, *Hlines* reports false positives. *RHTcircle* is more robust, as it takes into account the gradient orientation in the voting procedure. Both detectors, *Hlines* and *RHTcircle*, report

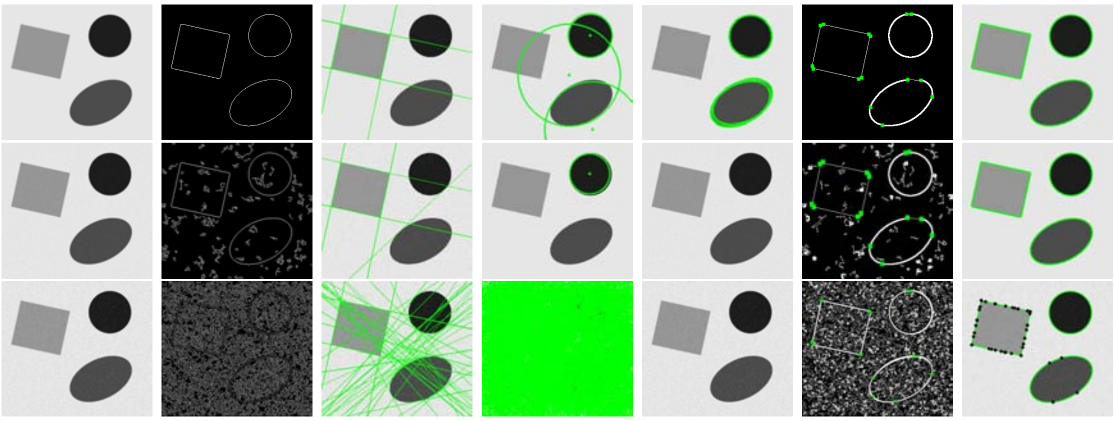


Figure 4.18: *Detection results on a simple image of geometric shapes (square, circle, ellipse) for different noise levels. From left to right: original image, edge map, Hline, RHTcircle, RHTellipse, Etemadi (the ends of the correct detections are marked in green, the rest are false positives), ELSD.*

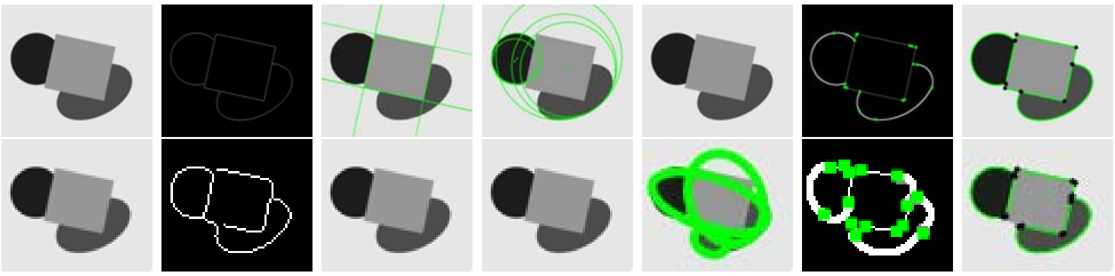


Figure 4.19: *Detection results on an image of overlapping geometric shapes at different scales (521×457 pixels and 66×58 pixels). From left to right: original image, edge map, Hline, RHTcircle, RHTellipse, Etemadi, ELSD.*

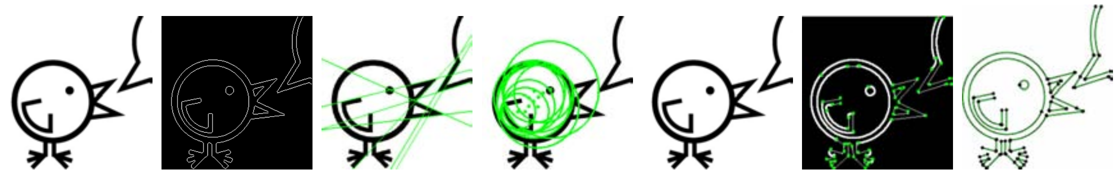


Figure 4.20: *More complex combination of geometrical features. From left to right: original image, edge map, Hline, RHTcircle, RHTellipse, Etemadi, ELSD.*

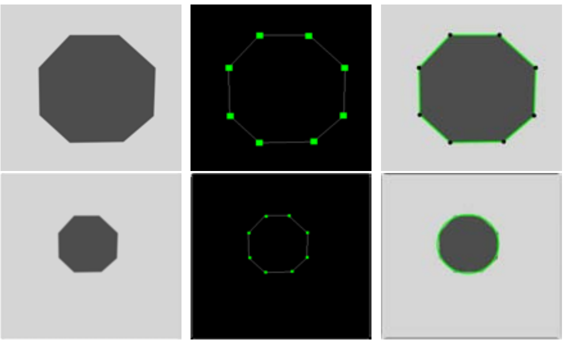


Figure 4.21: *Example of a difficult shape. From left to right: original image, Etemadi, ELSD for two different scales.*

a high number of false positives when higher noise levels are considered. *RHTellipse* is very sensitive to noisy edge maps and reports no detection, even for small noise levels. Etemadi's detector finds the correct features (it approximates the ellipses through a set of circular arcs), but it also reports a high number of false positives, being unable to handle noisy edge maps. The performance of *ELSD* is not affected by small noise levels. At higher noise levels, the features found by *ELSD* are oversegmented, but the detection is still accurate, and no false positive is found.

Figure 4.19 shows a slightly more complex image containing overlapping geometric features, at two different scales. All the three Hough-based methods report false negatives or false positives, at both scales. *Etemadi* made no false detection, but the precision of the result and of the geometric interpretation is affected by scale. Owing to the improved fitting technique, *ELSD* is very accurate even when the features are very small. Figure 4.20 presents a more complex combination of geometric features. *ELSD* is very accurate in detecting the linear and circular features. Nonetheless, the current version of *ELSD* has some difficulties in interpreting polygonal shapes. Figure 4.21 presents such an example. In case of polygonal shapes that can be fairly approximated by a circle/ellipse, *ELSD* will choose the latter. This indicates that a new candidate type must be considered in the model selection phase, namely a polygonal feature. Etemadi's algorithm returns the correct interpretation. Some other relevant results which point out to particular drawbacks of the compared detectors are given in the sequel (figures 4.22 – 4.29).

Finally, figures 4.30 – 4.33 show the result of *ELSD* on natural images. *ELSD* succeeds in providing a fair economic description of the image content, with no edge detector applied beforehand, nor tuned parameters. The behaviour of *ELSD* on natural images indicates leads for image vectorisation.

4.11 Conclusion

In this chapter we have approached some critical aspects of the primitive detection problem in digital images. Namely, we are interested in the control of the number of false detections. To this end, we proposed a parameterless primitive detector, which controls efficiently the number

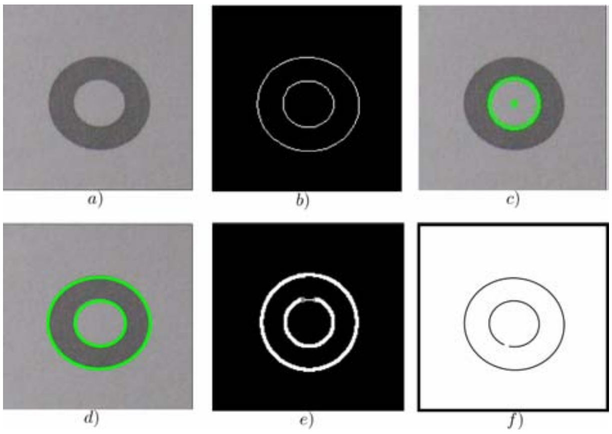


Figure 4.22: *a), b) Original image of a circular ring and its edge map. c) RHTcircle detects only one circle, it cannot handle concentric circles. d) RHTellipse detects correctly the two circles. e)Etemadi: one circle, one circular arc and one short line segment. f) ELSD: one complete circle and one circular arc.*

of false detections. The primitives of interest here are line segments, circular arcs and elliptical arcs. In our work, we distinguish two essential steps for a primitive detector: primitive candidate selection and primitive candidate validation. Extending the line segment detector proposed by Grompone von Gioi et al. [Grompone von Gioi 2010], the proposed primitive detector *ELSD* merges a greedy candidate selection and an *a contrario* validation technique. These ingredients yield a primitive detector free from parameter tuning, which formally guarantees the control of the number of false positives. Targeting several primitive families, the proposed detector requires eventually a model selection phase, carried out using the *NFA* as model selection criterion. Using *NFA* for both validation and model selection is a convenient solution, but the precision of the detector should be improved by introducing a new candidate type, namely the polygonal primitive, in order to have a cleaner model selection. Equally, the continuous formulation of *NFA* showed interesting capabilities, especially for the model selection phase, deserving further attention. The theoretical issues discussed in this chapter, i.e. the usage of the *NFA* as model selection criterion, and the independence hypothesis for the *a contrario* model, require also a deeper analysis.

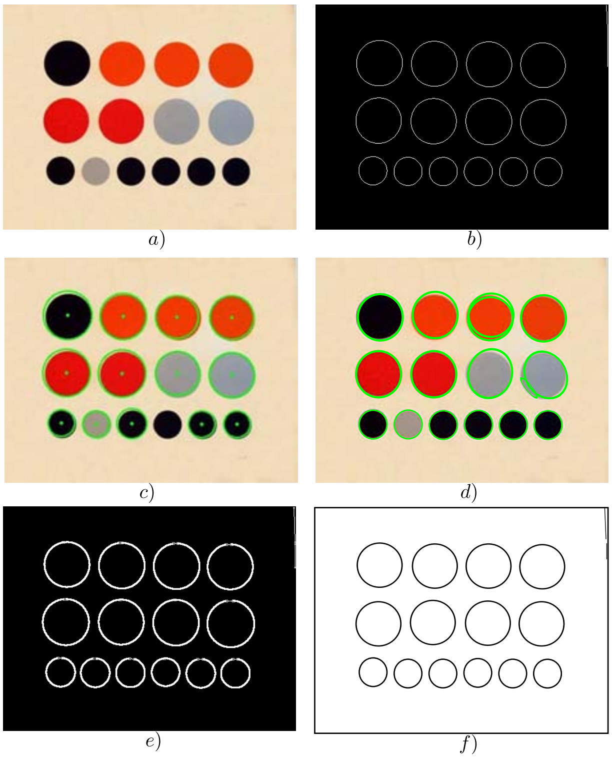


Figure 4.23: *a*), *b*) *Original image of a set of* 14 *circles and its edge map. c*) *RHTcircle misses one circle (last row). d*) *RHTellipse detects correctly* 11 *circles, the others are imprecise. e*)*Etemadi:* 10 *complete circles,* 4 *circular arcs, and* 12 *line segments. f*) *ELSD:* 6 *ellipses,* 8 *circles, and* 2 *line segments.*

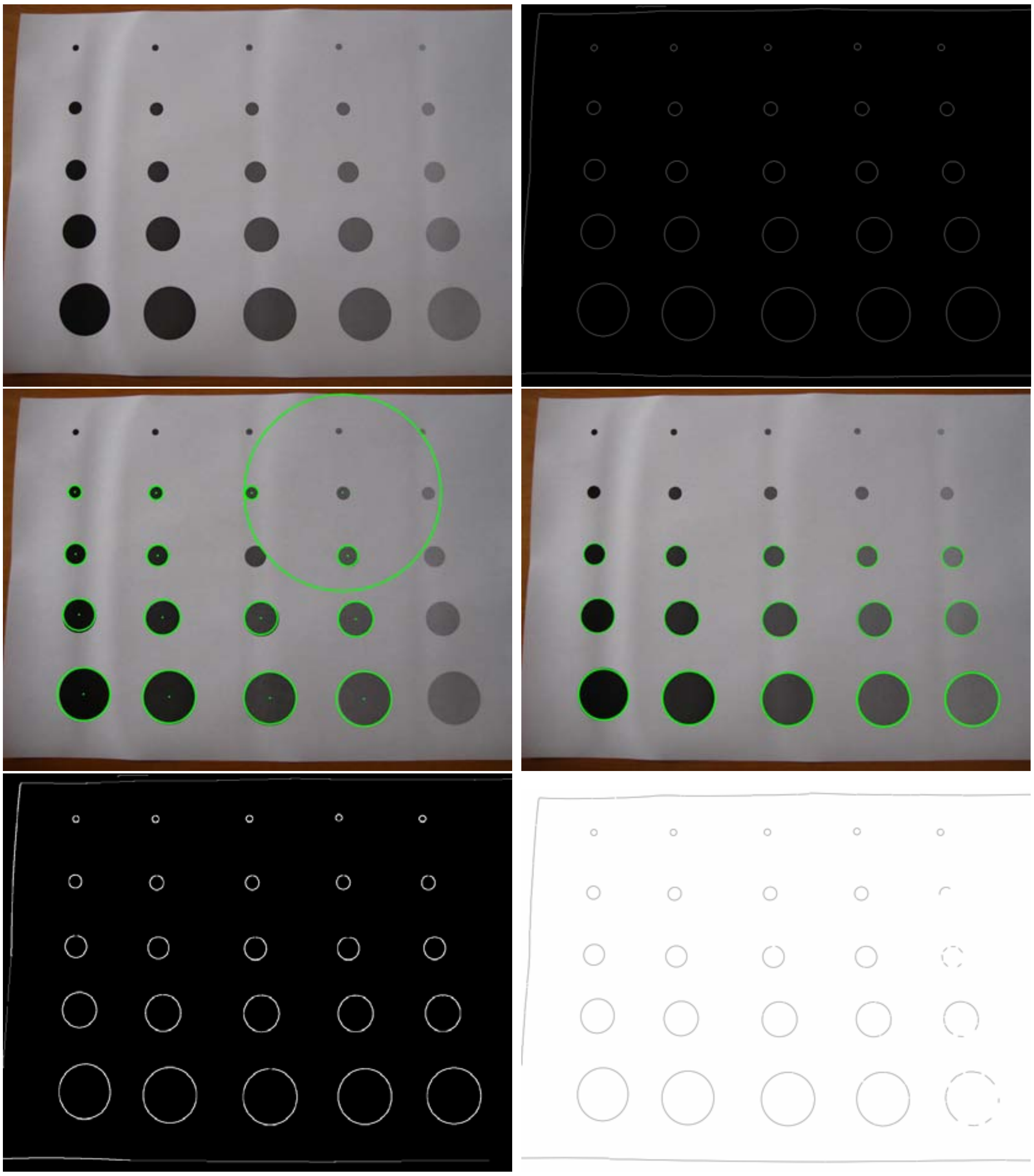


Figure 4.24: *First row: original image (*1600 × 1200 *pixels) of a set of* 25 *circles and its edge map. Second row: RHTcircle fails to detect circles of different radii and generates equally one false positive. RHTellipse detects correctly larger circles, the smaller are missed. e)Etemadi:* 26 *circular arcs, and* 29 *line segments. f) ELSD:* 8 *ellipses,* 24 *circular arcs, and* 20 *line segments. The noise affects more the orientations of pixels with low gradient (right column), breaking the convexity of the shape. Some circles are seen as a set of line segments and circular arcs.*

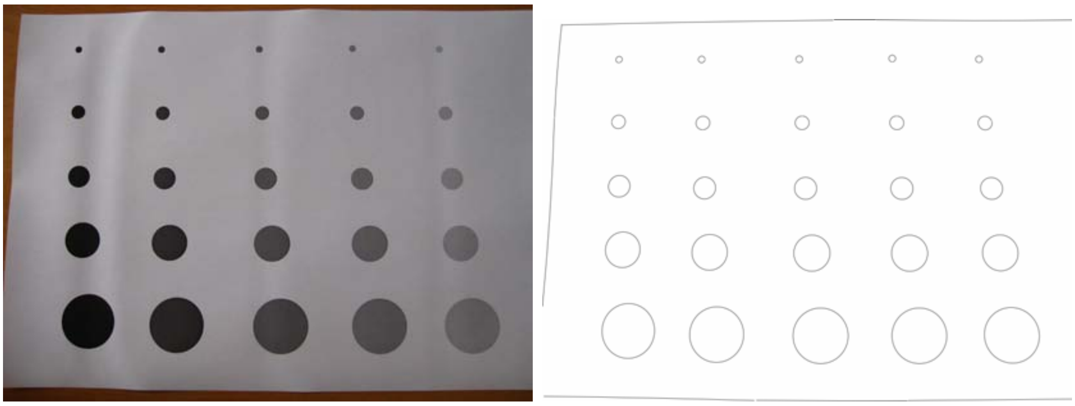


Figure 4.25: *Left: Same image (*1600 × 1200 *pixels) of a set of* 25 *circles. Right: ELSD:* 21 *ellipses,* 9 *circles, and* 3 *line segments. Before applying the detector, a Gaussian smoothing, followed by subsampling is performed, denoising the gradient orientations.*

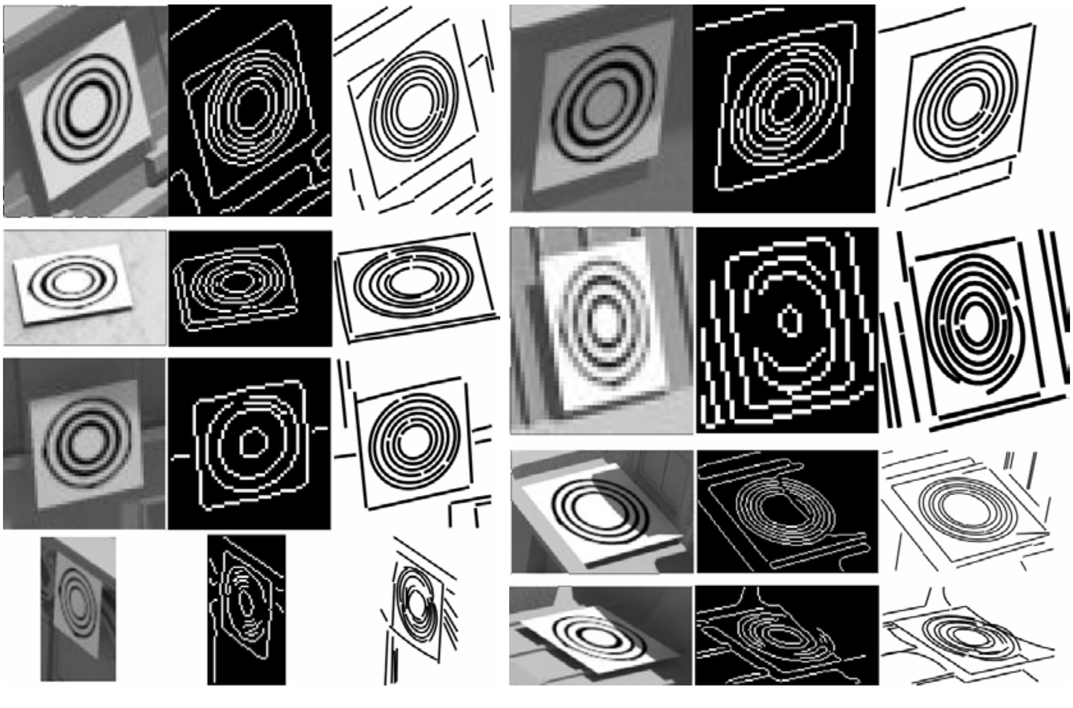


Figure 4.26: *Eight computer generated images of a marker containing concentric circles, together with their edge maps and the result of ELSD. The results of the RHTcircle/ellipse and Etemadi are not satisfactory, also (mostly) because the edge maps have poor quality.*

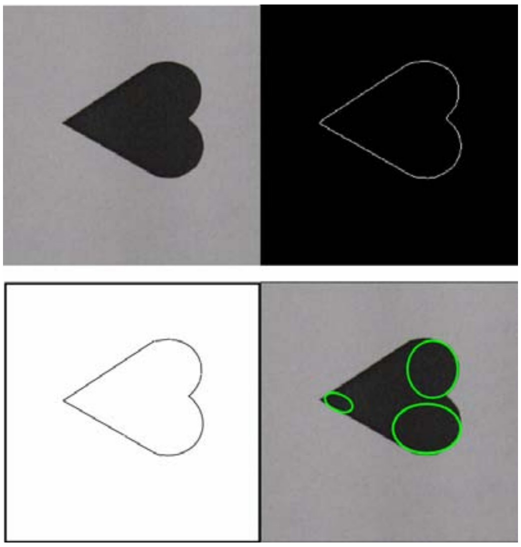


Figure 4.27: *Example of a "difficult" combination of linear and circular/elliptical shapes. First row: original image and its edge map. Second row: ELSD result (*4 *circular arcs and* 3 *line segments – the upper circular part is oversegmented) and RHTellipse result (*3 *ellipses).*

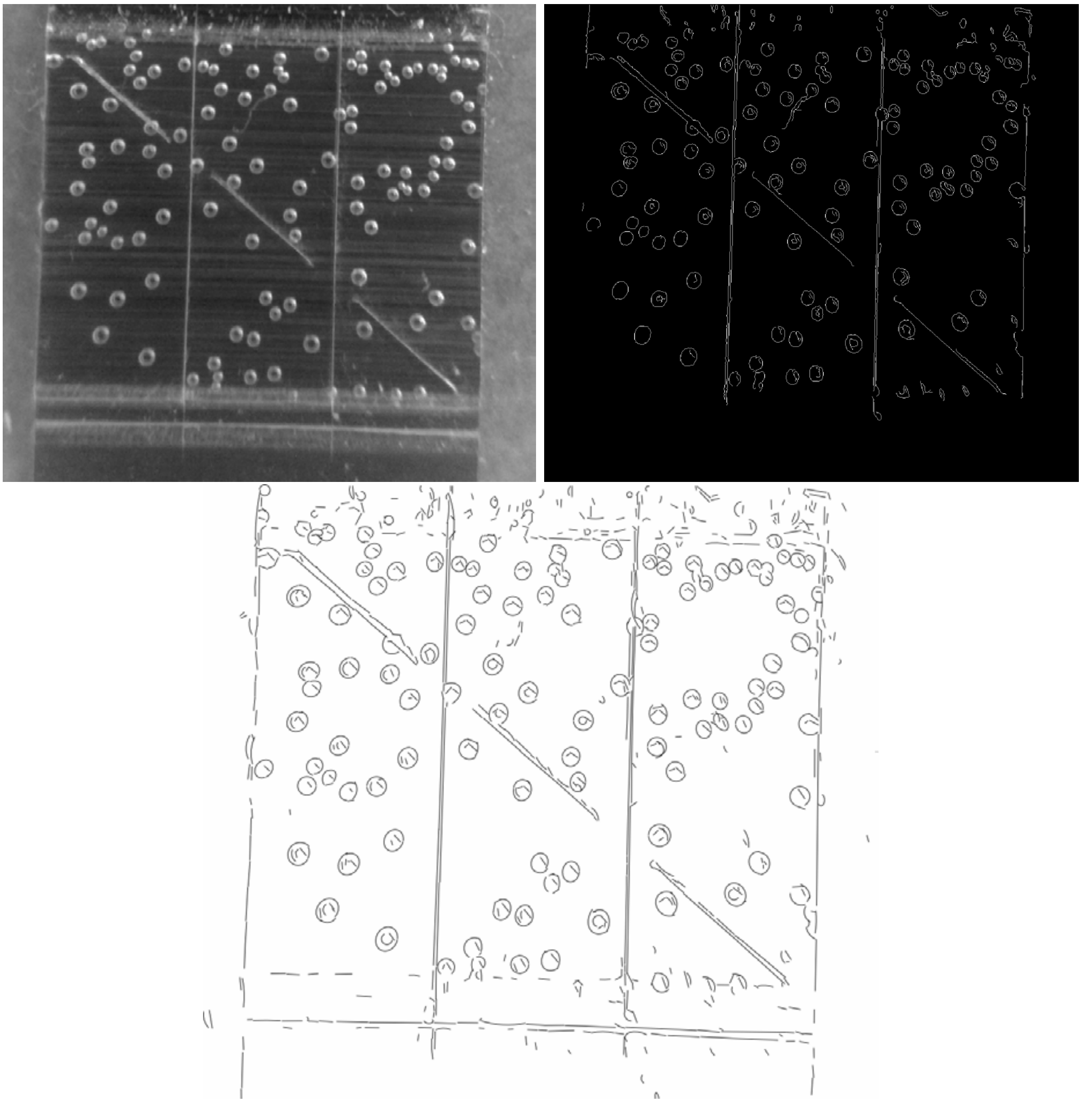


Figure 4.28: *Bubble Tag™ image (*750×670 *pixels), together with its edge map and the raw result of ELSD. RHTellipse reported no detection on this image. The results of ELSD are not completely satisfactory, and they require a post-processing step, as the bubbles are often represented by fragmented circular/elliptical arcs. This is due to the low quality of the image. ELSD execution time: ∼5s.*

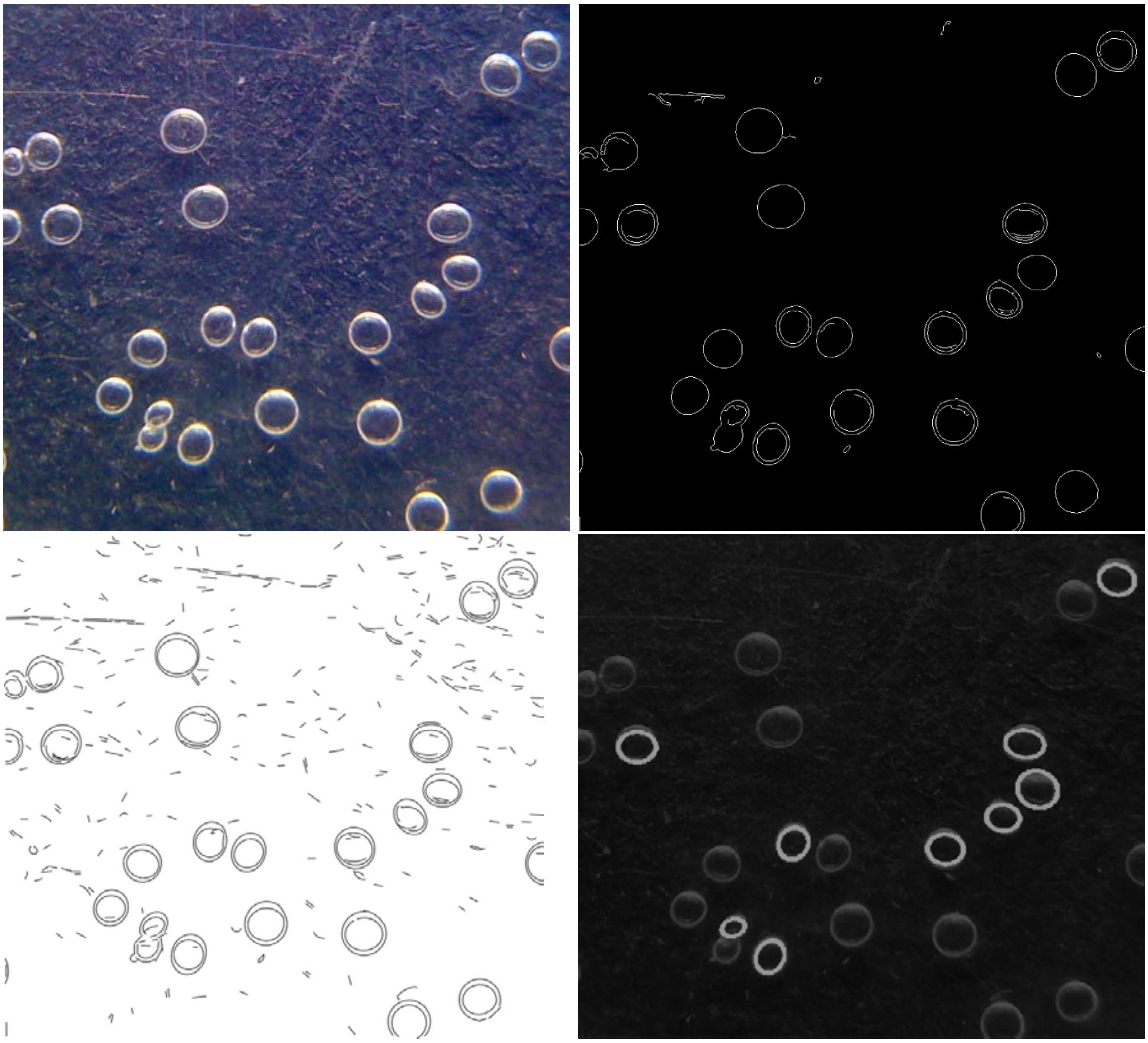


Figure 4.29: *First row: Zoom on a Bubble Tag™ image, together with its edge map. Second row: the result of ELSD and of RHTellipse.*

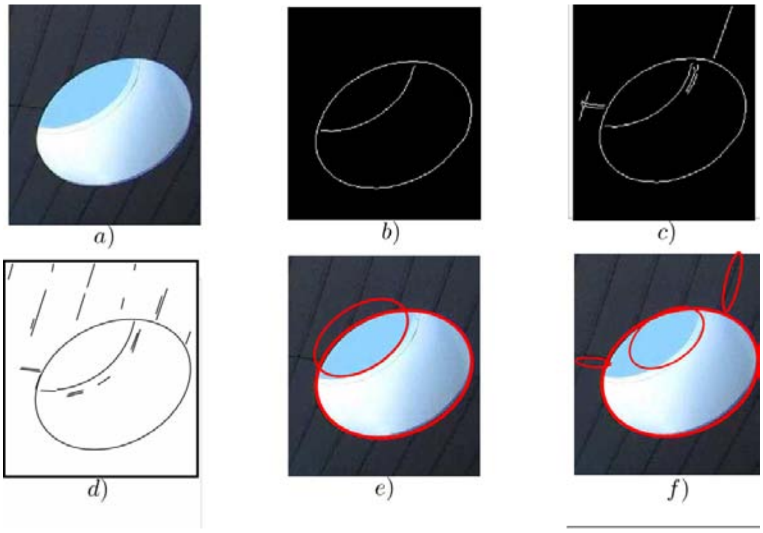


Figure 4.30: *a), b), c) Natural image and its edge maps, obtained with different thresholds for the edge detector. d) ELSD:* 1 *ellipse,* 1 *circular arc and* 18 *line segments. e) RHTellipse result obtained on the edge map from b), f) RHTellipse result obtained on the edge map from c). The parameters of the edge detector are crucial in obtaining a reliable detection result, and they need to be conveniently tuned for each (type of) image.*

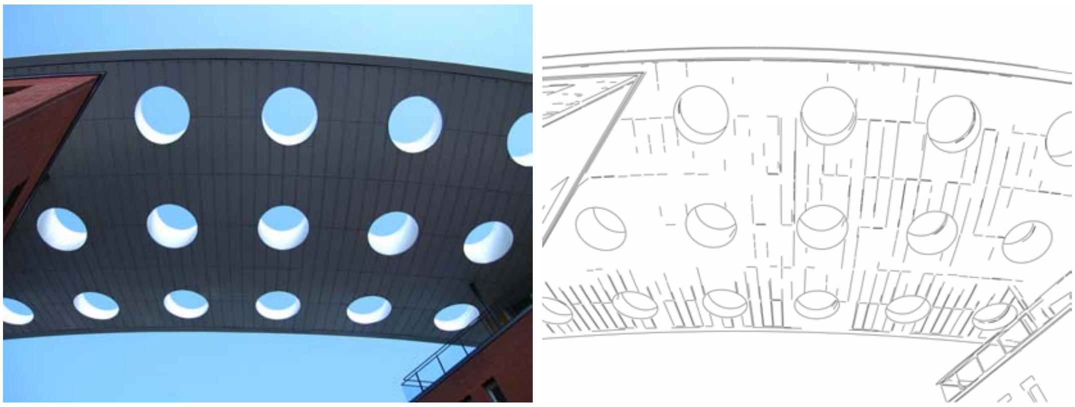


Figure 4.31: *Natural image* (1600×1200 *pixels). ELSD reports* 27 *ellipses,* 133 *circular arcs, and* 424 *line segments. Execution time:* $\sim$ *3s. RHTellipse reported no detection, even after* 1000 *iterations.*

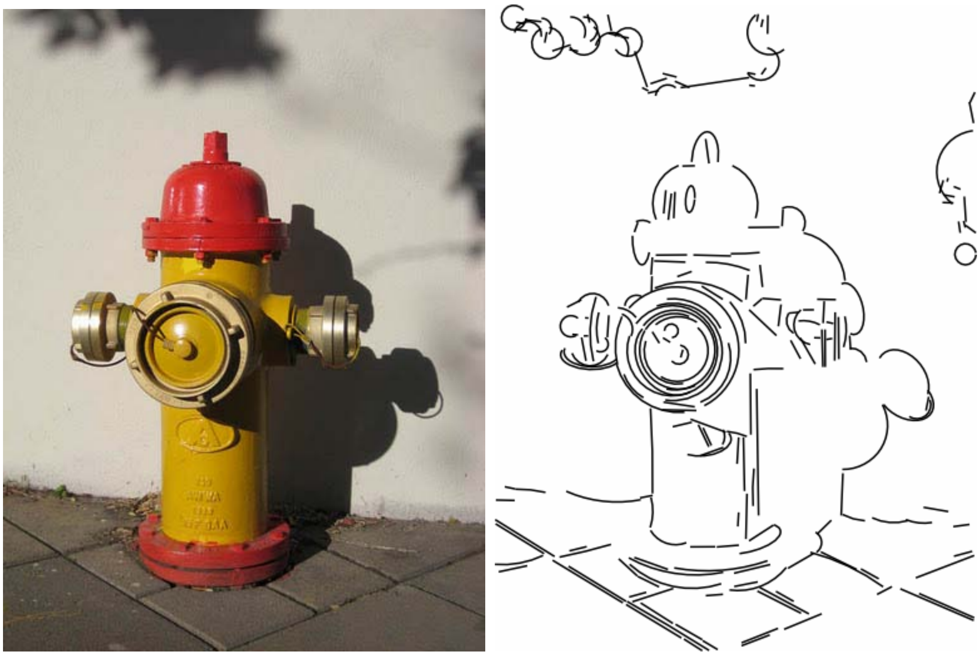


Figure 4.32: More results of *ELSD* on natural images.

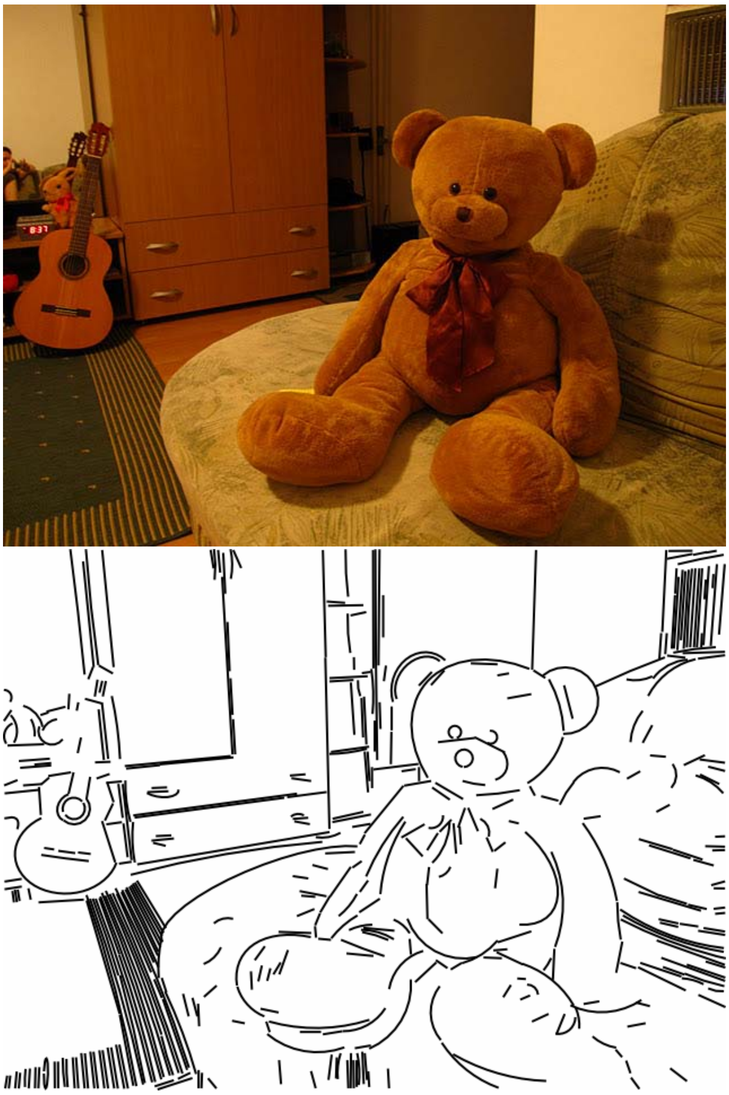


Figure 4.33: More results of *ELSD* on natural images.

Chapter 5

Bubble Tag™ Signature

Contents

5.1 Introduction

The final application of our work concerns the Bubble Tag™ identification problem, which can be stated as follows. Given an image of a Bubble Tag™, taken with a non-calibrated camera, the goal is to extract a *signature* which allows the identification of the Bubble Tag™ among a (large) database containing preregistered Bubble Tag™ images.

The previous chapter presented a parameterless ellipse detector, able to control the number of false positives. It serves as feature extractor in the image hashing process leading to Bubble Tag™ identification. In this chapter, we address the projective nature of the features extracted beforehand, namely the positions and sizes of the bubbles, *encoded* as ellipse parameters. Indeed, if we assume that bubbles are coplanar circular 2D objects[1] which undergo a projective transformation when read using a pinhole camera, the capture will yield a photography containing the projections of the circles, i.e. ellipses.

The identification process of a Bubble Tag™ comes eventually to establishing the projective equivalence of two images of the same Bubble Tag™, i.e., that the two images are related by a 2D projective transformation (homography) under the hypothesis that the images are taken with a pinhole camera. Such a projective equivalence indicates that the same bubble configuration is present in both images. More precisely, given the two images, the validity of the one-to-one correspondence will be carried out using convenient measures. It is important to mention that in a real-world scenario, this procedure would need to be repeated as many times as entries in the database containing the preregistered Bubble Tag™ images. Thus, time and memory consumption issues can occur in the process, in case of large databases.

[1] The bubbles are actually 3D entities, a quality which strengthens the difficulty of copying a Bubble Tag™ in forgery purposes (see Bubble Tag™ description in section 6). The 3rd dimension serves as means for guaranteeing the validity of a Bubble Tag™, i.e. we are not in the presence of a photography, but it will not, however, be considered in the identification process, as the 2D information is sufficient.

The one-to-one correspondence (comparison) can be assessed based on Bubble Tag™ metric properties (e.g. position/size of the bubbles, relative distance between bubbles). Nonetheless, one must take into account the distortions induced in the Bubble Tag™ images by a projective transformation[2], which is known to be non-preserving with respect to metric properties, such as lengths, angles, parallelism. Thus, two images of the same Bubble Tag™ obtained under different conditions (different cameras or different camera positions with respect to the Bubble Tag™), are not directly *comparable* from a metric point of view. Two directions can be foreseen in this situation. On the one hand, the distortions may be corrected by means of rectification techniques which recover the 2D Euclidean structure of the imaged plane, yielding comparable images. The second possibility would be to study the one-to-one correspondence using properties which are not affected by the projective transformation, i.e. *invariant-under-perspective* properties.

The first approach is closely related to camera calibration [Chen 1993, Wu 2006], and is commonly used in domains like object detection/tracking or virtual reality [Chen 2008], being justified by the need of having accurate knowledge about the objects of a scene (position, shape, size), relative to a chosen reference point, which are gained once the Euclidean structure is recovered. Numerous methods for Euclidean reconstruction have been proposed in the literature. Some approaches require the use of planar patterns [Zhang 2000] or they need several images of a rigid scene in order to accomplish this task; we speak about self-calibration in this case [Hartley 1994]. The use of projections of circular patterns is an open subject in the domain as it provides the possibility of reconstructing the scene using a single image [Gurdjos 2006].

For our work, given the large number of entries in the database, we have chosen to follow the second direction, namely the use of invariant-under-perspective properties. We will point out that it is possible to extract relevant information for Bubble Tag™ identification while bypassing some Euclidean reconstruction or metric rectification step. The proposed signature extraction technique was published in [Pătrăucean 2010b].

Chapter organisation. This chapter is organised as follows. Section 5.2 justifies the proposed approach and states the problem of identifying a Bubble Tag™ in a digital image. Section 5.3 introduces the proposed solution, based on remarkable attributes of the projective plane. The signature extraction algorithm is detailed in section 5.4. The experimental results and the conclusion are presented in section 5.5, and 5.6 respectively.

5.2 Signature Extraction Problem Statement

The problem we are dealing with is similar to a biometric system working under a "1 to many" authentication protocol. The Bubble Tag™ can be seen as *biometric* data that we assign to an object, in order to authenticate it. Thus, we have at our disposal a database containing a large number of Bubble Tag™ images, taken using non-calibrated cameras, and from different viewpoints. Then, having a new image (*query* image), the goal is to find in the database the *reference* image, i.e. the image that contains the same Bubble Tag™ as the query image. The idea is to represent the Bubble Tags™ through convenient signatures, which would allow an efficient identification.

When designing such a signature, we must keep in mind the fact that we are dealing with

[2] We assume only the geometric distortions, not the optical ones.

a relatively large database (thousands, even millions of images). Moreover, the signatures extracted from two different images of the same Bubble Tag™ will not be necessarily identical because of the changes in the shooting conditions or noise. In this context, the identification problem becomes a problem of finding the nearest neighbour, in terms of a similarity measure, which is to be defined. The obvious solution in this case would be the linear search, i.e. the query signature is compared with all the records in the database and the candidate that resembles the most is chosen as a match. The size of the database makes this method unsuitable in practice, due to time and memory issues. Alternatives could be offered by hashing techniques or k-d trees. The use of an *LSH* (*L*ocality *S*ensitive *H*ashing) approach turns out to be more efficient in order to defeat these aspects. This will be justified in chapter 6. In this context, we need to represent the database as a set P, which contains the image signatures in a *vectorial form* $p_i \in \mathbb{R}^d$. This requirement justifies the use of our solution over the classic metric rectification method [Hartley 2004, p. 55]. Indeed, the idea of comparing a rectified image with a reference "metric" image would yield 2D queries to the database containing matrix signatures, which would not be suitable for the chosen hashing procedure.

The function used to compute the vectorial (1D) signature can be expressed as $sig : f \rightarrow P \subset \mathbb{R}^d$, with f being a set of features extracted from the image. The problem of finding the nearest neighbour becomes an optimisation problem, having as objective function $\min(\mathrm{dist}(\mathrm{sig}(query), p_i))$, where dist is a distance to be adequately chosen [Andoni 2006]. Chapter 6 gives more details on the *LSH* method used in our work.

To summarise, the goal of the signature extraction process is to compute a *vectorial signature* in order to identify a Bubble Tag™ among a (large) database. Our solution does not involve passing through the Euclidean reconstruction step: based exclusively on invariant attributes of the projective plane, it produces an *invariant-under-perspective signature*.

5.3 Extraction of a Quasi-Affine Invariant Signature

In order to obtain a vectorial signature, we propose a technique for computing a pair of invariant-under-perspective values for any pair of bubbles (circles) of the Bubble Tag™, which will be applied *recursively* to the entire Bubble Tag™, yielding a *discriminant, global invariant-under-perspective signature*. The advantage of having an invariant-under-perspective signature (to which we will also refer as "quasi-affine invariant signature", or simply "invariant signature") is that it avoids the need of a metric rectification of the image.

In this section, we assume that the Bubble Tag™ can be represented as a coplanar configuration of circles in 3-space. In the sequel of this chapter, we use the syntagm "Bubble Tag™" strictly to denote such a configuration. Additionally, we suppose that the Bubble Tag™ is read using a camera which obeys the pinhole camera model, with the Bubble Tag™ lying entirely in front of the camera. Geometrically speaking, this means that all circles are in the same half-space bounded by the plane which passes through the camera centre and is parallel to the sensor plane. This is a highly realistic hypothesis as it is obviously the case that happens in practice. The image of a Bubble Tag™ under such hypotheses is obtained by a perspective transformation which is said to be *quasi-affine* with respect to the Bubble Tag™ [Hartley 2004, p. 518].

The affine Euclidean 3D plane which supports the Bubble Tag™ is embedded into a projective plane, which will be referred to as the *reference plane*. The (pure) affine 3D plane supporting the sensor, and which supports also the image of the Bubble Tag™, is embedded into another projective plane, which will be referred to as the *image plane*.

In the reference plane, the equation of any circle $\mathcal{F}$ centred at (x_c, y_c) and whose radius is r, writes in matrix form as

$$\mathbf{p}^\top \mathsf{C}_\mathcal{F} \mathbf{p} = 0, \tag{5.1}$$

with $\mathbf{p} = (x, y, 1)^\top$ being the vector of augmented Cartesian coordinates of a point and

$$\mathsf{C}_\mathcal{F} = \begin{pmatrix} 1 & 0 & -x_c \\ 0 & 1 & -y_c \\ -x_c & -y_c & x_c^2 + y_c^2 - r^2 \end{pmatrix} \tag{5.2}$$

being the matrix of the circle with respect to the reference plane representation.

The equation of the perspective projection of a circle, under the hypothesis of a pinhole camera, writes as:

$$\tilde{\mathsf{C}}_\mathcal{F} \sim \mathsf{H}^{-\top} \mathsf{C}_\mathcal{F} \mathsf{H}^{-1}, \tag{5.3}$$

where $\tilde{\mathsf{C}}_\mathcal{F}$ is the matrix of the image of the circle with respect to the image plane representation, $\sim$ is the projective equality, and H is the 3×3 matrix of the *quasi-affine homography* mapping points on the reference plane to their images on the image plane.

Throughout this chapter, when referring to vectors/matrices of entities with respect to the image plane representation, we will systematically add the symbol $\tilde{}$, as in (5.3).

$\tilde{\mathsf{C}}_\mathcal{F}$ in (5.3) could represent the matrix of an ellipse, of a hyperbola or of a parabola with respect to the affine representation of the image plane. In our case, under the hypothesis of a quasi-affine homography H, we implicitly assume that H maps circles to ellipses [Hartley 2004, p. 515].

Since classic metric properties of *one* circle are not preserved under perspective projection, we focus our attention on relative properties of *two* circles and we will show that some of them (e.g. relative distance, relative radius) are preserved, thus they are invariant. To this end, studying the generalised eigenvalues of the matrices of the images of a pair of circles proves to be useful because they *encode* information regarding the (Euclidean) relative position of the two conics [Gurdjos 2006].

Generalised Eigenvalues and Absolute Signature

Let us consider a pair of circles $(\mathcal{F}_1, \mathcal{F}_2)$ as generators of a *pencil of circles*, that we denote by $\{\mathcal{F}_1, \mathcal{F}_2\}$. It represents the 1D linear family of circles with matrices of the form $\mathsf{C}(\lambda) = \mathsf{C}_1 - \lambda \mathsf{C}_2$, where C_1 and C_2 are the Euclidean matrices of $\mathcal{F}_1$ and $\mathcal{F}_2$ respectively, and $\lambda \in \mathbb{C} \cup \{\infty\}$ is a parameter, with the convention $\mathsf{C}(\infty) = \mathsf{C}_2$. The three real solutions $\lambda_1, \lambda_2, \lambda_3$ for the unknown λ in the characteristic equation

$$\det(\mathsf{C}_1 - \lambda \mathsf{C}_2) = 0 \tag{5.4}$$

represent the *generalised eigenvalues* of $(\mathsf{C}_1, \mathsf{C}_2)$ [Golub 1996, p. 375].

The pencil $\{\mathcal{F}_1, \mathcal{F}_2\}$ contains three degenerate circles[3] $\mathcal{D}_k, k \in \{1,2,3\}$, whose parameters λ are the generalised eigenvalues $\lambda_k, k = \{1,2,3\}$ of $(\mathsf{C}_1, \mathsf{C}_2)$. Let D_k be the Euclidean matrix associated to $\mathcal{D}_k$. Thus,

$$\mathsf{D}_k = \mathsf{C}_1 - \lambda_k \mathsf{C}_2, k = \{1,2,3\} \tag{5.5}$$

.

The generalised eigenvalues allow the definition of the *absolute signature* Σ of a degenerate conic $\mathcal{D}_k$. This signature is given by

$$\Sigma(\mathsf{D}_k) = |\eta - \nu|, \tag{5.6}$$

where η and ν count the (strictly) positive and the negative (ordinary) eigenvalues μ of D_k, i.e. the solutions of the equation $\det(\mathsf{D}_k - \mu \mathsf{I}_3) = 0$, where I_3 is the 3×3 identity matrix. When D_k is a complex matrix, then we will use the convention $\Sigma(\mathsf{D}_k) = \infty$.

The same definitions apply seamlessly for the image of the pencil $\{\mathcal{F}_1, \mathcal{F}_2\}$ under H, keeping in mind that the degenerate members of the image of the pencil are the images of the degenerate members of the pencil i.e.,

$$\tilde{\mathsf{C}}_1 - \tilde{\lambda}_k \tilde{\mathsf{C}}_2 \sim \mathsf{H}^{-\top} \mathsf{D}_k \mathsf{H}^{-1},$$

where $\tilde{\lambda}_k$, $k = \{1,2,3\}$, denote the generalised eigenvalues of $(\tilde{\mathsf{C}}_1, \tilde{\mathsf{C}}_2)$.

Remarkable Invariant-under-Perspective Properties

Considering the notions previously defined, namely the generalised eigenvalues of a pair of matrices and the absolute signature of a degenerate conic, we enunciate two invariant attributes P_1, P_2 of the reference plane, onto which the proposed technique is grounded:

P_1 The set of the generalised eigenvalues of the matrix pair $(\mathsf{C}_1, \mathsf{C}_2)$ associated with two general conics $\mathcal{F}_1$ and $\mathcal{F}_2$ is invariant under projective transformations, including perspectivities, up to a scale factor. In other words, if $\boldsymbol{\lambda}$ and $\tilde{\boldsymbol{\lambda}}$ denote the vectors whose components are the generalised eigenvalues of the matrix pairs $(\mathsf{C}_1, \mathsf{C}_2)$ and $(\tilde{\mathsf{C}}_1, \tilde{\mathsf{C}}_2)$ respectively (C and $\tilde{\mathsf{C}}$ being linked by (5.3)), then $\tilde{\boldsymbol{\lambda}} \sim \mathsf{P}\boldsymbol{\lambda}$ where P is some permutation (order-3) matrix.

P_2 The absolute signature Σ of a degenerate conic $\mathcal{D}$ is invariant under perspective. In other words, the absolute signatures of a degenerate conic and of its images under projective transformations are the same.

A proof of P_1 can be found in [Mundy 1992]. The proof of P_2 is a straightforward consequence of Sylvester's law of inertia [Golub 1996, p. 403], as it can be easily shown that the absolute signature $\Sigma(\mathsf{D}_k)$ is invariant under congruence transformations of $\Sigma(\mathsf{D}_k)$ i.e., $\Sigma(\mathsf{D}_k) = \Sigma(\mathsf{T}^\top \mathsf{D}_k \mathsf{T})$, for any non-singular $\mathsf{T} \in \mathbb{R}^{3\times 3}$.

Using the properties P_1 and P_2, we will introduce a procedure for computing two invariant values for a pair of circles. Afterwards, we will apply it in a *recursive* manner for the entire configuration of circles, leading to an *invariant signature*.

[3] A degenerate conic $\mathcal{D}$ consists of either two lines having $\mathbf{m}$ and $\mathbf{n}$ as vectors, thus $\mathsf{D} \sim \mathbf{m}\mathbf{n}^\top + \mathbf{n}\mathbf{m}^\top$, or a repeated line $\mathbf{m}$, such that $\mathsf{D} \sim \mathbf{m}\mathbf{m}^\top$.

Relative Position and Relative Radius of Two Circles

Let us consider a pair of coplanar circles $(\mathcal{F}_1, \mathcal{F}_2)$ represented by the matrix-pair $(\mathsf{C}_1, \mathsf{C}_2)$ in the reference plane. We start here with the symbolic computation of the generalised eigenvalues of $(\mathsf{C}_1, \mathsf{C}_2)$ according to (5.4). In order to simplify the computation and without loss of generality, let us attach some Euclidean affine representation to the reference plane, in whose bases $\mathcal{F}_1$ and $\mathcal{F}_2$ have matrices:

$$\mathsf{C}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -r_1^2 \end{pmatrix}, \quad \mathsf{C}_2 = \begin{pmatrix} 1 & 0 & -\delta \\ 0 & 1 & 0 \\ -\delta & 0 & \delta^2 - r_2^2 \end{pmatrix}. \tag{5.7}$$

Thus, $\mathcal{F}_1$ is a circle of radius $r_1 > 0$, centred at the origin O; $\mathcal{F}_2$ is a circle of radius $r_2 > 0$, centred at Cartesian coordinates $(0, \delta)$, with $\delta \geq 0$.

Using Maple to solve (5.4), we give now the symbolic expression of $\boldsymbol{\lambda} \in \mathbb{C}^3$, the vector whose components are the generalised eigenvalues of $(\mathsf{C}_1, \mathsf{C}_2)$ as functions of δ, r_1 and r_2:

$$\boldsymbol{\lambda} = \left(\frac{1}{2} \frac{\alpha + \sqrt{\beta}}{r_2^2}, \ \frac{1}{2} \frac{\alpha - \sqrt{\beta}}{r_2^2}, \ 1 \right)^{\top}, \tag{5.8}$$

$$\text{where} \quad \alpha \equiv r_1^2 + r_2^2 - \delta^2 \in \mathbb{R}, \tag{5.9}$$

$$\beta \equiv (\delta + r_1 + r_2)(\delta + r_1 - r_2)(\delta - r_1 + r_2)(\delta - r_1 - r_2) \in \mathbb{R}. \tag{5.10}$$

Note that λ_1 and λ_2 are complex conjugate in their most general form. They cannot be zero since some algebra yields $\lambda_1 \lambda_2 = r_1^2 / r_2^2 > 0$ and they are different from one if $r_1^2 \neq r_2^2$.

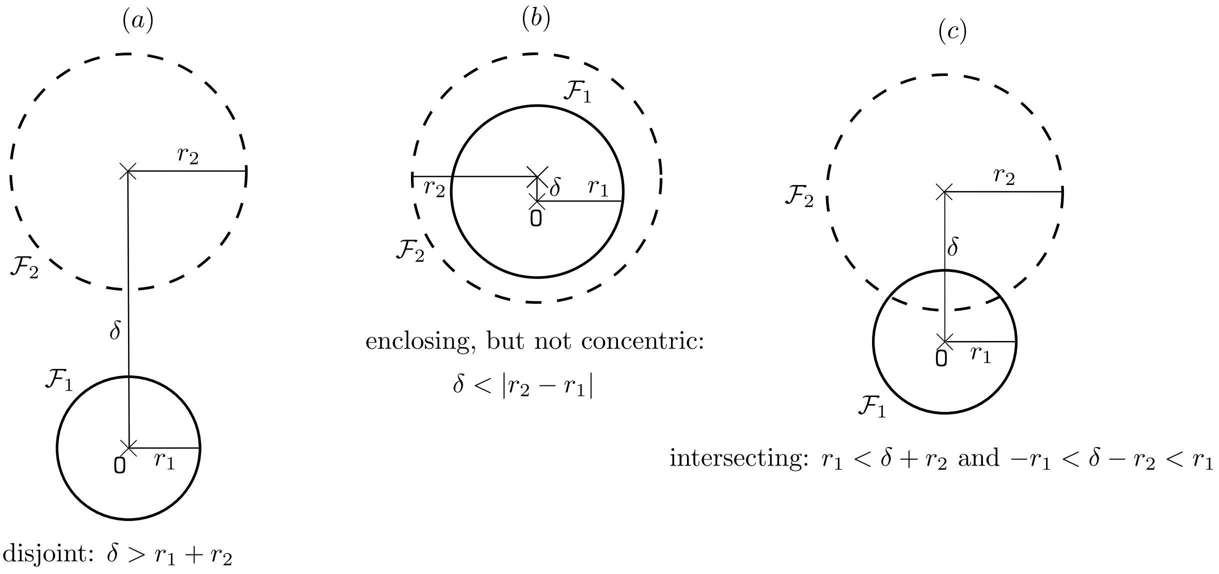
The fact that λ_1 and λ_2 are complex or not depends on the sign of β, so let us now investigate in which case this happens. When circles have zero, one or two real finite points in common, we will say that circles are ***disjoint***, ***tangent*** or ***intersecting***, respectively. The configurations of disjoint circles and tangent circles include the case where one circle encloses the other one. It can be shown that

$$\mathcal{F}_1 \text{ and } \mathcal{F}_2 \text{ are} \begin{cases} \text{intersecting,} \\ \text{tangent,} \\ \text{disjoint,} \end{cases} \text{if and only if} \begin{cases} \beta < 0, \\ \beta = 0, \\ \beta > 0. \end{cases}$$

As a consequence, it is now easy to establish that the first two generalized eigenvalues in (5.8) are complex if and only if the circles are intersecting (i.e., $\beta < 0$). The proof of it does not show any special difficulty and follows directly from the constraints on d, r_1 and r_2 corresponding to the relative positions of two circles as illustrated in figure 5.1. In particular,

- disjoint circles can be separate: $\delta > r_1 + r_2$ (figure 5.1 left), or enclosing, but not concentric: $\delta < |r_2 - r_1|$ (figure 5.1 middle);

- the condition for intersecting circles is $-r_1 < \delta - r_2 < r_1$ and $r_1 < \delta + r_2$ (figure 5.1 right) and this also yields the condition for β to be negative.



disjoint: $\delta > r_1 + r_2$

enclosing, but not concentric: $\delta < |r_2 - r_1|$

intersecting: $r_1 < \delta + r_2$ and $-r_1 < \delta - r_2 < r_1$

Figure 5.1: *Relative position of two circles.*

Let us now introduce the following notations:

$$d = \frac{\delta}{r_1}, \quad r = \frac{r_2}{r_1}. \tag{5.11}$$

Hence, d denotes the relative distance (w.r.t. the radius r_1 of the first circle) and r denotes the relative radius.

The key point here, following the work of Gurdjos et al. in [Gurdjos 2006], is that one can recover the pair (d, r) from the generalised eigenvalues of $(\tilde{\mathsf{C}}_1, \tilde{\mathsf{C}}_2)$, where $\tilde{\mathsf{C}}_1$ and $\tilde{\mathsf{C}}_2$ are the matrices of the images of the circles $\mathcal{F}_1$ and $\mathcal{F}_2$. For this reason, we will say that (d, r) is a double invariant of two circles.

(d, r) – a double invariant of two circles

Now let us compute the generalised eigenvalues and eigenvectors of $(\tilde{\mathsf{C}}_1, \tilde{\mathsf{C}}_2)$. Considering (5.3), we do not need Maple anymore since we can make use of the property P_1 and write:

$$\tilde{\boldsymbol{\lambda}} = s\boldsymbol{\lambda}, \tag{5.12}$$

where s is a scale factor. Let $\tilde{\boldsymbol{\lambda}} = (\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3)^\top$. By eliminating s in (5.12) we obtain:

$$\begin{cases} (\tilde{\lambda}_1 + \tilde{\lambda}_2)/\lambda_3 = (r^2 - d^2 + 1)/r^2 \\ (\tilde{\lambda}_1 - \tilde{\lambda}_2)/\lambda_3 = \sqrt{\beta}/r^2. \end{cases} \tag{5.13}$$

Now if we solve the system (5.13) for d and r and choose the positive solutions, we get:

$$\begin{cases} d = \sqrt{\tilde{\lambda}_1\tilde{\lambda}_2(\tilde{\lambda}_1 - \tilde{\lambda}_3)(\tilde{\lambda}_2 - \tilde{\lambda}_3)}/|\tilde{\lambda}_1\tilde{\lambda}_2| \\ r = |\tilde{\lambda}_3|/\sqrt{\tilde{\lambda}_1\tilde{\lambda}_2}. \end{cases} \tag{5.14}$$

The equation (5.14) has a major importance, because it expresses the pair (d, r) (unknown if we do not have a metric reconstruction of the image plane) as a function of the generalised eigenvalues $\tilde{\boldsymbol{\lambda}}$. The latter can be easily computed by using a method of ellipse detection in images and by solving the equation (5.4).

Nevertheless, a crucial aspect is the separation of the generalised eigenvalues $\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3$. Using some symbolic computations, the following proposition is stated to solve this issue.

Proposition 1 *There are three configurations to be considered regarding the signatures of degenerate circles.*

- *When* $\delta = 0$ (concentric circles)*, we have:*

$$\Sigma(\tilde{\mathsf{C}}_1 - \tilde{\lambda}_1\tilde{\mathsf{C}}_1) = 2 \text{ and } \Sigma(\tilde{\mathsf{C}}_1 - \tilde{\lambda}_2\tilde{\mathsf{C}}_2) = \Sigma(\tilde{\mathsf{C}}_1 - \tilde{\lambda}_3\tilde{\mathsf{C}}_3) = 1. \tag{5.15}$$

- *When* $\beta < 0$ (intersecting circles)*, we have:*

$$\Sigma(\tilde{\mathsf{C}}_1 - \tilde{\lambda}_1\tilde{\mathsf{C}}_2) = \Sigma(\tilde{\mathsf{C}}_1 - \tilde{\lambda}_2\tilde{\mathsf{C}}_2) = \infty \text{ and } \Sigma(\tilde{\mathsf{C}}_1 - \tilde{\lambda}_3\tilde{\mathsf{C}}_2) = 0. \tag{5.16}$$

- *When* $\beta \geq 0$ (other configurations)*, we have:*

$$\Sigma(\tilde{\mathsf{C}}_1 - \tilde{\lambda}_1\tilde{\mathsf{C}}_2) = \Sigma(\tilde{\mathsf{C}}_1 - \tilde{\lambda}_2\tilde{\mathsf{C}}_2) = 2 \text{ and } \Sigma(\tilde{\mathsf{C}}_1 - \tilde{\lambda}_3\tilde{\mathsf{C}}_2) = 0. \tag{5.17}$$

Proof. To simplify the symbolic computations and since absolute signatures are projectively invariant (cf. P_2), we can investigate the absolute signatures of degenerate circles through their matrices $\mathsf{D}_k = \mathsf{C}_1 - \lambda_k\mathsf{C}_2$, $k \in \{1..3\}$, in Euclidean representation, scaled by some adequate factor ε_k. We will use[4] $\varepsilon_j \equiv 2/\lambda_j$, $j \in \{1, 2\}$ and $\varepsilon_3 \equiv 2$.

The absolute signature of $\varepsilon_k\mathsf{D}_k$, denoted by Σ_k here, can be inferred from the sign of the product of its two non-zero[5] (ordinary) eigenvalues, denoted by $\Theta_1^{\lambda_k}$ and $\Theta_2^{\lambda_k}$. If $\Theta_1^{\lambda_k}$ and $\Theta_2^{\lambda_k}$ have the same sign, then $\Sigma_k = 2$; otherwise $\Sigma_k = 0$. Let us start with Σ_3. Maple returns:

$$\Theta_1^{\lambda_3} = \gamma + \sqrt{\gamma^2 + 4d^2}, \quad \Theta_2^{\lambda_3} = \gamma - \sqrt{\gamma^2 + 4d^2}, \tag{5.18}$$

$$\text{where } \gamma \equiv r^2 - d^2 - 1 \in \mathbb{R}. \tag{5.19}$$

By expanding the product $\Theta_1^{\lambda_3}\Theta_2^{\lambda_3} = -4d^2 \leq 0$, we conclude that $\Theta_1^{\lambda_3}$ and $\Theta_2^{\lambda_3}$ *always* have opposite signs i.e., $\Sigma_3 = 0$ except if circles are concentric, and in this case we have $\Sigma_3 = 1$.

[4] Remind that λ_1 and λ_2 cannot be zero.

[5] If we compute the eigenvalues of $\varepsilon_k\mathsf{D}_k$ using Maple, we get symbolic expressions for two of them and 0 for the other one. Non-zero eigenvalues here refer to the first two.

Now, let us carry on with Σ_1 and Σ_2. Maple returns:

$$\Theta_1^{\lambda_1} = 2\gamma, \quad \Theta_2^{\lambda_1} = \gamma + \sqrt{\beta}, \tag{5.20}$$

$$\Theta_1^{\lambda_2} = 2\gamma, \quad \Theta_2^{\lambda_2} = \gamma - \sqrt{\beta}. \tag{5.21}$$

- Assume $\delta = 0$ (concentric circles). Then (5.20) simplifies to $\Theta_1^{\lambda_1} = \Theta_2^{\lambda_1} = 2(r^2 - 1) \neq 0$, while (5.21) simplifies to $\Theta_1^{\lambda_2} = 2(r^2 - 1) \neq 0$ and $\Theta_2^{\lambda_2} = 0$. Hence $\Sigma_1 = 2$ and $\Sigma_2 = 1$, and since $\Sigma_3 = 1$ in this case as explained above, the proof is ended for (5.15).

- Assume $\beta < 0$. As shown above, $\Sigma_3 = 0$. All the above eigenvalues are complex, so the signatures Σ_1 and Σ_2 are undefined ($\Sigma_1 = \Sigma_2 = \infty$), which ends the proof for (5.16).

- Assume $\beta \geq 0$. All eigenvalues are now real. It is worthy of note that they cannot be zero. Indeed, $\gamma = 0$ holds if and only if $r = \pm\sqrt{d^2 + 1}$. Substituting r in (5.10), we get $\beta = -4d^2 < 0$, whatever the sign, which is contrary to the assumption. We cannot have $\gamma = \pm\sqrt{\beta}$ neither, as this would entail that $\gamma^2 - \beta = 0$. If we solve this equation for d, we obtain $d = 0$ which is also impossible since circles cannot be concentric. As a result, both $\Theta_1^{\lambda_k}$ and $\Theta_2^{\lambda_k}$ are non zero.

  We can claim that $sign(\Theta_2^{\lambda_1}) = sign(\Theta_2^{\lambda_2})$ i.e., $\Theta_2^{\lambda_1}\Theta_2^{\lambda_2} = (\gamma + \sqrt{\beta})(\gamma - \sqrt{\beta}) = \gamma^2 - \beta > 0$, since Maple simplifies it to $4d^2 > 0$, keeping in mind that d cannot be zero. We can now prove that $\Sigma_j = 2$ i.e., $sign(\Theta_1^{\lambda_j}) = sign(\Theta_2^{\lambda_j})$, for $j = \{1, 2\}$. If $\gamma > 0$ then $sign(2\gamma) = sign(\gamma + \sqrt{\beta})$. If $\gamma < 0$ then $sign(2\gamma) = sign(\gamma - \sqrt{\beta})$. Since $sign(\gamma + \sqrt{\beta}) = sign(\gamma - \sqrt{\beta})$ in any case, this means that $\Sigma_1 = \Sigma_2 = 2$, which ends the proof for (5.17).

■

Denote by $\tilde{\lambda}_-$ and $\tilde{\lambda}_+$ the two general eigenvalues that have symmetrical roles in (5.14), i.e. $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$, and by $\mathcal{D}_-$ and $\mathcal{D}_+$ the corresponding degenerate conics, represented through the projective matrices $\tilde{\mathsf{D}}_-$ and $\tilde{\mathsf{D}}_+$ as defined in (5.5). The third degenerate conic will be represented by $\tilde{\mathsf{D}}_3$. Considering both Proposition 1 and the invariance of the absolute signature (P_2), we eventually obtain the important inequality:

$$\Sigma(\tilde{\mathsf{C}}_1 - \tilde{\lambda}_\pm\tilde{\mathsf{C}}_2) \geq 1 \geq \Sigma(\tilde{\mathsf{C}}_1 - \tilde{\lambda}_3\tilde{\mathsf{C}}_2). \tag{5.22}$$

The inequality (5.22) is a projective invariant that allows establishing the distinction between the three generalised eigenvalues. If the generalised eigenvalues $\tilde{\lambda}_1, \tilde{\lambda}_2, \tilde{\lambda}_3$ are separated so that (5.22) stands, then d and r, computed according to (5.14), are invariant under perspective, in virtue of P_1.

Therefore, having the projective representation of the circles $(\mathcal{F}_1, \mathcal{F}_2)$, we can obtain a double invariant (d, r), which encodes metric measures of the two circles (see algorithm 5).

5.4 Signature Extraction Algorithm

The previous section detailed a technique for computing a couple of invariant values for two coplanar circles. This result will be used jointly with an additional invariant-under-perspective

```
Input: The two matrices $\tilde{\mathsf{C}}_1$ and $\tilde{\mathsf{C}}_2$ representing the images of two coplanar circles.
Output: The pair $(d, r)$ yielding the relative position and radius of the circles (see text).
$\tilde{\lambda}_{k\in\{1,2,3\}}$ = generalised_eigenvalues $(\tilde{\mathsf{C}}_1, \tilde{\mathsf{C}}_2)$;
for $k = 1..3$ do
    compute the (ordinary) eigenvalues of $\tilde{\mathsf{D}}_k = \tilde{\mathsf{C}}_1 - \tilde{\lambda}_k \tilde{\mathsf{C}}_2$;
    compute $\Sigma(\tilde{\mathsf{D}}_k) = |\eta - \nu|$, where $\eta$ and $\nu$ count the positive and negative eigenvalues;
end
sort $\tilde{\lambda}_{k\in\{1,2,3\}}$ such that the inequality (5.22) stands;
rename $\tilde{\lambda}_1$ and $\tilde{\lambda}_2$ as $\tilde{\lambda}_+$ and $\tilde{\lambda}_-$;
compute $(d, r)$ according to (5.14);
```

Algorithm 5: Computation of the double invariant (d, r)

property P_3, in order to apply the proposed procedure to the entire configuration of coplanar circles.

P_3 The convex hull of a planar set of points is invariant under quasi-affine transformation [Hartley 2004, p. 515].

The proposed algorithm represents a Bubble Tag™ through its successive convex layers (whose structures are invariant under perspective, according to P_3), as shown in figure 5.2. Afterwards, it computes the values (d, r) given by (5.14) for every couple of neighbour circles in the convex layers. Eventually, the signature of a Bubble Tag™ is obtained as the concatenation of the pairs (d, r).

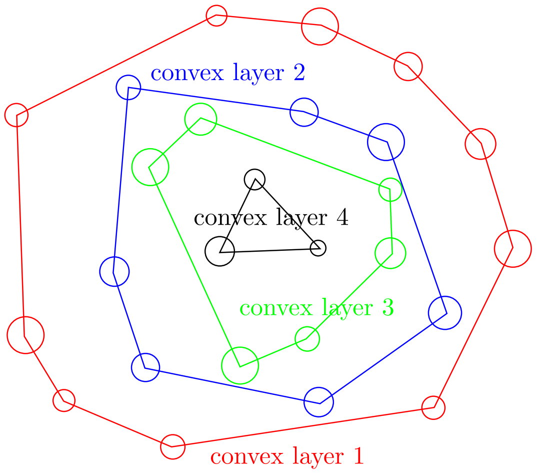


Figure 5.2: *Bubble Tag™ representation.*

Therefore, the proposed signature extraction technique is based exclusively on invariant-under-perspective properties P_1, P_2, P_3. Thus, the signature itself will be invariant under perspective, yielding the metric rectification no longer necessary.

The algorithm aims to identify a Bubble Tag™ through progressive elimination of potential candidates. The signature will contain several *subsignatures*, one for each convex layer. The query of the database is to be done starting with the subsignature associated to the external convex layer. At the end of this step we keep as potential candidates only the n nearest neighbours. The process is repeated using as query the next inferior convex layers until we get a match that is at a distance $d < d_{min}$.

The main steps of the identification procedure are given in algorithm 6.

```
Input: Bubble Tag™ image $I$
Output: The corresponding match from the database
Detect ellipses in $I$;
Compute the convex layers $c_i$ of the ellipse centres;
foreach $c_i$ do
    foreach pair of ellipses $(\mathcal{F}_{ij}, \mathcal{F}_{ij+1})$ do
        compute $(d_j, r_j)$;
    end
    compute the cyclic permutations of $c_i$;
    query the database;
    if $d < dmin$ then
        return match;
    end
end
```

Algorithm 6: Bubble Tag™ Identification

The identification algorithm starts with the ellipse detection in the query image. Then it computes the structures of the convex layers using the *qhull* algorithm [Barber 1996]. The layers are processed successively and the algorithm stops as soon as the correct match has been found. The query of the database is done considering all the cyclic permutations of the subsignatures, in order to handle the rotation problem.

Open problem. The use of convex layers yields the proposed procedure less robust to noise, as the structure of the convex layers may change due to small errors in ellipse detection. This may result eventually in considerable changes in the final signature. The literature does not reflect a high interest on this issue, as the convex hulls are used, for example, to measure the area/perimeter of the domain covering a given set of points, thus the exact composition of the convex hull is less important. In [Takahashi 2010], Takahashi and Kudo address the separability of the convex hulls in a classification problem. The identified classes are delimited by their convex hulls. They introduce the concept of *expanded convex hull*, in order to widen the distance between the identified classes. We have explored this idea in our context, for the Euclidean representation of a Bubble Tag™, but the results were rather modest. Moreover, this approach does not fit seamlessly to our problem, as the computation of the *expanded convex hulls* relies on metric properties (e.g. distances between convex hulls) which are not invariant under perspective.

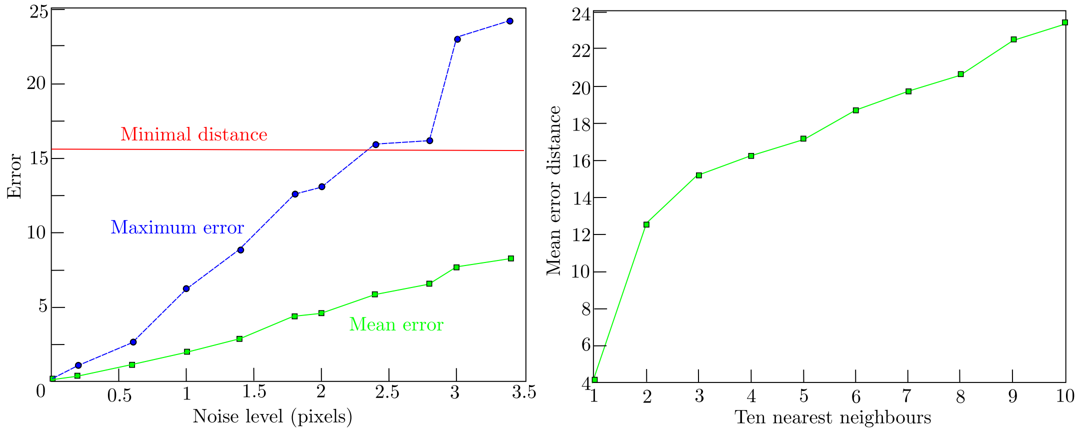


Figure 5.3: *Left: Robustness against noise evaluated on computer-generated images. Red line - minimal distance between two stored subsignatures of external convex layers. Right: Mean distances between the external convex layers of a query "coin tag" and the ten nearest neighbours. In all cases, the nearest neighbour found was the correct match.*

5.5 Experimental Results

The proposed signature extraction procedure is, for the time being, a proof of concept. We do not have the technological means needed to acquire readable projective views of the Bubble Tag™. The results that we present were obtained using computer-generated images and images of "coin tags" (figure 5.4).

Tests carried out on computer-generated images. We have used a database containing 300 bubble tags, randomly generated. Each image (size 1200 × 800 pixels) contains a random number of circles (between 80 and 110), with radii of 10 to 20 pixels. The simulation conditions were chosen to imitate as well as possible the real bubble configurations. In order to obtain the perspective projection corresponding to the image shooting, we have generated random homographies, such that the circles would be transformed into ellipses. The tests seek to evaluate the robustness against noise. The results are shown in figure 5.3 left. For every noise level, we have considered 100 experiments and we have computed the mean error and the maximal error induced by noise. For every test we have retained the ten nearest neighbours found in the database, using as similarity measure the Euclidean distance. The results show that up to a noise level inferior to 2.5 pixels, the first convex layer is sufficient to correctly identify the Bubble Tag™. For a noise level superior to this threshold, the correct match has been found using the first two convex layers. The threshold value depends on the minimal distance between two records from the database.

Tests carried out on "coin tags". We have considered 50 different configurations of coins and we have taken two images for each of them (example figure 5.4). One image is used as reference to be stored in the database and the other one is used as query image.

Figure 5.3 right shows the mean distance to the 10 nearest neighbours (over 50 tests), found in the database. The correct match was returned in every case. The matching results given in this section were obtained using the family of hash functions detailed in chapter 6.

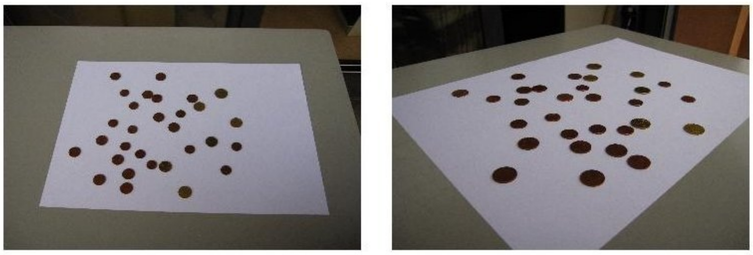


Figure 5.4: *"Coin Tag". Left: Reference image. Right: Query image.*

5.6 Conclusion

We have presented a method for identifying configurations of circles in images taken from different viewpoints. The proposed signature is invariant under perspective and has a vectorial form, suitable for an *LSH* search approach. We use the fact that the convex hull of a planar set of points is (quasi-)invariant under perspective and that it is possible to obtain a double invariant for a pair of circles using the generalised eigenvalues of the matrices of their projections. Considering the pairs of circles that are neighbours in the convex layers and computing these invariant values for each pair, we can obtain an invariant-under-perspective signature by concatenating these values. Our approach reduces the complexity of the identification process and moreover, it provides a solution to handle the rotation problem between two images. The tests carried out on computer-generated images and on real images show the robustness against noise and the performance in computational time. Nonetheless, the stability of the convex layers remains an open problem.

Chapter 6

Target Application: Bubble Tag™ Identification

The work presented in this thesis targets the automatic Bubble Tag™ identification, within a “1 to many” authentication protocol, in wine authentication purposes. The Bubble Tag™ used in our work (figure 6) has a rectangular form, of about 1 $cm \times 1$ $cm \times 1$ mm, and may contain about $80 - 100$ bubbles. Additionally, a linear marker is inlaid “by hand” after the Bubble Tag™ formation, in order to ease the handling of similarity transformations (rotation, translation, scaling). A more complete description of the Bubble Tag™ is given in Appendix 1.

Proposed Pipeline for Bubble Tag™ “1 to Many” Retrieval

A scheme of the complete process of a “1 to many” wine bottle authentication based on the Bubble Tag™ is given in figure 6.1.

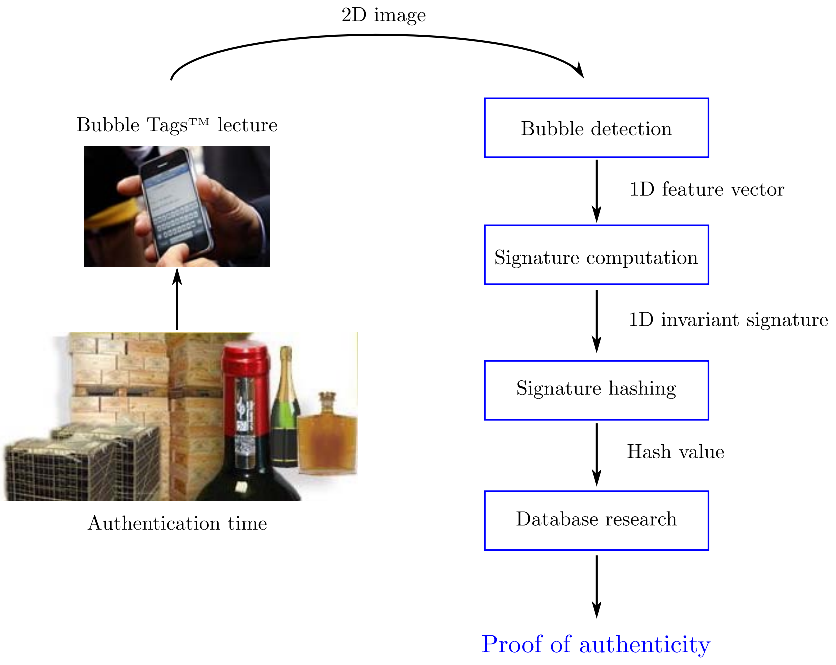


Figure 6.1: *Operation chain in wine bottle authentication, using the Bubble Tag™.*

The process starts with the Bubble Tag™ lecture, using an ordinary camera, i.e. a non-calibrated camera, whose position is non-rigid with respect to the Bubble Tag™. At the lecture time, the system has to verify that the tag to be identified is valid, i.e. it is a 3D bubble configuration, and not a copy/photography of a Bubble Tag™. To this end, we have designed a simple controlled coloured lighting system which induces coloured sheen on the edges of the bubbles. The simple presence of the sheen in an image taken under these lighting conditions confirms that the tag is a valid 3D Bubble Tag™. Additionally, locating the sheen in the image could enable one to extract the positions of the bubbles, even in low-resolution images, by using simple operations of mathematical morphology. The lack of some formal results prevents us from describing in detail this work here.

Once in possession of a (gray-level) Bubble Tag™ image, the system carries out the feature extraction phase, i.e. bubbles (ellipses) detection and linear marker (line segments) detection using the algorithm proposed in chapter 4, which conveniently takes in charge the simultaneous detection of line segments and ellipses. The ellipse parameters are used subsequently as raw information for a signature extraction procedure, which yields an invariant-under-perspective signature (chapter 5). Note, however, that with the proposed signature, one can dispense with the linear marker, as the designed signature handles intrinsically all the quasi-affine distortions. Finally, in order to check the authenticity of the given object, the system queries a database to retrieve, if it exists, the database entry corresponding to the obtained query signature. For efficiency reasons, the search in the database should be performed using some locality hashing techniques, described in Appendix 2.

Appendix 1: Bubble Tag™ Description

"*The Bubble Tag™ is made of a translucent polymer inside which, as a result of a random phenomenon, self-generated bubbles appear. This constellation of bubbles forms the basis of the identity given to a product or document. Each Bubble Tag™ is unique and impossible to replicate, even by Prooftag.*" (Prooftag web site www.prooftag.com).

According to [Prooftag 2006], a crude approximation for the probability of obtaining two identical Bubble Tags™ can be computed by considering the probability of observing two identical Bubble Tag™ images[1], i.e. the images of two bubble configurations containing the same number of bubbles, having the same positions and the same sizes. For a start, let us consider the toy example where one wants to compute the probability of observing two identical bubble configurations, each configuration containing one bubble, in images of $m \times n$ pixels. We assume that the bubbles have circular form in images. Formally, the position of each bubble can be represented by a random variable having a uniform distribution. The two bubbles are in the same position if their centres are at a distance of less than 1 pixel. Thus the probability that the two bubbles have the same position is $p_{pos} = \frac{1}{mn}$. Equally, we consider the radius as a Gaussian random variable, of mean μ and variance σ^2, thus following a normal distribution $N(\mu, \sigma^2)$. Two bubbles have the same radius if the difference between the two radii is less than 1 pixel. After

[1] We are talking here about images taken with a fixed camera, having the optical axis orthogonal to the plane containing the Bubble Tag™, thus the bubble configurations do not undergo significant geometric distortions (scaling, rotation, projection) between successive shootings.

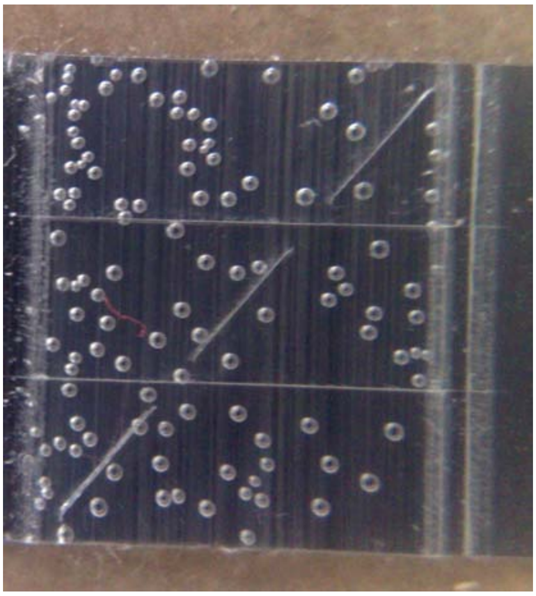


Figure 6.2: *Bubble Tag™ image.*

some computation, we find that the radius difference is again a random variable following a normal centred distribution, of variance $2\sigma^2$. Thus the probability of having a radius difference less than 1 is $p_{rad} = \int_0^1 \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{8\sigma^2}}$. The random variables representing the position and the radius difference being independent, the probability of having identical bubbles is:

$$p_{identic\_1} = p_{pos} p_{rad} = \frac{1}{mn} \int_0^1 \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{8\sigma^2}}.$$

For the general case, when the bubble configurations consist of b independent bubbles, the probability that the two configurations are identical is:

$$p_{identic\_b} = b! \left(\frac{1}{mn} \int_0^1 \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{t^2}{8\sigma^2}} \right)^b.$$

As a numerical example, if we consider bubble configurations of 40 bubbles, with radii having a standard deviation $\sigma = 2$, in images of 400 × 300 pixels:

$$p_{identic\_40} = 3.6 \times 10^{-184}.$$

The weak probability of obtaining two identical Bubble Tags™ allows qualifying them as *unique*, thus suitable for use in authentication purposes.

Appendix 2: "1 to Many" Bubble Tag™ Authentication Protocol Using *LSH* Functions

The "1 to many" Bubble Tag™ authentication protocol – given a query signature of a Bubble Tag™, we need to find its corresponding reference signature in a database – can reveal itself a highly resource consuming process, in both query time and memory usage, as the goal is not to find an exact match, but the nearest match (neighbour). This is a common problem in various domains: information retrieval, pattern recognition, image and video databases. The solution would be to find the nearest neighbour in some metric space, by comparing the query signature with each signature in the database. Problems occur when the database is large and the objects to be compared are complex, which is the case in the Bubble Tag™ problem: the database may contain thousands of entries, each one obtained by concatenating the invariant values (d, r), as described in Chapter 5, having thus about 2×100 elements (for a Bubble Tag™ containing 100 bubbles). The query processing time grows linearly with the number of items in the database and the complexity of the objects.

Problem Statement

In the following, we will consider a Bubble Tag™ signature, extracted using the technique presented in Chapter 5, as a *point* belonging to some metric space $\mathbb{X}$.

The nearest neighbour problem can be stated as an optimization problem [Andoni 2006]: the goal is to find in a database the point $\mathbf{p}$ which minimises an objective function, namely the distance to the query point $\mathbf{q}$.

Due to the aforementioned issues concerning the query time, the problem should be defined as: given a set (database) $\mathbf{P}$ of n points $\mathbf{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$ in some metric space $\mathbb{X}$, preprocess $\mathbf{P}$ as to efficiently answer queries which require finding the point in $\mathbf{P}$ closest to a query point $\mathbf{q} \in \mathbb{X}$ [Indyk 1998]. Existing classic methods to accomplish this task rely on trees and hashes.

Trees and Hashes

A very popular data structure used when carrying out searching tasks in multidimensional spaces is the *k-d tree* [Bentley 1975]. By building a tree, the search operation begins by interrogating the top node to see in which branch the node that corresponds to the query should be looked for. The process is repeated recursively until the desired result is achieved. The problem with this type of multidimensional algorithms is that they do not provide a real solution when the dimensionality is large enough, as the required memory space and query time grow exponentially with the dimension. Thus they bring no (or insignificant) improvement comparing to a linear algorithm, when the dimensionality of the search space is greater than a few dimensions. This phenomenon is known as "the curse of dimensionality" [Clarkson 1994].

A second technique relies on building hash tables [Califano 1993]. By using hash functions, large (possibly variable-sized) amount of data is converted into a small datum, usually a single integer that serves as an index into the hash table. A *collision* appears when two points hash to the same value. The well designed hash functions ensure that symbols which are close together fall into different buckets. This makes a hash table a good means for finding exact matches.

To efficiently solve the nearest neighbour problem, the *locality sensitive hashing* techniques prove to be more appropriate [Indyk 1998, Andoni 2006, Slaney 2008].

Locality Sensitive Hashing (*LSH*)

The *LSH* approach is based on the idea of using hash functions that provide a much higher probability of collision for objects that are close to each other than for those that are far apart.

Given a set $\mathbf{P}$ of points in a d-dimensional space $\mathbb{R}^d$, let $\mathcal{H}$ be a family of hash functions h mapping $\mathbb{R}^d$ to a subspace $\mathbb{U}$.

Definition 1. The family $\mathcal{H}$ is called *locality sensitive* if it satisfies the following conditions [Slaney 2008]:

(a) for any points $\mathbf{p}$ and $\mathbf{q}$ from $\mathbb{R}^d$ that are close to each other, there is a high probability that they fall into the same bucket, i.e. $\mathbb{P}[h(\mathbf{p}) = h(\mathbf{q})] \geq P_1$ for $\|\mathbf{p} - \mathbf{q}\| \leq r_1$.

(b) for any points $\mathbf{p}$ and $\mathbf{q}$ in $\mathbb{R}^d$ that are far apart, there is a low probability that they fall into the same bucket, i.e. $\mathbb{P}[h(\mathbf{p}) = h(\mathbf{q})] \leq P_2$ for $\|\mathbf{p} - \mathbf{q}\| \geq cr_1 = r_2$.

Here, $\|\cdot\|$ denotes the L_2 vector norm, and r_1, r_2, c are some real values, with $c > 1$.

In order for an *LSH* family to be useful, it has to satisfy $P_1 > P_2$ (probabilities of collision). If **(a)** and **(b)** stand, $\mathcal{H}$ is called (r_1, cr_1, P_1, P_2)-sensitive.

Equivalently, the *LSH* family can be defined as:

Definition 2. A family $\mathcal{H} = \{h : \mathbb{R}^d \rightarrow \mathbb{U}\}$ is called *locality-sensitive* if, for any $\mathbf{q}$, the function $p(t) = \mathbb{P}_{\mathcal{H}}[h(\mathbf{q}) = h(\mathbf{p}) : \|\mathbf{q} - \mathbf{p}\| = t]$ is strictly decreasing in t. That is, the probability of collision of points $\mathbf{q}$ and $\mathbf{p}$ is decreasing with the distance between them [Andoni 2006].

We will briefly describe the approach used by Andoni et al. in *E2LSH* [Andoni 2005]. We thank Alexandr Andoni for providing the *E2LSH* library, which allowed us to test their proposed family of hash functions in our problem.

The *E2LSH* Algorithm [Andoni 2006]

Informally, the *LSH* functions can be seen as *projection* operations that map data points from a high-dimensional space to a low-dimensional subspace. First, the neighbours of the query point are identified. Afterwards, the points are subject to a number of random projections of different directions, each time the nearby points being tracked. The points that appear close to each other in multiple projections are kept, as they are probably the true neighbours. To illustrate this idea, let us consider the simple case of some points on a sphere, which undergoes a number of projections 3D →2D [Slaney 2008]. Two points that are close on the sphere, will remain close in all projections, but in some particular cases, other (intrusive) points will equally appear as close, namely the points that are diametrically opposed on the sphere, when the projection is done in a direction parallel with the diameter relying the point of interest and the intrusive point.

The projection used in *E2LSH* [Andoni 2006] is the dot product (scalar projection), given by $h(\mathbf{q}) = \mathbf{x} \cdot \mathbf{q}$, where $\mathbf{q}$ is a query point in a high-dimensional space, and $\mathbf{x}$ is a vector with components that are selected independently from a Gaussian distribution. The choice of the family of hashing functions and of $\mathbf{x}$ is justified by the properties of the *stable distributions*.

Definition 3. A distribution $\mathcal{D}$ over $\mathbb{R}$ is called α-*stable* if there exists $\alpha \geq 0$, such that for any n real numbers $p_1, \ldots, p_n$ and i.i.d. random variables $X_1, \ldots, X_n$, with distribution $\mathcal{D}$, the random variable $\sum_i p_i X_i$ has the same distribution as the variable $(\sum_i |p_i|^\alpha)^{1/\alpha} X$, where X is a random variable with distribution $\mathcal{D}$ [Zolotarev 1986].

In particular, a Gaussian (normal) distribution $\mathcal{D}_G$, defined by the density function $g(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$, is a 2-*stable* distribution [Zolotarev 1986].

The intuition behind the hash functions is as follows. The dot product $\mathbf{x} \cdot \mathbf{p}$ projects each vector to the real line. From the definition of α-stable distributions (definition 3), it follows that for two vectors $(\mathbf{p}_1, \mathbf{p}_2)$, the distance between their projections $(\mathbf{x} \cdot \mathbf{p}_1 - \mathbf{x} \cdot \mathbf{p}_2)$ is distributed as $\|\mathbf{p}_1 - \mathbf{p}_2\|_\alpha X$, where X has an α-stable distribution. If one "chops" the real line into equi-width segments of appropriate size w and assigns hash values to vectors based on which segment they project onto, it is intuitively clear that this function will be locality-preserving in the sense described above.

Formally, the hash function will be given by $h_{\mathbf{x},b}(\mathbf{p}) : \mathbb{R}^d \rightarrow \mathbb{Z}$:

$$h_{\mathbf{x},b}(\mathbf{p}) = \lfloor \frac{\mathbf{x} \cdot \mathbf{p} + b}{w} \rfloor, \tag{6.1}$$

where $\lfloor \cdot \rfloor$ is the floor operation, $\mathbf{x}$ is a d dimensional vector with elements chosen independently from an α-stable distribution, and b is a real number chosen uniformly from the range $[0, w]$. b makes the quantisation ("chopping") error easier to analyse, with no loss in performance.

One can compute the probability of collision for two vectors $\mathbf{p}_1, \mathbf{p}_2$, under the hash function given in (6.1). Let $f_\alpha(t)$ denote the density function of the absolute value of the α-stable distribution. If we consider $c = \|\mathbf{p}_1 - \mathbf{p}_2\|_\alpha$ and $\mathbf{x}$ a vector whose entries are drawn from an α-stable distribution, it results that $\mathbf{x} \cdot \mathbf{p}_1 - \mathbf{x} \cdot \mathbf{p}_2$ is distributed as cX, where X is drawn from an α-stable distribution. Thus:

$$p(c) = \mathbb{P}[h_{\mathbf{x},b}(\mathbf{p}_1) = h_{\mathbf{x},b}(\mathbf{p}_2)] = \int_0^w \frac{1}{c} f_\alpha\left(\frac{t}{c}\right)\left(1 - \frac{t}{w}\right) dt.$$

For a fixed w, the probability of collision $p(c)$ decreases monotonically with $c = \|\mathbf{p}_1 - \mathbf{p}_2\|_\alpha$, satisfying the definition of the locality sensitive function (definition 2).

In order to obtain a satisfactory gap between P_1 and P_2, introduced in definition 1, several functions like (6.1) are concatenated, the resulting family $\mathcal{G}$ containing l functions, each one with k subfunctions h: $g_j(\mathbf{q}) = [h_{1,j}(\mathbf{q}), \ldots, h_{k,j}(\mathbf{q})]$, where $h_{t,j}(1 \leq t \leq k, 1 \leq j \leq l)$ are chosen independently and uniformly at random from $\mathcal{H}$. In practice this means that k dot products are performed in parallel and the k inner products are quantised such that similar points will fall in the same buckets in all dimensions.

This process of projection and quantisation places each data point $\mathbf{p}$ of $\mathbf{P}$ in a hash bucket. When a query $\mathbf{q}$ is processed, the buckets where $\mathbf{q}$ is hashed are scanned and the stored points are retrieved. In order to efficiently retrieve the stored points, a conventional (exact) hashing method is used. Finally, after retrieving the candidate points, the distances to the query point are computed and the nearest point to the query point is reported.

We have applied the *E2LSH* algorithm in our work with its default parameters, on computer-generated images of bubble configurations, and on "coin tags", as detailed in section 5.5. The obtained results were very encouraging. For databases of ~ 300 entries, the mean query time was about $10^{-4}s$.

This study on the techniques used in problems that involve research in high-dimensional spaces, together with the recommended use of an *LSH* approach for Bubble Tag™ identification, was published in [Pătrăucean 2010a].

Bibliography

[Abdi 2007] H. Abdi. Bonferroni and sidak corrections for multiple comparisons. Sage, 2007. (Cited on pages 50, 127 and 141.)

[Ahn 1997] S. J. Ahn and M. Schultes. *A New Circular Coded Target for Automation of Photogrammetric 3D-Surface Measurements.* In Proceedings of the 4th Conference on Optical 3-D Measurement Techniques, 1997. (Cited on page 19.)

[Ahn 2001] S. J. Ahn, W. Rauh and H.-J. Warnecke. *Least-Squares Orthogonal Distances Fitting of Circle, Sphere, Ellipse, Hyperbola, and Parabola.* Pattern Recogn., pages 2283–2303, 2001. (Cited on pages 21, 22, 24, 28, 123 and 137.)

[Akaike 1973] H. Akaike. Information theory and an extension of the maximum likelihood principle, volume 1, pages 267–281. Akademiai Kiado, 1973. (Cited on pages 73, 129 and 143.)

[Alizadeh 2000] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown and L. M. Staudt. *Distinct Types of Diffuse Large B-Cell Lymphoma Identified by Gene Expression Profiling.* Nature, vol. 403, pages 503–511, 2000. (Cited on page 50.)

[Anderson 1954] R. L. Anderson. *The Problem of Autocorrelation in Regression Analysis.* J. Am. Stat. Assoc., vol. 49, pages 113–129, 1954. (Cited on page 78.)

[Andoni 2005] A. Andoni and P. Indik. *E2LSH 0.1 User Manual.* online, 2005. (Cited on page 115.)

[Andoni 2006] A. Andoni and P. Indik. *Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions.* In Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, pages 459–468, 2006. (Cited on pages 4, 99, 114, 115, 116, 133 and 147.)

[Babu 2004] M. M. Babu. Computational genomics: Theory and application, chapitre An Introduction to Microarray Data Analysis, pages 105–124. Horizon scientific press, 2004. (Cited on pages 50 and 51.)

[Ballard 1987] D. H. Ballard. *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms.* chapitre Generalizing the Hough Transform to Detect Arbitrary Shapes, pages 714–725. 1987. (Cited on pages 21 and 40.)

[Barber 1996] C. B. Barber, D. P. Dobkin and H. Huhdanpaa. *The Quickhull Algorithm for Convex Hulls.* ACM Trans. Math. Softw., vol. 22, pages 469–483, 1996. (Cited on page 107.)

[Barinova 2010] O. Barinova, V. Lempitsky and P. Kohli. *On the Detection of Multiple Object Instances Using Hough Transforms.* In Proceedings of the 23th IEEE Conference on Computer Vision and Pattern Recognition, 2010. (Cited on pages 82, 127 and 140.)

[Benjamini 1995] Y. Benjamini and Y. Hochberg. *Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing.* J. Roy. Stat. Soc. B Met., vol. 57, no. 1, pages 289–300, 1995. (Cited on page 52.)

[Bentley 1975] J. L. Bentley. *Multidimensional Binary Search Trees Used for Associative Searching.* Commun. ACM, vol. 18, pages 509–517, 1975. (Cited on page 114.)

[Bolles 1981] R. C. Bolles and M. A. Fischler. *A RANSAC-Based Approach to Model Fitting and its Application to Finding Cylinders in Range Data.* In Proceedings of the 7th International Joint Conference on Artificial Intelligence, pages 637–643, 1981. (Cited on page 27.)

[Bookstein 1979] F. L. Bookstein. *Fitting Conic Sections to Scattered Data.* Comput. Vision Graph., vol. 9, no. 1, pages 56–71, 1979. (Cited on page 27.)

[Burns 1986] J. B. Burns, A. R. Hanson and E. M. Riseman. *Extracting Straight Lines.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 8, pages 425–455, 1986. (Cited on pages 20 and 57.)

[Burrus 2009] N. Burrus, T. M. Bernard and J. M. Jolion. *Image Segmentation by A Contrario Simulation.* Pattern Recogn., vol. 42, pages 1520–1532, 2009. (Cited on page 38.)

[Califano 1993] A. Califano and I. Rigoutsos. *FLASH: A Fast Look-Up Algorithm for String Homology.* In Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology, pages 56–64, 1993. (Cited on page 114.)

[Calvet 2011] L. Calvet, P. Gurdjos, V. Charvillat, S. Gasparini and P. Sturm. *Suivi de caméra à partir de marqueurs plans composés de cercles concentriques : paradigme et algorithmes.* In Proceedings of the 13th Congrès des jeunes chercheurs en vision par ordinateur (ORASIS), 2011. (Cited on pages 4 and 19.)

[Canny 1986] J. Canny. *A Computational Approach to Edge Detection.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 8, pages 679–698, 1986. (Cited on pages 20, 39, 40, 59, 126 and 140.)

[Cao 2007] F. Cao, J. Delon, A. Desolneux, P. Musé and F. Sur. *A Unified Framework for Detecting Groups and Application to Shape Recognition.* J. Math. Imaging Vis., vol. 27, pages 91–119, 2007. (Cited on page 70.)

[Cao 2008] F. Cao, J. L. Lisani, J.-M. Morel, P. Musé and F. Sur. A theory of shape identification, volume 1948 of *Lecture Notes in Mathematics.* Springer, 2008. (Cited on page 47.)

[Chen 1993] Z. Chen, C.-M. Wang and S.-Y. Ho. *An Effective Search Approach to Camera Parameter Estimation Using an Arbitrary Planar Calibration Object.* Pattern Recogn., vol. 26, pages 655–666, 1993. (Cited on page 98.)

[Chen 2008] Y. Chen, H. Ip, Z. Huang and G. Wang. *Full Camera Calibration from a Single View of Planar Scene*. In Proceedings of the 4th International Symposium on Advances in Visual Computing, pages 815–824, 2008. (Cited on page 98.)

[Chernov 1984] N. I. Chernov and G. A. Ososkov. *Effective Algorithms for Circle Fitting*. Comput. Phys. Commun., vol. 33, pages 329–333, 1984. (Cited on page 19.)

[Chernov 2005] N. Chernov and C. Lesort. *Least Squares Fitting of Circles*. J. Math. Imaging Vis., vol. 23, pages 239–252, 2005. (Cited on pages 22, 24, 25, 27, 39, 124 and 138.)

[Chia 2011] A. Chia, S. Rahardja, D. Rajan and M. K. Leung. *A Split and Merge Based Ellipse Detector with Self-Correcting Capability*. IEEE T. Image Process., vol. 20, pages 1991–2006, 2011. (Cited on page 42.)

[Clarkson 1994] K. L. Clarkson. *An Algorithm for Approximate Closest-Point Queries*. In Proceedings of the 10th Annual Symposium on Computational Geometry, pages 160–164, 1994. (Cited on page 114.)

[De Marsico 2010] M. De Marsico, M. Nappi and R. Daniel. *Iris Segmentation for Identification Systems*. In Proceedings of the 20th International Conference on Pattern Recognition, pages 2857–2860, 2010. (Cited on page 19.)

[Desolneux 2000] A. Desolneux, L. Moisan and J.-M. Morel. *Meaningful Alignments*. Int. J. Comput. Vision, vol. 40, pages 7–23, 2000. (Cited on pages 38, 43, 44, 52, 54, 56, 69, 78, 127, 140 and 141.)

[Desolneux 2002] A. Desolneux, S. Ladjal, L. Moisan and J.-M. Morel. *Dequantizing Image Orientation*. IEEE T. Image Process., vol. 11, pages 1129–1140, 2002. (Cited on page 58.)

[Desolneux 2007] A. Desolneux, L. Moisan and J.-M. Morel. From gestalt theory to image analysis: A probabilistic approach, 1st edition. Springer Publishing Company, Incorporated, 2007. (Cited on pages 38, 43, 45, 46, 47, 53, 54, 58, 66, 70, 127, 140 and 141.)

[Duda 1972] R. O. Duda and P. E. Hart. *Use of the Hough Transformation to Detect Lines and Curves in Pictures*. Commun. ACM, vol. 15, pages 11–15, 1972. (Cited on pages 21, 40, 46, 56, 127 and 140.)

[Duda 2001] R. O. Duda, P. E. Hart and D. G. Stork. Pattern classification, 2nd edition. John Wiley & Sons, 2001. (Cited on pages 70, 129 and 143.)

[Dudoit 2003] S. Dudoit, J. P. Shaffer and J. C. Boldrick. *Multiple Hypothesis Testing in Microarray Experiments*. Stat. Sci., vol. 18, pages 71–103, 2003. (Cited on page 52.)

[Elliman 2002] D. Elliman. *TIF2VEC, An Algorithm for Arc Segmentation in Engineering Drawings*. In Selected Papers from the 4th International Workshop on Graphics Recognition Algorithms and Applications, pages 350–358, 2002. (Cited on page 42.)

[Etemadi 1992] A. Etemadi. *Robust segmentation of edge data*. In Proceedings of the 1992 International Conference on Image Processing and its Applications, pages 311–314, 1992. (Cited on pages 42, 64 and 82.)

[Fari 2004] H. Fari and E. P. Simoncelli. *Differentiation of Discrete Multidimensional Signals.* IEEE Trans. on Imag. Proc., vol. 13, pages 496–508, 2004. (Cited on page 20.)

[Fitzgibbon 1999] A. W. Fitzgibbon, M. Pilu and R. B. Fisher. *Direct Least-Squares Fitting of Ellipses.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 21, pages 476–480, 1999. (Cited on pages 21, 28, 31, 39, 124 and 137.)

[Forstner 1987] W. Forstner and E. Gulch. *A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular Features.* In Proceedings of the 1987 Intercommission Conference on Fast Processing of Photogrammetric Data, pages 281–305, 1987. (Cited on pages 29 and 63.)

[Frahm 2006] J.-M. Frahm and M. Pollefeys. *Ransac for (Quasi-)Degenerate Data (qdegsac).* In Proceedings of the 19th IEEE Conference on Computer Vision and Pattern Recognition, pages 453–460, 2006. (Cited on page 22.)

[Gabriel 1959] K. R. Gabriel. *The Distribution of the Number of Successes in a Sequence of Dependent Trials.* Biometrika, vol. 46, pages 454–460, 1959. (Cited on page 78.)

[Gander 1994] W. Gander, G. H. Golub and R. Strebel. *Least-Squares Fitting of Circles and Ellipses.* BIT Numerical Math., vol. 43, pages 558–578, 1994. (Cited on pages 21, 22, 24, 25, 27, 123, 124, 137 and 138.)

[Golub 1996] G. H. Golub and C. F. Van Loan. Matrix computations, 3rd edition. Johns Hopkins University Press, 1996. (Cited on pages 100, 101, 131, 144 and 145.)

[Golub 1999] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri and C. D. Bloomfield. *Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring.* Science, vol. 286, pages 531–537, 1999. (Cited on page 50.)

[Gordon 2007] A. Gordon, G. Glazko, X. Qiu and A. Yakovlev. *Control of the Mean Number of False Discoveries, Bonferroni and Stability of Multiple Testing.* Ann. Appl. Stat., vol. 1, pages 179–190, 2007. (Cited on pages 52, 127 and 141.)

[Gringorten 1966] I. I. Gringorten. *A Stochastic Model of the Frequency and Duration of Weather Events.* J. Appl. Meteorol., vol. 5, pages 606–624, 1966. (Cited on page 78.)

[Grompone von Gioi 2007] R. Grompone von Gioi, J. Jakubowicz and G. Randall. *Multisegment Detection.* In Proceedings of the 14th International Conference on Image Processing, volume 2, pages 253–256, 2007. (Cited on page 70.)

[Grompone von Gioi 2008] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel and G. Randall. *On Straight Line Segment Detection.* J. Math. Imaging Vis., vol. 32, pages 313–347, 2008. (Cited on pages 45, 47, 53, 61 and 78.)

[Grompone von Gioi 2009] R. Grompone von Gioi and J. Jakubowicz. *On Computational Gestalt Detection Thresholds.* J. Physiology-Paris, vol. 103, pages 4–17, 2009. (Cited on pages 52, 55, 74, 129 and 143.)

[Grompone von Gioi 2010] R. Grompone von Gioi, J. Jakubowicz, J.-M. Morel and G. Randall. *LSD: A Fast Line Segment Detector with a False Detection Control*. IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, pages 722–732, 2010. (Cited on pages 20, 38, 39, 57, 58, 59, 69, 86, 128 and 142.)

[Guennebaud 2007] G. Guennebaud and M. Gross. *Algebraic Point Set Surfaces*. ACM Trans. Graph., vol. 26, 2007. (Cited on pages 19, 24, 25, 29, 33, 124 and 138.)

[Gurdjos 2006] P. Gurdjos, P. Sturm and Y. Wu. *Euclidean Structure from $N \geq 2$ Parallel Circles: Theory and Algorithms*. In Proceedings of the 9th European Conference on Computer Vision, volume 1, pages 238–252, 2006. (Cited on pages 19, 98, 100 and 103.)

[Gurdjos 2009] P. Gurdjos, V. Charvillat, G. Morin and J. Guenard. *Multiple View Reconstruction of a Quadric of Revolution from Its Occluding Contours*. In Proceedings of the 9th Asian Conference on Computer Vision, pages 1–12, 2009. (Cited on page 19.)

[Haralick 1992] R. M. Haralick and L. G. Shapiro. Computer and robot vision, 1st edition. Addison-Wesley Longman Publishing Co., Inc., 1992. (Cited on page 28.)

[Hartley 1994] R. Hartley. *An Algorithm for Self-Calibration from Several Views*. In Proceedings of the 9th IEEE Conference on Computer Vision and Pattern Recognition, page 908, 1994. (Cited on page 98.)

[Hartley 2004] R. Hartley and A. Zisserman. Multiple view geometry in computer vision, 2nd edition. Cambridge University Press, 2004. (Cited on pages 9, 13, 22, 28, 31, 32, 99, 100, 106, 126, 130, 132, 140, 144 and 146.)

[Hastie 2001] T. Hastie, R. Tibshirani and J. Friedman. The elements of statistical learning. data mining, inference, and prediction. Springer-Verlag, 2001. (Cited on pages 70, 71, 73, 129 and 143.)

[Heikkila 1998] J. Heikkila. *Moment and Curvature Preserving Technique for Accurate Ellipse Boundary Detection*. In Proceedings of the 14th International Conference on Pattern Recognition, pages 734–, 1998. (Cited on page 29.)

[Helgert 1970] H. J. Helgert. *On Sums of Random Variables Defined on a Two-State Markov Chain*. J. Appl. Probab., vol. 7, pages 761–765, 1970. (Cited on page 78.)

[Igual 2007] L. Igual, J. Preciozzi, L. Garrido, A. Almansa, V. Caselles and B. Rougé. *Automatic Low Baseline Stereo in Urban Areas*. Inverse Probl. and Imaging, vol. 1, pages 319–348, 2007. (Cited on page 74.)

[Indyk 1998] P. Indyk and R. Motwani. *Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality*. In Proceedings of the 30th Annual ACM Symposium on Theory of Computing, pages 604–613, 1998. (Cited on pages 4, 114, 115, 133 and 147.)

[Johnson 2003] M. Johnson and K. Ramchandran. *Dither-Based Secure Image Hashing Using Distributed Coding*. In Proceedings of the 10th International Conference on Image Processing, pages 751–754, 2003. (Cited on page 4.)

[Kanatani 2004] K. Kanatani and N. Ohta. *Automatic Detection of Circular Objects by Ellipse Growing.* vol. 4, pages 35–50, 2004. (Cited on pages 19, 42 and 60.)

[Kanatani 2008] K. Kanatani and Y. Sugaya. *Compact Algorithm for Strictly ML Ellipse Fitting.* In Proceedings of the 19th International Conference on Pattern Recognition, pages 1–4, 2008. (Cited on pages 22, 27 and 39.)

[Kanatani 2010] K. Kanatani and Y. Sugaya. *Unified Computation of Strict Maximum Likelihood for Geometric Fitting.* J. Math. Imaging Vis., vol. 38, pages 1–13, 2010. (Cited on page 27.)

[Kasa 1962] I. Kasa. *A Curve Fitting Procedure and its Error Analysis.* IEEE Trans. Inst. Meas., vol. 25, pages 8–14, 1962. (Cited on pages 21, 24, 27, 123 and 137.)

[Kim 2002] E. Kim, M. Haseyama and H. Kitajima. *Fast and Robust Ellipse Extraction from Complicated Images.* In Proceedings of the 2002 International Conference on Information Technology and Applications, 2002. (Cited on pages 42 and 60.)

[Kiryati 1991] N. Kiryati, Y. Eldar and A.M. Bruckstein. *A Probabilistic Hough Transform.* Pattern Recogn., vol. 24, pages 303–316, 1991. (Cited on pages 41, 127 and 140.)

[Kullback 1951] S. Kullback and R. A. Leibler. *On Information and Sufficiency.* Ann. Math. Stat., vol. 22, pages 49–86, 1951. (Cited on page 73.)

[Ladd 1975] D. W. Ladd. *An Algorithm for the Binomial Distribution with Dependent Trials.* J. Am. Stat. Assoc., vol. 70, pages 333–340, 1975. (Cited on pages 78 and 79.)

[Leavers 1993] V. F. Leavers. *Which Hough Transform?* CVGIP: Image Underst., vol. 58, pages 250–264, 1993. (Cited on page 21.)

[Lehmann 2005] E. L. Lehmann and J. P. Romano. Testing statistical hypotheses, 3rd edition. Springer Texts in Statistics, 2005. (Cited on page 48.)

[Li 1986] H. Li, M. A. Lavin and R. J. Le Master. *Fast Hough Transform: A Hierarchical Approach.* Comput. Vision Graph. Image Process., vol. 36, pages 139–161, 1986. (Cited on page 41.)

[Lowe 1985] D. G. Lowe. Perceptual organization and visual recognition. Kluwer Academic Publishers, 1985. (Cited on pages 44 and 73.)

[Lowe 1987] D. G. Lowe. *Three-Dimensional Object Recognition from Single Two-Dimensional Images.* Artif. Intell., vol. 31, pages 355–395, 1987. (Cited on pages 4 and 42.)

[Mai 2008] F. Mai, Y. S. Hung, H. Zhong and W. F. Sze. *A Hierarchical Approach for Fast and Robust Ellipse Extraction.* Pattern Recogn., vol. 41, pages 2512–2524, 2008. (Cited on page 42.)

[Mairal 2010] J. Mairal. *Sparse Coding for Machine Learning, Image Processing and Computer Vision.* PhD thesis, Ecole Normale Superieure de Cachan, France, 2010. (Cited on page 4.)

[Marr 1979] D. Marr and E. Hildreth. *Theory of Edge Detection*. In Proceedings of the Royal Society London, 1979. (Cited on page 20.)

[Matas 1998] J. Matas, C. Galambos and J. Kittler. *Progressive Probabilistic Hough Transform*. In Proceedings of the 1998 British Machine Vision Conference, 1998. (Cited on pages 41, 127 and 140.)

[McLaughlin 1998] R. A. McLaughlin. *Randomized Hough Transform: Improved Ellipse Detection with Comparison*. Pattern Recogn. Lett., vol. 19, 1998. (Cited on pages 40, 127 and 140.)

[Metzger 1975] W. Metzger. Gesetze des sehens. Waldemar Kramer, 1975. (Cited on pages 38, 127 and 141.)

[Moisan 2004] L. Moisan and B. Stival. *A Probabilistic Criterion to Detect Rigid Point Matches between Two Images and Estimate the Fundamental Matrix*. Int. J. Comput. Vision, vol. 57, pages 201–218, 2004. (Cited on pages 22, 38 and 69.)

[Morin 1993] L. Morin. *Quelques contributions des invariants projectifs à la vision par ordinateur*. PhD thesis, Institut National Polytechnique de Grenoble, France, 1993. (Cited on page 9.)

[Mundy 1992] J. L. Mundy and A. Zisserman. Geometric invariance in computer vision. MIT Press, 1992. (Cited on pages 9, 101, 131 and 145.)

[Nguyen 2009] T. M. Nguyen, S. Ahuja and Q. M. J. Wu. *A Real-Time Ellipse Detection Based on Edge Grouping*. In Proceedings of the 2009 International Conference on Systems, Man and Cybernetics, pages 3280–3286, 2009. (Cited on pages 42 and 60.)

[O'Gorman 1976] F. O'Gorman and M. B. Clowes. *Finding Picture Edges through Collinearity of Feature Points*. IEEE Trans. Comput., vol. 25, pages 449–456, 1976. (Cited on page 41.)

[Ouellet 2008] J. N. Ouellet and P. Hebert. *Precise Ellipse Estimation without Contour Point Extraction*. Mach. Vision Appl., vol. 21, pages 59–67, 2008. (Cited on pages 28 and 29.)

[Paton 1970] K. Paton. *Conic Sections in Chromosome Analysis*. Pattern Recogn., vol. 2, pages 39–51, 1970. (Cited on pages 19 and 27.)

[Pătrăucean 2010a] V. Pătrăucean, P. Gurdjos and J. Conter. *"Bubble Tag"-Based System for Object Authentication*. In Proceedings of the 8th International Conference on Communications, pages 133–136, 2010. (Cited on page 117.)

[Pătrăucean 2010b] V. Pătrăucean, P. Gurdjos and J. Conter. *Bubble Tag Identification Using an Invariant-under-Perspective Signature*. In Proceedings of the 20th International Conference on Pattern Recognition, pages 408–411, 2010. (Cited on page 98.)

[Pătrăucean 2011] V. Pătrăucean, P. Gurdjos, G. Morin and J. Conter. *Détection de primitives linéaires et circulaires par une approche a contrario*. In Proceedings of the 13th Congrès des jeunes chercheurs en vision par ordinateur (ORASIS), 2011. (Cited on pages 20 and 38.)

[Porrill 1990] J. Porrill. *Fitting Ellipses and Predicting Confidence Envelopes Using a Bias Corrected Kalman Filter*. Image Vision Comput., vol. 8, pages 37–41, 1990. (Cited on page 28.)

[Pratt 1987] V. Pratt. *Direct Least-Squares Fitting of Algebraic Surfaces*. SIGGRAPH Comput. Graph., vol. 21, pages 145–152, 1987. (Cited on pages 21, 24, 25, 27, 124, 137 and 138.)

[Prooftag 2006] Prooftag. *Code à bulles, Technical Report*. 2006. http://www.prooftag.fr. (Cited on page 112.)

[Rabin 2009] J. Rabin. *Approches Robustes pour la Comparaison d'Images et la Reconnaissance d'Objets*. PhD thesis, Télécom ParisTech, 2009. (Cited on page 38.)

[Rice 2010] K. Rice. *A Decision-Theoretic Formulation of Fisher's Approach to Testing*. The Am. Stat., vol. 64, pages 345–349, 2010. (Cited on page 53.)

[Rosin 1996] P. L. Rosin. *Assessing Error of Fit Functions for Ellipses*. Graph. Models Image Process., vol. 58, pages 494–502, 1996. (Cited on pages 24 and 27.)

[Rosin 1998] P. L. Rosin. *Ellipse Fitting Using Orthogonal Hyperbolae and Stirling's Oval*. Graph. Models Image Process., vol. 60, pages 209–213, 1998. (Cited on pages 14, 24 and 64.)

[Rust 1995] R. T. Rust, D. Simester, R. J. Brodie and V. Nilikant. *Model Selection Criteria: An Investigation of Relative Accuracy, Posterior Probabilities, and Combinations of Criteria*. Manage. Sci., vol. 41, pages 322–333, 1995. (Cited on page 73.)

[Safaee-Rad 1991] R. Safaee-Rad, I. Tchoukanov, B. Benhabib and K. C. Smith. *Accurate Parameter Estimation of Quadratic Curves from Grey-Level Images*. CVGIP: Image Underst., vol. 54, pages 259–274, 1991. (Cited on page 24.)

[Sampson 1982] P. D. Sampson. *Fitting Conic Sections to "Very Scattered" Data: An Iterative Refinement of the Bookstein Algorithm*. Comput. Vision Graph., vol. 18, no. 1, pages 97–108, 1982. (Cited on pages 21, 27, 39, 124 and 137.)

[Semple 1952] J. G. Semple and G. T. Kneebone. Algebraic projective geometry. Oxford University Press, 1952. (Cited on pages 9, 26, 125 and 139.)

[Shafiq 2001] M. S. Shafiq, T. S. Tümer and H. C. Güler. *Marker Detection and Trajectory Generation Algorithms for a Multicamera Based Gait Analysis System*. Mechatronics, vol. 11, pages 409–437, 2001. (Cited on page 21.)

[Slaney 2008] M. Slaney and M. Casey. *Locality-Sensitive Hashing for Finding Nearest Neighbors*. IEEE Signal Proc. Mag., vol. 25, pages 128–131, 2008. (Cited on page 115.)

[Späth 1997] H. Späth. *Orthogonal Least Squares Fitting by Conic Sections*. In Proceedings of the 2nd International Workshop on Recent Advances in Total Least Squares Techniques and Errors-in-Variables Modeling, pages 259–264, 1997. (Cited on page 24.)

[Stewart 1995] C. V. Stewart. *MINPRAN: A New Robust Estimator for Computer Vision.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 17, pages 925–938, 1995. (Cited on page 53.)

[Sturm 2007] P. Sturm and P. Gargallo. *Conic Fitting Using the Geometric Distance.* In Proceedings of the 8th Asian Conference on Computer Vision, volume 2, pages 784–795, 2007. (Cited on pages 21, 22, 24, 123 and 137.)

[Takahashi 2010] T. Takahashi and M. Kudo. *Margin Preserved Approximate Convex Hulls for Classification.* In Proceedings of the 20th International Conference on Pattern Recognition, pages 4052–4055, 2010. (Cited on page 107.)

[Taubin 1991] G. Taubin. *Estimation of Planar Curves, Surfaces, and Nonplanar Space Curves Defined by Implicit Equations with Applications to Edge and Range Image Segmentation.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 13, pages 1115–1138, 1991. (Cited on pages 21, 27, 33, 124 and 137.)

[Urs 1972] R. Urs. *An Iterative Procedure for the Polygonal Approximation of Plane Curves.* Comput. Vision Graph., vol. 1, pages 244–256, 1972. (Cited on page 42.)

[Walsh 1951] J. E. Walsh. *A Large Sample t-Statistic which is Insensitive to Non-Randomness.* J. Am. Stat. Assoc., vol. 46, pages 79–88, 1951. (Cited on page 78.)

[West 1992] G. A. West and P. L. Rosin. *Multi-Stage Combined Ellipse and Line Detection.* In Proceedings of the 1992 British Machine Vision Conference, pages 197–206, 1992. (Cited on pages 42, 64 and 65.)

[Wu 2006] Y. Wu, X. Li, F. Wu and Z. Hu. *Coplanar Circles, Quasi-Affine Invariance and Calibration.* Image Vision Comput., vol. 24, pages 319–326, 2006. (Cited on pages 19 and 98.)

[Xu 1990] L. Xu, E. Oja and P. Kultanen. *A New Curve Detection Method: Randomized Hough Transform (RHT).* Pattern Recogn. Lett., vol. 11, pages 331–338, 1990. (Cited on pages 32, 40, 127 and 140.)

[Yang 2003a] Y. Yang. *Can the Strengths of AIC and BIC Be Shared?* Biometrika, vol. 92, page 2003, 2003. (Cited on page 73.)

[Yang 2003b] Y. H. Yang and T. Speed. Statistical analysis of gene expression microarray data, chapitre Design and Analysis of Comparative Microarray Experiments, pages 35–93. Chapman and Hall/CRC, 2003. (Cited on pages 50 and 52.)

[Zhang 2000] Z. Zhang. *A Flexible New Technique for Camera Calibration.* IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, pages 1330–1334, 2000. (Cited on page 98.)

[Zhang 2005] S.-C. Zhang and Z.-Q. Liu. *A Robust, Real-Time Ellipse Detector.* Pattern Recogn., vol. 38, pages 273–287, 2005. (Cited on pages 19 and 60.)

[Zolotarev 1986] V. M. Zolotarev. One-dimensional stable distributions. American Mathematical Society, 1986. (Cited on page 116.)