Early Cognitive Vision: Feedback Mechanisms for the Disambiguation of Early Visual Representation

Dissertation

zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades

"Doktor rerum naturalium" der Georg-August-Universität Göttingen

vorgelegt von

Nicolas Pugeault

aus

Strasbourg

Göttingen 2008

Referentin/Referent:	Prof. Florentin Wörgötter
Koreferentin/Koreferent:	Prof. Norbert Krüger
Tag der mündlichen Prüfung:	15. Januar 2008

Abstract

Recent years have seen considerable progress in low level image processing as well as model based, vision applications. Joining the two fields proves to be a difficult problem due to the local ambiguity and noisiness of visual processes, and to requirements of robustness and accuracy of high level processes. This thesis presents an Early Cognitive Vision framework that aims at providing a rich and reliable scene representation from visual information. This framework preserves conflicting hypothesis in its early stages, and makes use of feedback mechanisms between different visual processes and layers of representation to achieve disambiguation. In a first part, symbolic local image descriptors are extracted from the responses of early vision filters, and perceptual grouping constraints are applied to the resulting image representation. A second part discusses the use of stereopsis to reconstruct an equivalent 3D representation of the visual information, and the interactions between perceptual grouping, stereopsis and 3D reconstruction processes. The third part of this thesis integrates visual information across time to further disambiguate the representation. This framework has been used successfully in several contexts, that are discussed in the conclusion of this thesis.

Acknowledgements

First and foremost, I would like to thank Professor Norbert Krüger. His enthousiasm, curiosity and vision had a major influence in my scientific development, and I believe that our long discussions improved this work to a great extent. Second, this thesis owe a great deal to the continuing support, encouragements, and precious advice of my supervisor, Professor Florentin Wörgötter.

I would also like to thank my colleagues from the universities of Stirling, Odense and Goettingen, Matthias Hennig, Bernd Porr, Ausra Saudargiene, Tomas Kulvicius, Renaud Detry, Emre Baseski, Dirk Kraft, Yan Shi, Lars Baunegaard and Anders Kjaer-Nielsen for friendly and stimulating discussions.

These long years of work would have been a lonely time indeed without my friend Sinan Kalkan. Only his invaluable help allowed me to submit and defend the present thesis in Goettingen.

None of this would have been possible without the love and encouragements of my family. I thank them for giving me the greatest gifts of all: the opportunity and desire to learn.

Last but not least, I want to thank Marina for her patience and love, that enlightens my days.

Contents

Chapter 1

Introduction

The principal person in a picture is light.

- Manet

Interpreting visual information is a seemingly simple task: humans and animals extract relevant information from a scene in a nearly instantaneous manner, giving us the illusion of simplicity. This apparent simplicity is but a lure, how such a feat is accomplished is still obscure to modern science.

Computer vision research has faced this difficulty since its earlier stages. Although it is possible to capture and reproduce images with extreme likeness, the process of actually *interpreting* these images is mostly unknown. Images captured by a camera are encoded as arrays of pixels encoding local colour (or only intensity) information, *i.e.*, the response of three photoreceptors with different spectral sensitivities (classically red–green–blue). When combined, these allow for the description and reproduction of colour according to human perception. When considering the problem of *interpreting* those images, the problem arises that this pixel information is only remotely related to physical properties of the environment.

First, the value of each pixel in the image is a function of the light reflected by objects' surfaces and captured by those photoreceptors. The light reflected by surface can be modelled (under some simplifying assumptions) as a function of the ambient light, and the surface's orientation and *reflectance function*. This reflectance function is a property of the material from which the reflecting surface is made. One common fundamental assumption is that all viewed surfaces are *Lambertian*, *i.e.*, perfectly matte. However, this assumption holds at best partially because the reflectance function of most surfaces in the

natural world feature a mix of matte and reflective components (*e.g.*, think of a white wall). Moreover, a pixel's colour may vary with the illumination and ultimately with the spectral sensitivity of the photoreceptor; hence the same surface produces very different image information under different viewing conditions. Nonetheless, this assumption is fundamental for vision as it allows us to infer surfaces properties from the reflected light encoded by a pixel. For example, a black pixel is an indication of a dark surface in the scene.

Second, the area of a surface that reflects light onto a given pixel is determined by the properties of the optics focusing such reflected light onto the photoreceptors. For example, if the lens is characterised by a small focal length (or if the reflecting surface is far away), it follows that each photoreceptor (and therefore each pixel in the image) will sample light reflected by an area larger than for a larger focal length (or for a closer surface). Because the colour information captured by a photoreceptor is the sum of the light focused on its sensitive area by the optics, a pixel does not strictly describes an ideal point in space, but a whole area.

The physical processes of image formation therefore lead to the following problems, from the perspective of visual perception:

Loss of depth information: During a camera image acquisition process the light reflected by 3D surfaces onto the camera's photoreceptors is encoded on a planar grid: the image. In this process, the depth of the reflecting surfaces is lost. Recovering this depth is a critical step of visual perception, and a difficult task. So–called depth cues allow us to recover the pixel's depth information, and can be roughly categorised in two classes: 1) pictorial depth cues, that require only one image (*e.g.*, depth from defocus, depth from shading), and 2) multiple views cues (*e.g.*, stereo, depth from motion) that require several views of the same scene from different perspectives. The first class of cues are more difficult to model and develop only after a few months in infants. To the author's knowledge, no algorithm exists for reliably processing these cues in a general scenario.

The mathematics that underly multiple views depth cues are well known (see, *e.g.*, (Faugeras, 1993; Hartley and Zisserman, 2000)), and have been applied with some success in a variety of contexts. These are also the ones considered in this thesis (chapters 4 and 7).

Ambiguity and noise in the local signal: Because the pixel information they carry is so remotely connected to the scene's intrinsic qualities, local image patches can be very ambiguous — illustrated

in Fig. 1.1. In Fig. 1.1(a) the complex 3D structure of the object is difficult to infer from the local image information. The shape of the object should, as a result of occlusion, create a Y junction. In Fig. 1.1(b), the contour in the inside of the basket is locally invisible, due to the shadow cast by the basket's handle. Finally, Fig. 1.1(c) and Fig. 1.1(d) are locally difficult to distinguish (the so–called aperture problem). This is problematic for matching object's points across different views (*e.g.*, for stereo or pose estimation).

One surface can generate different signals: The same surface can generate very different image data under different viewing conditions, due to different perspective transformation, pixel sampling, illumination, and reflective properties of non–Lambertian surfaces. This is critical for applications that require matching object locations viewed from different perspectives — *e.g.*, stereopsis.

These difficulties (ambiguity, noisiness) are characteristics of the *inverse problems* (Tarantola, 2005), wherein vision belongs: a problem is inverse if it involves evaluating parameters of a model from sampled data. This thesis presents a framework for early vision to circumvent these difficulties making use of the ubiquitous redundancy in visual information to draw corrective feedback mechanisms between visual processes, and thereby extract a reliable scene description.

Section 1.1 presents a brief overview of some relevant works in the computer vision literature. Because of the massive amount of published studies, this account does is not an attempt to comprehensiveness, but rather to present chosen pieces from the vision literature that bear similarity with the work presented herein. Section 1.2 will then expose two simplifying approaches that are used to tackle the vision problem and position the present work relatively to these. Finally section 1.3 outlines the framework presented in this thesis, and discuss the structure of the present document.

1.1 Previous works

This fundamental ambiguity in visual information makes the interpretation of visual signal an extremely difficult task, and lead the first Artificial Intelligence attempts to computer vision to dead–ends — see (Marr, 1982).

In his seminal book, Marr (1982) established a new paradigm of visual perception as a modular hierarchy of progressively more abstracted representations: the primal sketch, the $2\frac{1}{2}$D sketch, and the

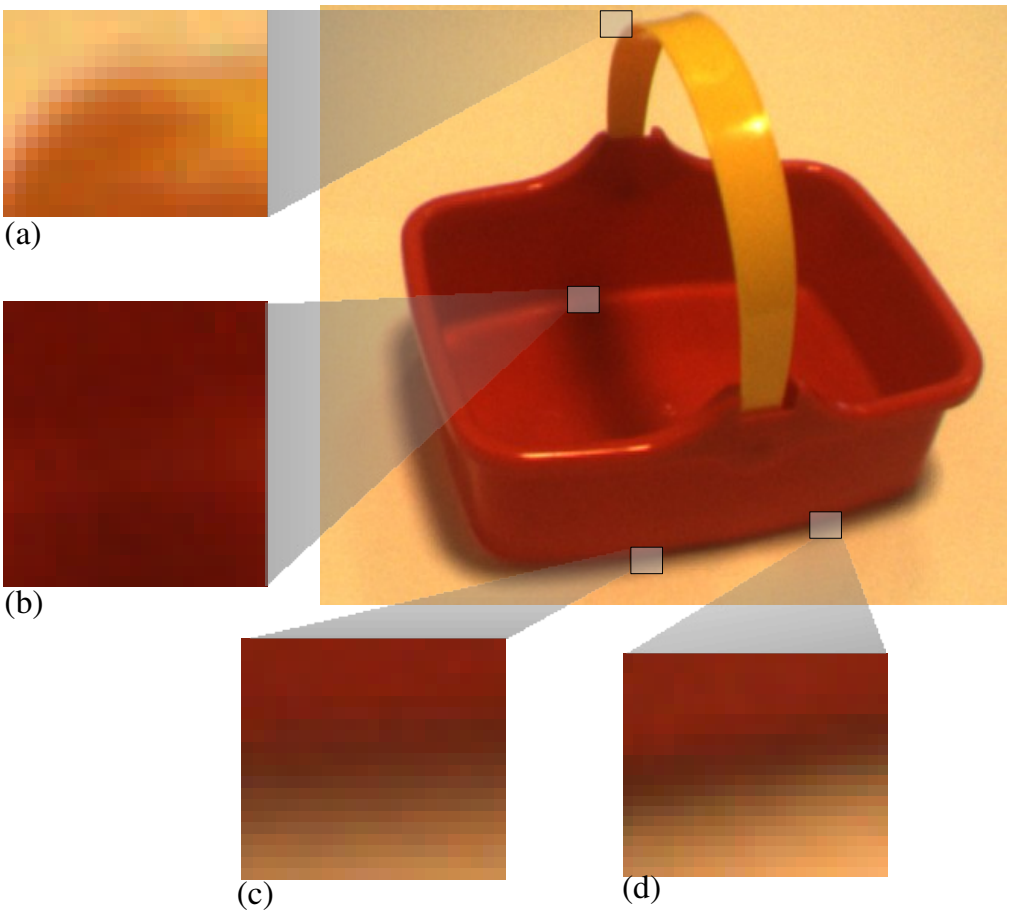


Figure 1.1: Illustration of the local ambiguity of images. It is difficult to infer the complex structure of the object in (a); in (b), the contour of the object is locally hidden by the shadow; (c) and (d) are difficult to distinguish locally.

3D model. The primal sketch form a first interpretation of the image signal in terms of local features (zero–crossings, blobs, terminations, *etc.*). This concept is akin to the representation we will discuss in chapter 2. The $2\frac{1}{2}$D sketch integrates the depth information (*e.g.*, from stereopsis) to describe the scene in terms of surfaces, depth discontinuities and surface orientation discontinuities. This is not yet a full 3D representation because it is viewpoint dependent, and therefore occluded part of the scene are not represented. This is comparable to the scene representation presented in chapter 4. Finally, the 3D model of the scene is a hierarchy of object centred 3D models that describe the scene's objects and their relations. This representation is similar to the accumulated representation presented in chapter 7.

As a methodology to investigate these representations and the processes that connect them, David Marr advocated for the distinction between different levels of analysis: 1) Computational theory, where the fundamental relation between the scene's intrinsic quality and some image observable property is rigorously investigated; 2) Algorithms and data structures; and 3) Implementation of the algorithms. Therefore, insights on the implementation level can be gained from neurophysiological results, that provide information about the response of cortical cells; insights on the algorithmic level can be gained from psychophysical results — by depriving a mechanism from critical information or placing it in pathological conditions. Marr argued that vision research should be based on a viable computational theory. This paradigm favours a research method where the overall vision problem is fragmented in independent sub–modules, for which the computational theory is tractable. Aloimonos and Shulman (1989) discussed that a large amount of modern vision research studies one such modules, in the form:

compute Y from X

where X is a cue (stereopsis, texture, *etc.*) and Y is an intrinsic property of the scene (shape, depth, *etc.*).

This paradigm has led to a thourough understanding of several critical aspects of vision like stereopsis (Scharstein and Szeliski, 2002), optic flow (Barron et al., 1994), edge detection (Ziou and Tabbone, 1998), and feature extraction (Mykolajczyk and Schmid, 2004; Mikolajczyk and Schmid, 2005), *etc.* Despite great progress in the theoretical understanding of these visual modalities and significant improvements in the algorithms that compute them, local ambiguities in the visual signal, and in local operators that apply on it, proved irreducible. This lead Aloimonos and Shulman (1989) to argue that most of the modules in the Marr paradigm are indeed trying to address ill–posed problems. Accordingly, they proposed the study of inter–modules integration. The approach presented herein goes further in this

direction by describing feedback loops between early vision processes.

Wörgötter et al. (2004) discussed that the intricate complexity of the vision task and its generality renders it intractable by a purely data–driven, or knowledge–based process:

Data–driven (feed–forward) approaches do not provide the semantic understanding of the scene that is required for complex interaction. It is clear that numerous visual tasks require some amount of prior knowledge to be achieved. For example, in order to recognise an object the system requires prior knowledge of this object's shape, for driving on a road the system requires some knowledge of road markings, traffic signs, *etc.*

Knowledge–driven (top–down) approaches require the designer's knowledge of the domain to be built in the system. This exogenous knowledge is bound to be inadequate because the system's data structures, sensory signals, and reasoning are vastly different from the designer's. Moreover, because the designer cannot foresee all contingencies, the systems encounters the so–called *frame problem*: Concisely put, how an autonomous system is to decide which information is *relevant* for a specific task (*e.g.*, Denett (1984)). This is an unsolved problem in Artificial Intelligence, and therefore severely limits the generality and robustness of knowledge–driven vision systems.

From these remarks, we draw the following conclusions: 1) for a system to operate efficiently, a certain amount of domain knowledge is required to efficiently interpret and use visual information; 2) this domain knowledge needs to be formulated in the system's frame of reference and as a result cannot be provided by the designer; and therefore 3) there is the need for a common representation of visual information that can mediate both the learning of the domain's properties when the system is in an infant state, and the efficient implementation of this knowledge to interpret and react to a visual stimulus, when the system is operating.

Wörgötter et al. (2004) advocated a hierarchy of representations, where feedback mechanisms within a representation and between earlier and higher representations, lead to the self–emergence of complex features. The present work was developed in this context. Krüger and Wörgötter (2004) discussed that although local visual information is ambiguous and noisy, it is dominated by *regularities* that advocate for the understanding of vision as a *process of recurrent predictions*.

1.2 Simplification of the problem

Because of the complexity of the vision problem discussed above, it is necessary to simplify the problem in some way. Horn (1986) discussed two possible simplifications of the vision problem:

Simplify the *domain of application*: The first approach can provide working systems for a limited number of tasks in a well defined scenario, but offer little insight into the workings of human vision.

Focus on a *specific module*: This second approach, that has been prominent since Marr's work, has led to a better understanding of several of vision's sub–tasks — like optic flow, motion estimation, and stereopsis. Nonetheless, this research has generally reached a hard limit on performance due to theoretical limitations in the problem's formulation.

In this work, we chose to investigate the importance of inter–modules feedbacks for the generation, and the disambiguation of a general–purpose representation of visual information. Because we want the proposed framework to be generic, we will not make any restricting assumption about the domain wherein the system operates. Moreover, although the focus of this work is rather general, note that it has already served as a vision front–end in different contexts: grasping (Aarno et al., 2007), object shape learning (see chapter 7 and (Pugeault et al., 2007a)), ego–motion estimation (see chapter 6 and (Pugeault et al., 2006a)). This work was developed in the course of the European project ECOVISION (2003), and is now used in the context of the two projects PACO-PLUS (2006) and DrivSco (2006).

There exists a large amount of evidence that the human visual system in its first cortical stages processes a number of aspects of visual data (see, *e.g.*, (Hubel and Wiesel, 1962; Oram and Perrett, 1994)). These aspects, in the following called visual modalities, cover, *e.g.*, local orientation (Hubel and Wiesel, 1962, 1969), colour (Hubel and Wiesel, 1969), junction structures (Shevelev et al., 1995), stereo (Barlow et al., 1967) and optic flow (Hubel and Wiesel, 1969). At the first stage of visual processing (called 'Early Vision' by Krüger et al. (pted)), these modalities are computed locally for a certain retinal position. At a later stage (called 'Early Cognitive Vision' by Krüger et al. (pted)), results of local processing become integrated with the spatial and temporal context. Computer vision has dealt to a large extent with these modalities separately and in many computer vision systems, one or more of the above–mentioned aspects are processed (see, *e.g.*, Marr (1982); Schiele and Crowley (1996); Lades et al. (1993)).

Krüger and Wörgötter (2004) described two main regularities in visual data (well recognised in the computer vision community) that support such a disambiguation process: (i) coherent motion of rigid bodies; and (ii) statistical interdependencies underlying most grouping processes (Elder and Goldberg, 2002; Geisler et al., 2001; Krüger, 1998a). These two regularities allow predictions between locally extracted visual events, and verification of the spatio–temporal coherence of transient perceptual hypotheses.

The establishment of such a disambiguation process presupposes communication of temporal and spatial information, requiring the local representation of visual data to comply with the two properties:

Property 1.2.1. Predictability *The local representation of visual data allows for rich predictions between related visual events — e.g., the change of position and appearance of a local patch under a rigid body motion.*

and

Property 1.2.2. Condensation *The local representation of visual data reduces the dimensionality of the local signal allowing the process to work with limited bandwidth.*

König and Krüger (2006) argued that properties 1.2.1 and 1.2.2 naturally result in symbolic representations.

1.3 Framework outline

The mechanisms we will consider herein are the following: image feature extraction, perceptual grouping, 3D shape reconstruction, ego–motion estimation and temporal integration of transient visual information. Although we acknowledge that there are other problems of interest, we believe that this is an adequate set of problems for this study. Fig. 1.2 gives a schematic outline of the framework elaborated throughout this thesis. In this figure, full arrows show feed–forward communication between processes, dashed arrows show inter–process feedback mechanisms.

Part I focuses on the extraction of early symbolic description from images. First, in chapter 2 we will present the extraction of image features that form a suitable basis for the subsequent processes. The image representation $\boldsymbol{I}$ used here was first discussed by Krüger et al. (2004), and a complete technical description is under submission (Krüger et al., 2007). It extracts from an image local, multi–modal

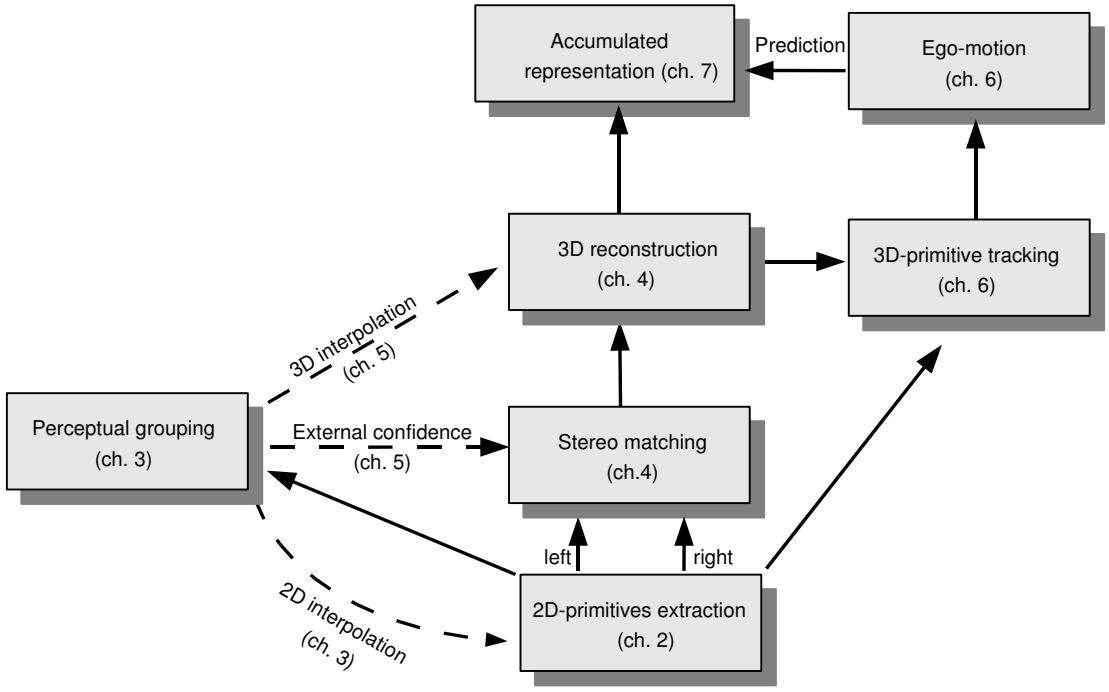


Figure 1.2: Presentation of the framework discussed in this thesis. The dashed lines represent feedback mechanisms, the solid ones bottom–up processes.

contour descriptors called 2D–primitives. These 2D–primitives are then used in chapter 3 for defining perceptual grouping mechanisms that extract image contours' properties. A first inter–process feedback mechanism is discussed, where extracted contours are used to correct 2D–primitives' properties. We show that accuracy can thereby be improved and noise reduced.

Part II departs from the retinotopic image information to represent visual information in space. Chapter 4 recovers depth information using stereopsis between 2D–primitives extracted from a stereo pair of images. This leads to the reconstruction of 3D–primitives that provide a symbolic $2\frac{1}{2}$D representation $\mathcal{S}$ of the scene's shape. Chapter 5 discusses feedback mechanisms between perceptual grouping and stereo reconstruction processes, that lead to a significant improvement in performance, reliability, and accuracy of the resulting $2\frac{1}{2}$D representation.

All the mechanisms presented up to this point are transient: they apply to an image, or an image pair, at an instant in time, and are continuously applied as new images are captured by the cameras. This raises several problems: first, some memory mechanism is required to provide a consistent, stable representation of the world; second, different $2\frac{1}{2}$D representations need to be integrated to provide a full 3D representation of the scene's shape; third, ego– and object motion need to be estimated with accuracy for the system to interact successfully with its environment (*e.g.*, navigation, obstacle detection).

Accordingly, Part III of this thesis will break with this paradigm, and discusses how this transient spatial information can be temporally integrated. Chapter 6 proposes an algorithm to estimate the ego–motion from the 3D–primitives extracted between two instants. Chapter 7 makes use of this motion knowledge to integrate transient scene representations into a stable *accumulated representation* $\mathcal{A}$. This can provide a full 3D representation of objects, if the system has viewed it from enough different perspectives.

Part I

Presentation of the image representation $\mathcal{I}$

Chapter 2

Extraction of the primitives

The human doesn't see things as they are, but as he is.

- Racter

In this chapter, we describe a novel representation of visual information, based on local symbolic descriptors called *visual primitives*. This representation was presented in (Krüger et al., 2007). A primitive combines different visual modalities into one local feature descriptor, and thus, allows for a condensed representation of the visual scene (satisfying property 1.2.2). Furthermore, primitives allow to formulate predictions (property 1.2.1) using statistical dependencies from grouping and motion. These statistical dependencies bootstrap a disambiguation process that is described by, *e.g.*, Pugeault et al. (2006b).

For all the reasons discussed in this thesis' introduction, an array of pixels is a representation inadapted to the task of image understanding. Furthermore, Hubel and Wiesel (1969) showed that cells concerned with early vision processing were acting as contrast detectors. Since, numerous techniques have been proposed to compute a more meaningful, stable, and invariant representation of the visual information. One side of the problem lies with defining local operators to transform raw intensity into more significant information. The other side of the problem is the robust extraction of features. The line between the two is often blurred, as feature extraction generally requires a pre–processing of the signal, and filtering operations give meaningful results only at specific locations. In the following, we will give a short overview of image processing techniques, and present the features that we will be using in this work.

2.1 Literature review of feature descriptors

Considerable work has been conduced over the last 50 years to find robust and meaningful image descriptors. A good image descriptor should be reliably extracted from pixel information and provide information that is semantically relevant for image interpretation, *i.e.*, that describes image structure. In their recent review of the different feature extraction techniques, Mikolajczyk and Schmid (2005) separate the process between *interest point detectors* and *local descriptor*. Interest points detectors aim at selecting a subset of locations in the image that contain salient structures, and were reviewed by C. Schmid and R. Mohr and C. Baukhage (2000), Feature descriptors aim at providing an efficient description of the local structure in an image; they were surveyed by Mykolajczyk and Schmid (2004). The local descriptor should be chosen relatively to the kind of interest point it is describing: *e.g.*, orientation adequately describes an edge, but would be inapt to describe a blob or a corner.

2.1.1 The concept of scale

The information contained in an image can be considered at different *scales*. Coarse scales, only describe the signal's major structures; fine scales also describe thin details. This idea is comforted by the fact that the receptive field of neurons in the early visual cortex of cats (Hubel and Wiesel, 1962) and primates (Hubel and Wiesel, 1968) spans several octaves.

Tony Lindeberg (1994) showed that such a scale space can be obtained from an image by convolving with the Gaussian kernel and its derivatives. The width of the Gaussian function defines the scale of the kernel. Lindeberg also studied the automatic scale selection for blob (Lindeberg, 1998b) and edge extraction (Lindeberg, 1998a).

In computer vision, it is widely accepted that operations tend to be more reliable in coarse scales (due to less redundancy in the signal) but less accurate (due to blurring); fine scales deliver less reliable, yet more accurate results. Therefore, it is commonplace to employ pyramidal coarse–to–fine processing to circumvent the ambiguity inherent local operations (*e.g.*, (Irani and Anandan, 2000; Pritchett and Zisserman, 1998)). Such processing start the matching from a coarse, less ambiguous scale, and iteratively refine it using progressively finer scales.

Although the present work only considers one scale at a time, it is worth noting that Felsberg et al. (2005) defined a Monogenic Scale Space that extends the properties of the monogenic signal, used in

this work, into scale–space. Alternatively, Lindeberg (1998a); Elder and Zucker (1998) proposed an automatic scale selection process for edge features. Therefore, the results herein could be extended into scale–space.

2.1.2 Interest point detector

Most of the locations in an image are locally homogeneous and therefore contain very little information. For instance, it is impossible to identify one specific location in a homogeneous area, and thus to find the corresponding location in another image. For this reason, such locations can be discarded early in the vision process, in order to focus on more informative areas (in a similar manner, the retinal ganglion cells, in the mammalian visual system, are only sensitive to contrast (Lennie, 2000, Fig. 29-11)). Furthermore, the mid–level vision framework introduced in the following chapters is based on high order relations between features in the image; the complexity of such operations increases quickly with the number of features. It is therefore desirable to discard locations where such algorithms produce a large overhead for little benefit, and conversely, to identify *interest points* where they can be processed successfully.

Definition 2.1.1. *An interest point detector is a process that selects a subset of locations in the image that are deemed adequate for further processes.*

In the following we will give a brief overview of interest point detectors that are commonplace in the vision literature. A prominent example of interest point detectors is the so–called Harris corner detector (Harris and Stephens, 1988). This operator is an isotropic version of the Moravec corner detector — based on a Gaussian smoothing of the local patch (hence effectively operating on a circular window, instead of a square one). The motivation for this operator is to consider the average change in the local patch, induced by a shift of the image in any direction. If we describe the change $E(x, y)$ produced by a shift (x, y) as a matrix:

$$E(x, y) = M(x, y)^T, \tag{2.1}$$

then the two eigenvalues α, β of M are computed. If both α and β are low, then the patch is unaltered and therefore homogeneous. If only one of α or β is low then the patch is altered along one single component, defining an edge. Finally, if both α and β are high then the image's structure is two dimensional and denotes a "corner" or "junction". Note that such "corners" can also be merely textured surfaces, rather

than proper three–dimensional corners. These locations have the advantage of being detected independently of the orientation of the patch, and are widely used in the literature (Baumberg, 2000; Torr and Zisserman, 2000; Zhang et al., 1995). Mikolajczyk and Schmid (2005) proposed a scale adapted version of the Harris detector, on a Gaussian scale–space, called *Harris–Laplace*. The scale is determined by the local maxima of the Laplacian–of–Gaussian (LoG), providing an additional scale invariance. An alternative approach is the *Hessian Laplace* used in (Mykolajczyk and Schmid, 2004; Lowe, 2004); points are localised in space as the local maxima of the Hessian determinant and in scale as the local maxima of the LoG. In contrast to the Harris–like detectors, this operator detects *blob–like* structures rather than corners. Mykolajczyk and Schmid (2004) proposed affine variations on these detectors, where localisation is obtained using either the Harris– or the Hessian–Laplace detectors. The affine neighbourhood is determined by an *affine adaptation* process (note that affine invariance is akin to viewpoint invariance).

Finally, edge detectors, like Canny's classical algorithm (Canny, 1986), Zero–crossings (Marr, 1982), or phase congruence (Kovesi, 1999) attempt to detect pixels in the image that correspond to object's contours. Kovesi (1999) detects edge pixels at the location of phase congruence over the Fourier components of the signal. Marr (1982) remarked that edges are characterised as Zero–crossings in the Laplacian of Gaussian (LoG): the Laplacian of a convolution of the image with a Gaussian Kernel. This can be approximated by a Difference of Gaussians (DoG), which is computationally inexpensive. Edges are seldom used in feature matching approaches because of the local ambiguity rising from the aperture problem: "Given one point along one contour, typically all other points along the same contour will be similar." This is inconvenient when the matching of features itself is the end product of the whole system. On the other hand, it has been argued that edges are critically important in image interpretation (Marr, 1982), contain all necessary information in images (Elder, 1999), and are the main locations where occlusion occurs (Ogale and Aloimonos, 2006). Moreover there is some evidence that the human visual system makes intensive usage of edge–like structures in its early stages (Hubel and Wiesel, 1969; Grimson, 1993). For these reasons, the present work makes use of an image representation based on features sparsely extracted on images' contours. Interest points are sampled sparsely along images' contours, using a threshold on the monogenic signal's magnitude, as is described in section 2.2.3. The advantage of the symbolic representation proposed herein is that it allows to use semantic knowledge about the kind of structure those interest points describe (*i.e.*, contours) to drive this sparse sampling, and the sub–pixel localisation of the interest points.

2.1.3 Feature descriptor

Assuming that a suitable set of interest points have been selected, numerous vision operations (stereopsis, tracking, *etc.*) require to match such interest points across different views. This requires: 1) a vector p that describes the point, and 2) a metric $d(p, p')$ between a pair of local descriptors p and p'.

Definition 2.1.2. *A feature descriptor is a vector that describes a local area of the image.*

An ideal feature descriptor has the following properties:

Property 2.1.1. Viewpoint invariance: *given p an interest point, we would like the corresponding point p' under another viewpoint to be such that $d(p, p') < \epsilon$ with ϵ a small quantity.*

and

Property 2.1.2. Distinctiveness: *for any two distinct, non–corresponding interest points p and p', we want $d(p, p') > \epsilon$ with ϵ a small quantity.*

The former is a fundamental problem because there is not enough information in a local image patch to design a viewpoint invariant descriptor in the general case — demonstrated by Burns et al. (1992). The latter is critical for several vision processes that require to address the matching problem, *e.g.*, stereopsis and motion estimation.

A fairly intuitive way to compare two image patches is to compute the *cross–correlation* between them. Furthermore, if the cross–correlation operator is normalised, such a comparison is illumination invariant. On the down side, it is sensitive to viewpoint, rotation, and scale changes, and suffers from its high dimensionality (effectively a vector of 100 values for a greyscale patch of 10x10 pixels). Nonetheless, cross–correlation of intensity patches centred at Harris corners is still a prominent feature in the computer vision literature.

Lowe (2004) proposed a scale invariant region detector, combined with a region descriptor based on the distribution of image gradient in this region. Location is quantised to a 4x4 grid and orientation into 8 bins, resulting in a descriptor of dimension 128, called SIFT. This descriptor is invariant to rotation and scaling, and robust to affine and viewpoint transformations Hence SIFT is a good choice for matching processes. GLOH is a variant of the SIFT descriptor proposed by Mikolajczyk and Schmid (2005); position is sampled in a log–polar grid with three bins in the radial direction and eight in angular direction,

resulting in 17 location bins (a single location bin lies in the centcentre Furthermore, the orientation is quaquantised 16 bins for a total of 272 bins. Then the 128 most significant components are selected using PCA. PCA–SIFT is another variant where the position is sampled over a 39x39 grid, resulting to a vector of dimension $3,042$, then reduced to 36 using PCA.

Kovesi (1999) proposed to describe edges as point of phase congruence across different Fourier components. Alternatively, the response of Gabor, or other wavelet filters are frequently used for texture classification.

Derivatives computed up to a certain order effectively approximate a point neighbourhood. The set of local derivatives (*local jet*) were investigated by Koenderink and van Doorn (1987). They proposed to group these by invariance. The zeroth order contains the luminance information; the first order differential the gradient. From the second order differential a measure of the *elongated–ness*, *blob–ness*, or *feature–ness* of the patch is derived (these three values sum up to one). The third order is interpreted as a measure of *curvature*, *splay* or *edge–ness* of an elongated blob. Finally, the fourth order gives a measure of the *curvature trend*.

Schaffalitzky and Zisserman (2002); Baumberg (2000) use the response of a complex filter as descriptors. van Gool et al. (1996) proposed to use moments of the local image patch that are affine and photometric invariants.

The visual primitives proposed by Krüger et al. (2004); Krüger et al. (2007), provide a rich semantic description of the image, while achieving data compression. In this work we will use these primitives, that we will describe briefly in the next section.

2.2 The visual primitives

The primitives describe the properties of an image patch centred at a specific location (or point of interest) in the image according to different operators. Each of these local operators contains different information about the local patch, called *modality* in the following. In this sense the primitives are described as *local* and *multi–modal* feature descriptors. Moreover, the primitives encode a *symbolic* description of the local signal: in this work we will focus on edge–primitives that attach a semantic meaning to the local image patch.

In section 2.2.1 we will present a signal processing operator called the *monogenic signal*, and that

provides the local expression of orientation, phase, and magnitude. Section 2.2.2 exposes how the notion of *intrinsic dimension* is computed from this filter's output. Then section 2.2.3 describes how interest points are selected and located. Section 2.2.4 explains how the different modalities are computed at these interest points. Then section 2.2.5 discuss the primitives' sub–pixel localisation accuracy.

2.2.1 Low–level image processing: the monogenic signal

The extraction of a primitive starts with a rotation invariant quadrature filter that performs a *split of identity* of the signal (Felsberg and Sommer, 2001): it decomposes the signal into local amplitude (see Fig. 2.1, top row), orientation (see Fig. 2.1, second row), and phase (see Fig. 2.1, third row) information.[1]

The local amplitude is an indicator of the likelihood for the presence of an image structure. Orientation encodes the geometric information of the local signal while phase can be used to differentiate between different image structures ignoring orientation differences.

Phase encodes the grey level transition of the local image patch across the edge (as defined by the orientation) in a compact way (as one parameter only). For example, a pixel positioned on a bright line on a dark background has a phase of 0 whereas a pixel positioned on a bright/dark edge has a phase of $-\pi/2$ (see Fig. 2.2a and, *e.g.*, (Felsberg and Sommer, 2001; Granlund and Knutsson, 1995; Kovesi, 1999)).

Possible phases form continuum between $[-\pi, \pi[$, and are 2π–periodic: a phase of $-\pi$ represents the same contrast transition as a phase of π. Orientation θ (taking values in the the interval $[0, \pi)$) and phase ω are topologically organised on a half torus — see Fig. 2.2(c). If we extend the concept of orientation to that of a direction (therefore taking values in $[-\pi, \pi)$, see also (Jähne, 1997)) then the topology of the direction/phase space becomes a complete torus — see Fig. 2.2(b). On a local level, the direction is not decidable (Granlund and Knutsson, 1995); therefore, we will use the half torus topology.

The topology defined above is crucial for the definition of suitable metrics for phase and orientation. For example, a black–white step edge ($\omega = \pi/2$) with orientation θ is proximate to a white–black step edge ($\omega = -\pi/2$) of orientation $\pi - \theta$, but distant to a black–white step edge of orientation $\pi - \theta$. However, a white line on a black background with an orientation θ ($\omega = 0$) should have only a small distance to a white line on a black background with an orientation $\pi - \theta$ but a large one to any black line on a white background. Therefore, the extremities of the half–torus are linked in a continuous manner as shown in

[1] Note that amplitude, orientation and phase can be analogously computed by Gabor wavelets or steerable filters and that our representation does not depend on the filter introduced in (Felsberg and Sommer, 2001). For a discussion of different approaches to define harmonic filters as well as their advantages and problems, we refer to (Sabatini et al., 2007).

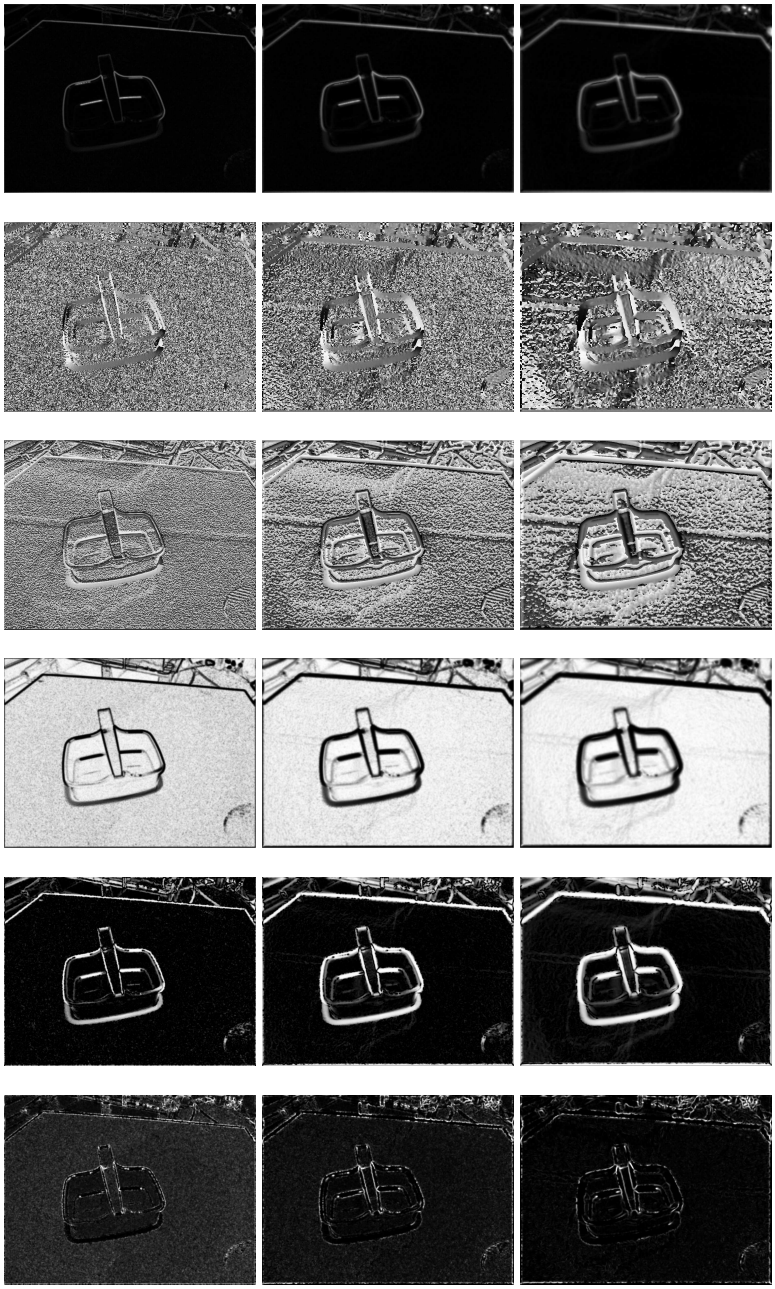
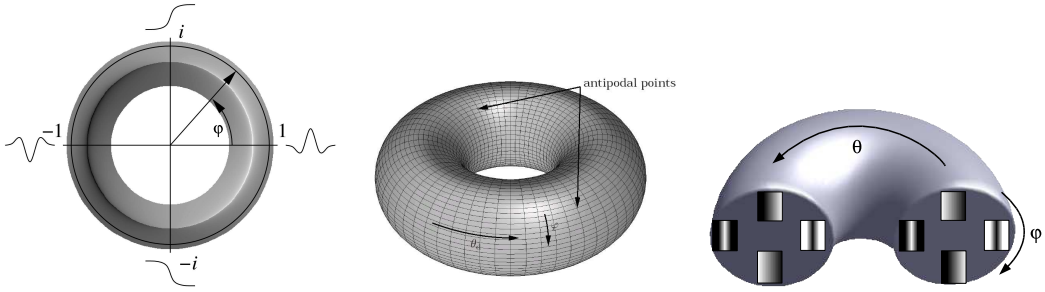


Figure 2.1: Illustration of the low–level processing for Primitive extraction. Each column shows the filter response for a different peak frequency: respectively 0.110 (left), 0.055 (middle) and 0.027 (right). Each row show a response maps for, from top to bottom, local amplitude, orientation, phase, intrinsically zero–dimensional (id0), one–dimensional (id1) and two–dimensional (id2) confidences. In all of those graphs white stands for a high response and black for a low one.



(a) phase (b) direction/phase space (c) orientation/phase space

Figure 2.2: a) Phase ω describes different intensity transitions, e.g., $\omega = \pi$ encodes a dark line on bright background, $\omega = -\pi/2$ encodes a bright/dark edge, $\omega = 0$ encodes a bright line on a dark background and $\omega = \pi/2$ encodes a dark/bright edge. The phase parameter embeds these distinct cases into a 2π-periodic continuum shown in (a). [Acknowledgement: Michael Felsberg] b) The torus topology of the orientation–phase space. The phase value ω is mapped on the cross section of the torus' tube whereas the orientation θ maps to the revolution angle the torus. c) When direction is neglected we get a half torus that is connected as indicated.

Fig. 2.2c. For a discussion of the orientation/phase metric, we refer to (Krüger and Felsberg, 2004).

We compute filter responses for three different scales, indicated hereafter by the peak frequency of the associated filter operations. [2] Fig. 2.1 shows the filter responses in terms of the local amplitude $m(\boldsymbol{x})$, orientation $\theta(\boldsymbol{x})$ and phase $\omega(\boldsymbol{x})$, alongside the resulting primitives, for three scales.

2.2.2 Intrinsic dimensionality

Different kinds of image structures coexist in natural images: homogeneous image patches, edges, corners, textures, *etc.*. Furthermore, certain concepts are only meaningful for specific classes of image structures. For example, the concept of orientation is well defined for edges or lines but not for junctions, homogeneous image patches or for most textures. In addition, the concept of position is different for a junction as compared to an edge or an homogeneous image patch — see Fig. 2.3. In homogeneous areas of the image no particular location can be defined (Fig. 2.3a), and therefore an equidistant sampling is appropriate. For a line or edge structure (Fig. 2.3b), position can be defined using energy maxima. However, because of the aperture problem, the energy maximum will span a one–dimensional manifold, and therefore the feature can be localised only up to this manifold. This results in a fundamental ambiguity in the localisation of edge/line local features. By contrast, the locus of a junction can be unambiguously

[2] Note that step edges have high amplitudes across scales, whilst line structures are represented as a line at coarse scales, and as two step–edges at fine scales, (see section 2.2.3 and (Lindeberg, 1998a)).

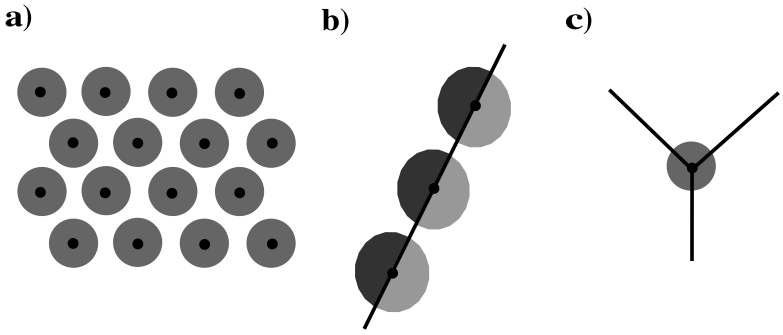


Figure 2.3: Different localisation problems faced by different classes of image structures: a) homogeneous area; b) edge or line; and c) junction (see text).

defined by the point of line intersection (see Fig. 2.3c). Similar considerations are required for other modalities such as colour, optic flow and stereo (see below).

Therefore, in order to design a symbolic descriptor that aptly describes all sort of local image patches, we need to semantically partition those patches according to their junction–ness, edge–ness or homogeneous–ness. This is formalised by the notion of *intrinsic dimension* (see, *e.g.*, (Zetzsche and Barth, 1990; Felsberg, 2002)).

Intrinsic dimension defines three classes of local image structures, illustrated in Fig. 2.3:

Zero–dimensional (id0): A local image patch is defined as *intrinsically zero–dimensional*, or *id0*, if it contains no structure. This is the case for homogeneous surfaces.

One–dimensional (id1): A local image patch is *intrinsically one–dimensional*, or *id1*, if the structure it contains is aligned in one direction. In other words, if there exists an orientation for which this patch is translation invariant. This is the case for edges (two adjacent areas with contrasting intensity) or for a contrasted line splitting an otherwise homogeneous area (*cf.* examples of phase in Fig. 2.2).

Two–dimensional (id2): A local image patch is *intrinsically two–dimensional* or *id2* if its structure spans more than one axis. This is the case for corners, junctions and strongly textured patches.

Going beyond the classical discrete classification of intrinsic dimension used by Zetzsche and Barth (1990); Jähne (1997), we utilise a *continuous* formalisation that was proposed by Felsberg et al. (2006); Felsberg and Krüger (2003); Krüger and Felsberg (2003). This allows to describe the dimensionality of an image patch as a mixture of those three ideal classes. It has been shown (Krüger and Felsberg,

2003; Felsberg and Krüger, 2003) that the topological structure of the intrinsic dimensionality must be understood as a triangle that is spanned by two measures: *origin variance* μ and *line variance* ν. The origin variance describes the deviation of the energy from a concentration at the origin; the line variance describes the deviation from a line structure (see Fig. 2.4). We define the intrinsic dimension triangle such that each vertex corresponds to one ideal case of intrinsic dimension (homogeneous, edge or corner). The triangle's surface represents image patches that contain mixed aspects from these three ideal classes. For any image patch, the origin and line variances design a point in this intrinsic dimension triangle (see Fig. 2.4d) and the confidence for this patch to belong to each of the three classes is computed using barycentric coordinates (see, *e.g.*, (Coxeter, 1969)); The confidence $c_{idx}(\boldsymbol{x})$ that a local patch belongs to one of the ideal classes (id0, id1, or id2) is the area of the sub–triangle defined by the origin and line variance of the patch, and by the ideal cases for the two other classes of intrinsic dimensions — see Fig. 2.4. Furthermore, because the three classes of dimensionality are mutually exclusive, we have the following equality:

$$c_{id0}(\boldsymbol{x}) + c_{id1}(\boldsymbol{x}) + c_{id2}(\boldsymbol{x}) = 1 \tag{2.2}$$

at any location $\boldsymbol{x}$ in the image.

In the present work we will only make use of the intrinsic dimensionality as an interest point detector: as we choose to focus only on contour structures, we only need to consider intrinsicly *one–dimensional* patches. Thus, we will extract interest points and create primitives at locations that satisfy $\mu(\boldsymbol{x}) > \tau_\mu$, and $\nu(\boldsymbol{x}) < \tau_\nu$, where $\tau_\mu = 0.3$ and $\tau_\mu = 0.3$. [3]

2.2.3 Sampling and sub–pixel localisation

Based on the pixel–wise processing described above, we now want to extract a condensed interpretation of a local image patch by selecting a sparse set of locations where visual modalities become associated. An important aspect of the condensation scheme is that all main parameters can be derived from one property of the basic filter operations called *line/edge bifurcation distance*.

Definition 2.2.1. *The line/edge bifurcation distance d_{leb} for a given scale is the minimal distance between two edges for them to produce two distinct amplitude maxima.*

Hence, a double edge will be represented by a pair of edge primitives if its width is larger than d_{leb},

[3] A similar effect could be achieved by applying a threshold to $c_{id1}(\boldsymbol{x})$.

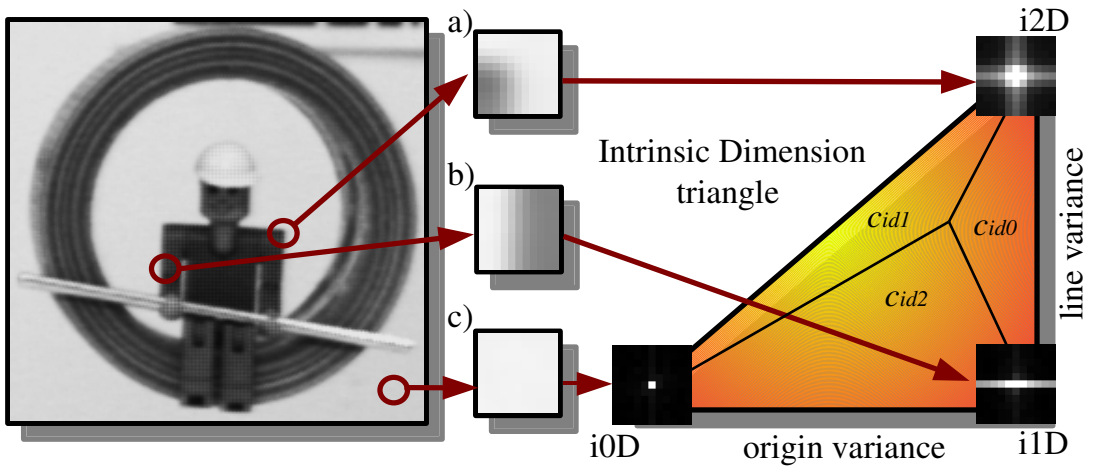


Figure 2.4: Illustration of the triangular topology of the intrinsic dimension — see (Felsberg et al., 2006). This figure exemplifies the three ideal intrinsic dimension classes: a) intrinsically zero–dimensional (id0); b) intrinsically one–dimensional (id1); and c) intrinsically two–dimensional (id2). Every image patch lie in a triangle spanning the space between these three ideal cases (see right). The horizontal axis is the origin variance and the vertical axis is the line variance. The confidence in each intrinsic dimension class is computed as the barycentric coordinates of the resulting point in the triangle.

by only one line primitive otherwise. Fig. 2.5(a) shows a narrow triangle. Vertical sections of the local amplitude in the vicinity of the vertex (to the right) have only one maximum that splits into two distinct maxima farther away from the vertex, where the distance between the two edges is larger. The line/edge bifurcation is illustrated in Fig. 2.5(b).

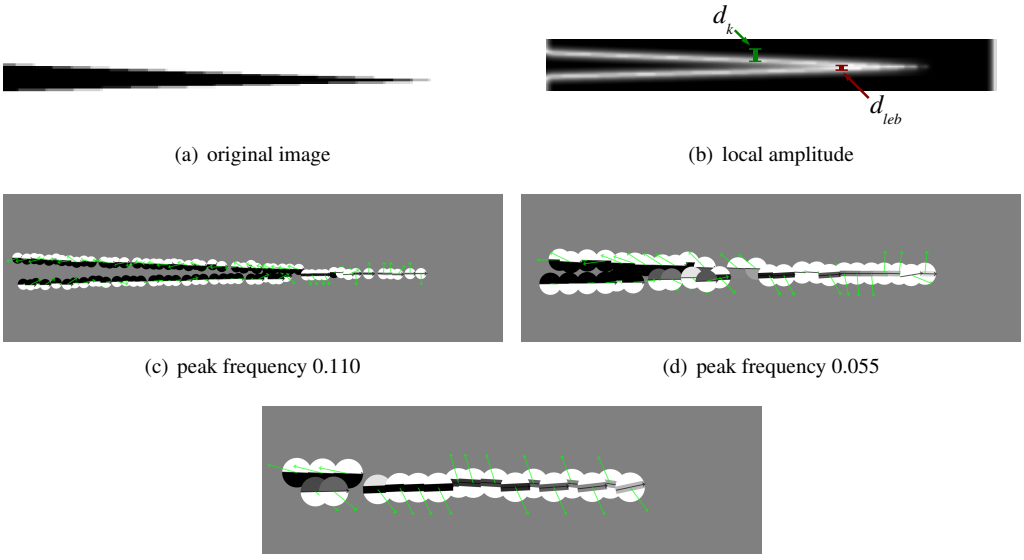
For a line or an edge, the position $\boldsymbol{x}_{id1}^{(k,l)}$ can be defined through energy maxima that are organised as a one–dimensional manifold. Therefore, an equidistant sampling along these energy maxima is appropriate — see Fig. 2.3b). For this, we look within the area $A^{(k,l)}$ for the energy maximum along a line orthogonal to the orientation at $A_c^{(k,l)}$:

$$\boldsymbol{x}_{id1}^{(k,l)} = \max_{\boldsymbol{x} \in g^{(k,l)}} m(\boldsymbol{x}), \tag{2.3}$$

where $g^{(k,l)}$ is a local line going through $A_c^{(k,l)}$ with orientation perpendicular to $\theta(A_c^{(k,l)})$.

Fig. 2.5(c), (d), and (e) show the primitives extracted after condensation for the three scales used in the present work — for peak frequencies of 0.11, 0.055 and 0.027, respectively.

Having discarded intrinsically zero–dimensional locations, we still face some redundancy in the image, and this in two ways: first an edge in the image will create a line of high id1 with a certain thickness. Nonetheless, it is only one edge, and therefore only one primitive should be extracted. Secondly, although



(a) original image (b) local amplitude

(c) peak frequency 0.110 (d) peak frequency 0.055

(e) peak frequency 0.027

Figure 2.5: Definition of the elimination parameters d_{leb} and d_k. See text.

all locations along a contour have high id1 confidence, we believe that such a contour is more efficiently represented by a sparser chain of primitives — see property 1.2.2.

First, we apply a winner–take–all competition each primitive and its immediate neighbours, based on an hexagonal grid bucketing. As a result, at most one primitive can be extracted in each single cell of the grid. This raises the following dilemma:

- Proximate, yet distinct, interest points should be preserved. For example, in the triangle in Fig. 2.5 two edges converge. At some point, double edges become interpreted as one line, position shift from the edges to the centre of this line, and phase becomes 0 or $\pm\pi$. Until there, the triangle is represented by two edges with phase $\pm\frac{\pi}{2}$. Hence, the elimination process should not discard these 'independent' edges although they can be rather close to each other. The limit of separability is the line/edge bifurcation distance d_{leb} defined above.

- Distant, yet redundant, interest points should be discarded. An edge will generate significant response within a radius d_k, that is larger than d_{leb}. As a consequence, eliminating candidates closer than d_{leb} preserves numerous redundant interest points. Conversely, eliminating candidates distant by less than d_k discards some distinct interest points.

The following describes a two step process that contends with the above dilemma.

Elimination based on the line/edge bifurcation distance d_{leb}

First, all interest points $\boldsymbol{x}^{(k,l)}$ become ordered according to the associated amplitude $m(\boldsymbol{x}^{(k,l)})$. Starting with the candidates with highest local amplitude, we discard all other candidates $\boldsymbol{x}^{(k',l')}$ within a radius d_{leb}.[4] Since we order interest points according to the local amplitude, a candidate corresponding to a stronger structure suppresses candidates with weaker structure. Thereby, all non–distinct edges (according to the line/edge bifurcation distance) become deleted but redundant edges are still being preserved.

Elimination based on the influence radius d_k

Edges can significantly affect the local magnitude within a radius d_k. In the second elimination step, starting again from the candidates with the highest local amplitude, the distance between pairs of remaining candidates is compared to d_k, empirically approximated by $d_k = 2.2d_{\text{leb}}$. For a pair of intrinsically two–dimensional structures it is sufficient to have a distance smaller than d_{leb} since they naturally represent maxima in the amplitude representation (Felsberg and Sommer, 2001). For an intrinsically one–dimensional structure, there will be a slant in the local amplitude surface at the redundant structure reaching its maximum at the edge/line structure and decreasing with distance from the edge (see Figs. 2.5 and 2.6). This slant can be checked to distinguish spatially close yet independent structures, that we want to keep, and nearby redundant structures, that we want to discard: For each candidate in a pair with distance smaller d_k, we test whether the structure is an amplitude maximum, along a line orthogonal to the local orientation. This is achieved by comparing each candidate's amplitude to its direct neighbours, on both sides of the edge, as indicated by the local orientation.[5] Then, redundant structures, *i.e.*, interest points that are not local maxima, are discarded.

2.2.4 Association of visual modalities

Because the interest points are extracted on edges, the symbolic descriptor is designed to describe a local edge in the image. The local phase and orientation of the edge is provided by the monogenic signal, and

[4] Note that for the quality of the process it is important that all positions are computed with sub–pixel accuracy already at this stage.

[5] Note that the criterion 'local maxima' that is applicable for id2 structures can not be applied since edge like structures form a ridge in the local amplitude surface (see Fig. 2.5).

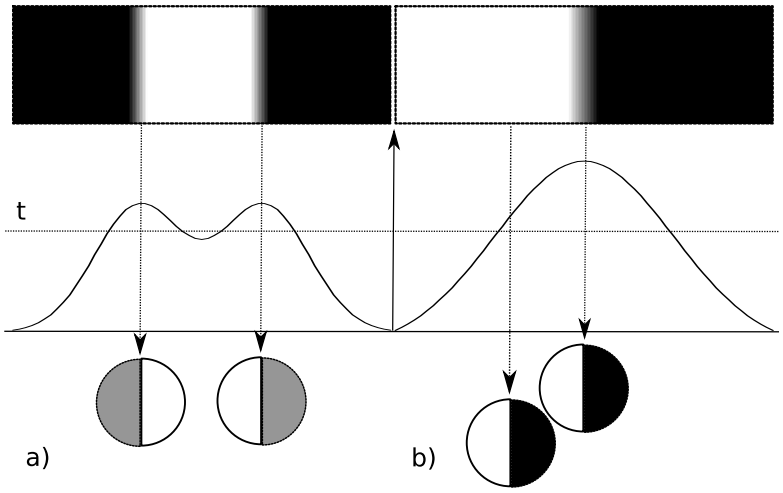


Figure 2.6: Extraction of redundant primitives due to the slant in the amplitude surface. (a) two interest points are correctly extracted; (b) because of the mild decay of the amplitude curve, the edge provokes the extraction of a distant, erroneous interest point (the amplitude of the response at this point is still above a given threshold t).

interpolated at the interest point's sub–pixel location. The colour is sampled on both sides of the line, as explained in section 2.2.4. The local optical flow is also sampled at this location — see section 2.2.4. The resulting symbolic descriptor, called a *2D–primitive* in this thesis, is described in section 2.2.4.

Colour

In order to represent accurately the colour structure of the edge, the colour information held by a 2D–primitive is made of several components. Also, we have seen that, depending on the phase, the 2D–primitive may express a step–edge or a line–like structure. Consequently, the colour information is defined relatively to the phase: if $\frac{\pi}{4} \leq |\omega| < \frac{3\pi}{4}$ (indicating an edge between two surfaces) the colour information is sampled on the left and right sides of the central line ($\boldsymbol{c} = (\boldsymbol{c}_l, \boldsymbol{c}_r)$.). Otherwise, the phase indicates a line and the colour is sampled not only on the left and right sides, but also in the middle encoding the colour on the line itself ($\boldsymbol{c} = (\boldsymbol{c}_l, \boldsymbol{c}_m, \boldsymbol{c}_r)$).

The RGB colour space has the advantage of being readily available from most image format. Yet, it involves a non–intuitive representation of the colour space: a fully saturated red will have for coordinates $R = (1, 0, 0)$ but a fully saturated yellow $Y = (1, 1, 0)$. For this reason, we will use the HSI colour space

for encoding the colour modality, and computing distances in this colour space.

Theoretically, the HSI colour space fails to be perceptually uniform (see (Sangwine and Horne, 1998)), unlike more sophisticated spaces like Munsell (also called HVC). On the other hand, the conversion from RGB to Munsell is non trivial, requiring the use of either correspondence tables (hence loosing accuracy) or heavy conversion operations. For this reason we will content ourselves with the HSI colour space in this work (note also that the performance of the colour modality in the algorithms described is always very good).

Optical flow

The projection of the 3D motion of the scene onto image pixels is called the *Motion Field*. From a sequence of images it is possible to estimate the apparent motion of brightness patterns in the image. This is called the *Optical Flow*. There is a fundamental difference between the two. For example, a sphere with a smooth surface rotating around its own axis under constant illumination would have a motion field describing this rotation, yet no apparent motion would be described by the optical flow (see (Horn, 1986)). It is generally agreed that the optical flow is the best approximation of the motion field that is in general attainable from the raw image data.

Kalkan et al. (2005) compared the performance of optic flow algorithms depending on the intrinsic dimensionality, *i.e.*, the effect of the aperture problem and the quality on low contrast structures. It appears that different optic flow algorithms are optimal in different contexts. In our system, we primarily use the algorithm proposed by Nagel and Enkelmann (1986), because it gives stable estimates of the normal flow at id1 structures.

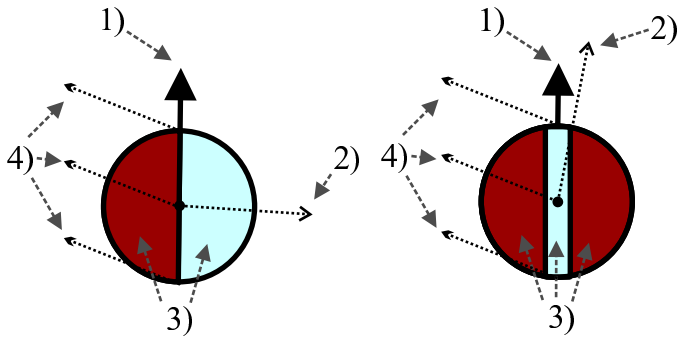
In the following we will write the local optic flow vector $\boldsymbol{f} = (f_u, f_v)^\top$.

The primitive descriptor

At each interest point a primitive is extracted, containing the aforementioned multi–modal description of the surrounding image patch.

This primitive is fully described by the vector:

$$\boldsymbol{\pi} = (\boldsymbol{x}, \theta, \omega, \boldsymbol{c}, \boldsymbol{f}, \lambda)^T \tag{2.4}$$



(a) An edge Primitive (b) A line Primitive

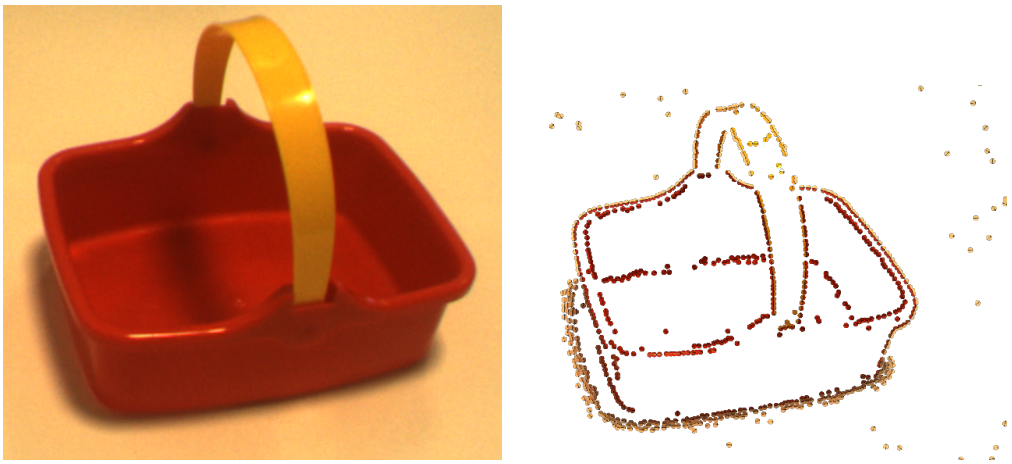
Figure 2.7: Illustration of the symbolic representation of a Primitive for a i1D interpretation, for a) a bright-to-dark step–edge (phase $\omega \neq 0$) and b) a bright line on dark background (phase $\omega \neq \frac{\pi}{2}$. 1) represents the orientation of the Primitive, 2) the phase, 3) the colour and 4) the optic flow.

where $\boldsymbol{x}$ contains the sub–pixel localisation of the feature, θ, the orientation (in the range $[0, \pi[$), ω, the phase (in the range $[-\pi, +\pi[$), $\boldsymbol{c}$, the colour (as defined above), $\boldsymbol{f}$, the optic flow, and λ, the size of the image area the feature describes (therefore we set $\lambda = d_k$).

Those local image descriptors are hereafter called *2D–primitives*. The set of 2D–primitives extracted from an image is called *Image Representation* $\boldsymbol{\mathcal{I}}$ The result of this processing can be seen in Fig. 2.8. For a detailed description of the 2D–primitive extraction process we refer to (Krüger et al., 2007). Fig. 2.8 shows the primitives extracted, with an origin variance $\mu > 0.3$ and a line variance $\nu < 0.3$ for the three scales considered in this work: namely for peak frequencies of 0.110 (Fig. 2.8b), 0.055 (Fig. 2.8c), and 0.027 (Fig. 2.8d). Different scales highlight different structures in the scene. Furthermore, lower peak frequency (*i.e.*, coarse scale) removes image noise and generates less spurious primitives, whereas smaller image structures become neglected — see (Lindeberg, 1998a; Elder and Zucker, 1998) for a discussion of the effect of scale in edge detection.

Orientation ambiguity and primitive switching

We explained earlier that the monogenic signal computation provides us with an estimation of the local orientation. Assuming that a contour is present at this location, this orientation value is an estimate of the local tangent to this contour. Hence, a 2D–primitive's orientation θ is bound within the interval $[0, \pi[$. For the phase and colour modalities to be defined unambiguously, both sides of the contour need to be identified. For this reason we arbitrarily assign a direction vector to this orientation. The direction vector



(a) original image (b) peak frequency 0.110

(c) peak frequency 0.055 (d) peak frequency 0.027

Figure 2.8: (a) one image of an object. (b,c,d): id1 primitives extracted, with origin variance $\mu > 0.3$ and line variance $\nu < 0.3$, for peak frequencies of: (b) 0.110, (c) 0.055, and (d) 0.027.

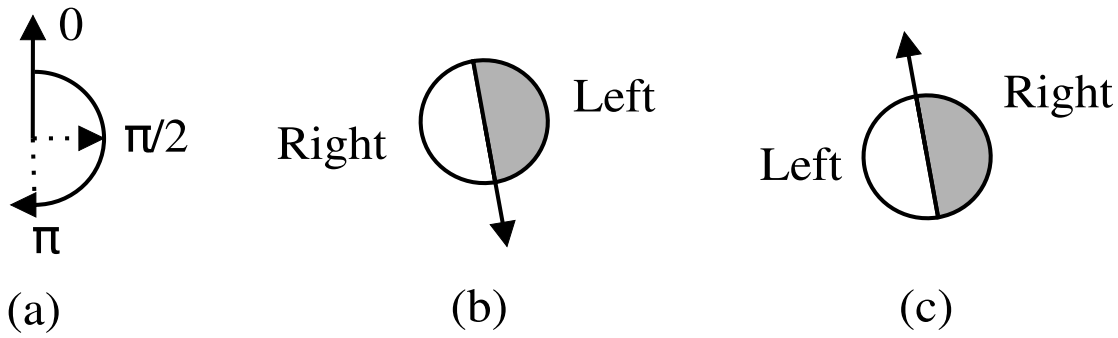


Figure 2.9: Illustration of the orientation ambiguity when interpreting 2D–primitives. Because 2D–primitives describe local edges, only their orientation is well defined: the actual direction is meaningless. Hence we need to choose an orientation convention, shown in (a), where θ is bound within $[0, \pi[$, where 0 encodes a vertical edge and $\frac{\pi}{2}$ an horizontal one. (b) and (c) show two different, yet equivalent, descriptions of the same edge. According to our convention, only b) is valid, ensuring the uniqueness of an edge's encoding.

$\boldsymbol{t}$ of a 2D–primitive is defined directly from its orientation θ as the following vector:

$$\boldsymbol{t} = \begin{pmatrix} \sin(\theta) \\ -\cos(\theta) \end{pmatrix} \tag{2.5}$$

thus we can identify each side of the contour as 'left' and 'right' areas relatively to this vector. As illustrated in Fig. 2.9, one image patch can have two primitive interpretations:

1. a direction of θ with the dark colour on the right side, called the *a priori* interpretation $\boldsymbol{\pi}$ — see Fig. 2.9(b).

2. a direction of $\theta + \pi$ where the dark colour is the left side, the alternative interpretation $\bar{\boldsymbol{\pi}}$ — see Fig. 2.9(c).

Note that *a priori* orientation is indeed within $[0, \pi[$ — see Fig. 2.9(a) — whereas the alternative interpretation's is within $[\pi, 2\pi[$.

Because the phase and colour properties are defined relatively to this assumed direction, their values also differ depending on the interpretation:

$$\begin{cases} \bar{\boldsymbol{t}} &= -\boldsymbol{t} \\ \bar{\theta} &= \theta + \pi \\ \bar{\omega} &= -\omega \\ \bar{\boldsymbol{c}_l} &= \boldsymbol{c}_r \\ \bar{\boldsymbol{c}_m} &= \boldsymbol{c}_m \\ \bar{\boldsymbol{c}'_r} &= \boldsymbol{c}_l \end{cases} \tag{2.6}$$

The other properties of the primitives remain the same for the two possible interpretations of the orientation. At the time of the primitive extraction, the *a priori* interpretation is assumed. Later processing may require the use of the alternative interpretation: we call the operation creating $\bar{\boldsymbol{\pi}}$ from $\boldsymbol{\pi}$ the *switching* of the primitive:

$$\mathcal{S} : \boldsymbol{\pi} \longrightarrow \bar{\boldsymbol{\pi}} \tag{2.7}$$

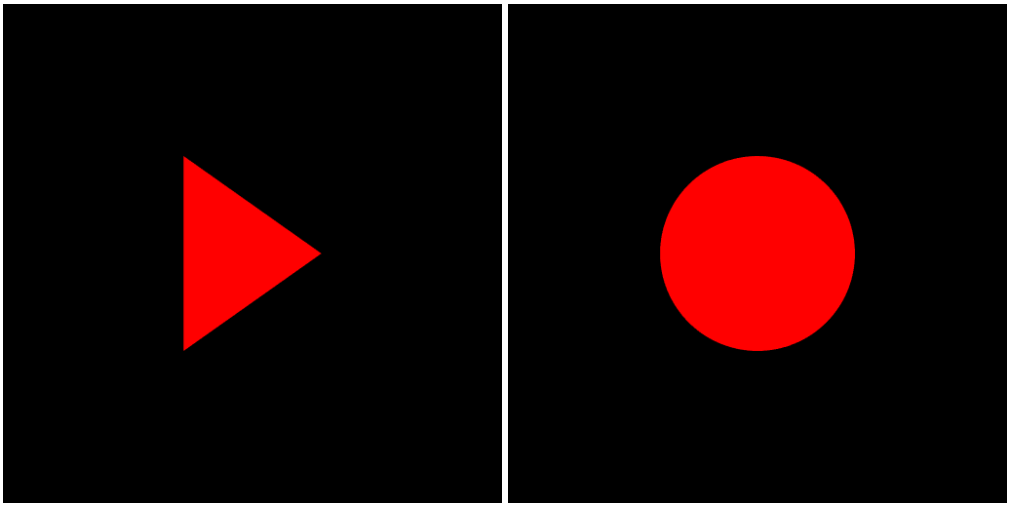
This operation is required to solve the ambiguity intrinsic to the orientation definition, during grouping, stereo and reconstruction — see sections 3.2.2 and 4.2.1.

2.2.5 Accuracy of the extracted primitives

The primitives are located with sub–pixel accuracy, using the assumption that they are linear structures of low local curvature (this holds because the non–id1 patches were neglected, see (Krüger et al., 2007)). We evaluated the accuracy of the primitives that were extracted in two simple artificial scenarios featuring, respectively, a red triangle or a red circle on a black background. In both cases, the primitives should be positionned exactly along the boundaries of the shape, and should feature a phase of $\pm\frac{\pi}{2}$ (for a pure step edge). The scenarios and the extracted primitives are illustrated in Fig. 2.10.

Fig. 2.11 shows the error in the localisation (a), orientation (b) and phase (c) of the primitives that were extracted in the triangle (solid line) and the circle (dashed line) scenarios, for an amount of colour noise from 0 to 10%. A noise of $n\%$ means that, for all pixels, each RGB component c is altered by a random, normally distributed value δc

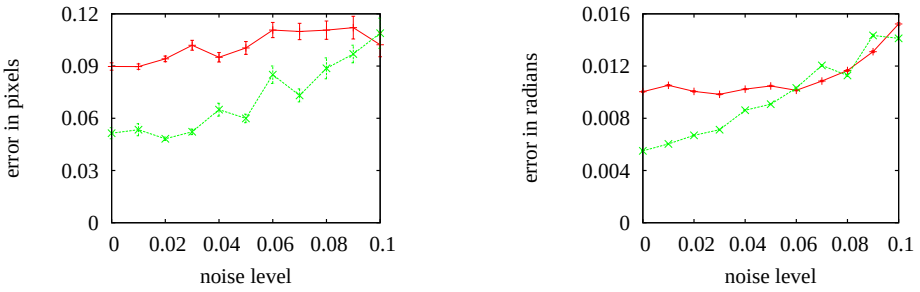
$$\bar{c} = c + \delta c \tag{2.8}$$



(a) images

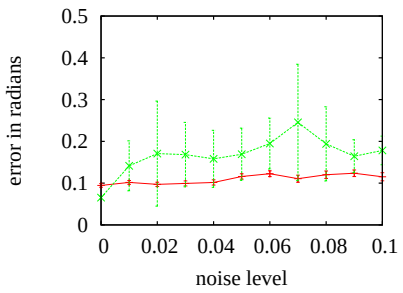
(b) primitives

Figure 2.10: Illustration of the primitives extracted from two simple artificial sequences, featuring a triangle (left) and a circle (right).



(a) localisation

(b) orientation



(c) phase

Figure 2.11: The graphs a, b and c illustrate the accuracy of, respectively, the localisation, orientation and phase of the extracted primitives; The solid line show the accuracy on the triangle scene whereas the dashed line show the accuracy for the circle scene, furthermore the vertical error bars show the variance.

where $100 \cdot |\delta c| \leq n$. Therefore, adding a noise of 100% would generate a completely random image.

In Fig. 2.11, the vertical bars on the curves show the variance of the error. In this graph, we can see that the localisation error is always below 0.1 pixels. The orientation error is below 0.02 radians and the phase error is around 0.2 radians. The phase is sensitive to localisation error, therefore a minor inaccuracy in the primitive localisation lead to significant, yet small, error in the phase estimation.

2.3 Primitive Metrics

In this section we will define a metric in the primitive space. Effectively, as a primitive is defined as a multi–modal vector, the primitives' metric space is be defined as a combination of the different modalities' metrics. Depending on the context and the purpose (grouping, stereopsis, temporal matching, *etc.*),

some of the modal distances may be discarded or replaced using application specific knowledge, by a more adequate measure — *e.g.*, the good continuation constraint in chapter 3.

Phase

Because the possible phase values form a continuum, we can define a metric in the phase–orientation space, taking care to preserve its specific topology:

$$d_\omega(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j) = |\arctan(\sin(\omega_j - \omega_i), \cos(\omega_j - \omega_i))| \tag{2.9}$$

This formulation ensures the preservation of the orientation–phase space half–torus topology, as discussed in section 2.2.1 and Fig. 2.2.

Colour

The colour information held by a primitive is composite: it consists of 2 or 3 colour components (depending on the phase value) and its interpretation is relative to the primitive orientation. Hence, the colour similarity between two primitives requires a more complex formulation. If the phase value is in $\frac{\pi}{4} \leq |\omega| < \frac{3\pi}{4}$, the 2D–primitive describes a step edge and the colour information is sampled over the 'left' and 'right' sides of this step–edge. Otherwise, the 2D–primitive describes a line structure and colour is additionally sampled along the line itself. Thus, if *both* primitives have a 'middle' colour value, the three colour components are compared. In all other cases, only the colour values for the 'left' and 'right' sides of the edge are compared (see equation 2.10). Also, and depending on the orientation of both primitives, the 'left' and 'right' colour components of one of the primitives may need to be switched as stated in section 2.2.4.

$$d_c(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j) = \begin{cases} \frac{d_{c,l}(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j)+d_{c,r}(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j)}{2} & \text{if } \frac{\pi}{4} \leq |\omega_i|, |\omega_j| < \frac{3\pi}{4} \\ \frac{d_{c,l}(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j)+d_{c,m}(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j)+d_{c,r}(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j)}{3} & \text{else.} \end{cases} \tag{2.10}$$

The colour distance is computed in the HSV space, which is a variant of the so called *perceptual* (IHS) colour spaces. For an extensive description of colour spaces see, e.g., (Palus, 1998). This colour space decomposes colour values into three components:

- The *hue* of the colour H.

- The *saturation S* is the relative purity of this colour; *e.g.*, a saturation of 0 means white for all hues, and a saturation of 1 means a pure colour.
- The *value V* is an estimation of the intensity (here estimated as the highest of the RGB components).

This space allows to describe colours in perceptual terms, rather than the red, green, and blue components. Furthermore, it holds the additional advantage that we can isolate the intensity of the colour. The 2D–primitives' phase encodes the intensity transition across a 2D–primitive. Consequently, we discard the value component from the colour modality, and evaluate the distance using only two components (hue and saturation), rather than three. The similarity for each side is computed as follows:

$$d_{\boldsymbol{c},x\in\{l,m,r\}}(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j) = 1 - \frac{|\tan^{-1}(\sin(H_j^x - H_i^x), \cos(H_j^x - H_i^x))| + |S_j^x - S_i^x|}{2} \tag{2.11}$$

H_i^l, S_i^l and V_i^l are respectively the hue, the saturation and the value of the left part of the primitive $\boldsymbol{\pi}_i$. Alternatively, the following metric make use of the three components, including intensity:

$$d_{\boldsymbol{c},x\in\{l,m,r\}}(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j) = 1 - \frac{|\tan^{-1}(\sin(H_j^x - H_i^x), \cos(H_j^x - H_i^x))| + |S_j^x - S_i^x| + |I_j^x - I_i^x|}{3} \tag{2.12}$$

In the following, and unless stated otherwise, we will use the two components colour distance.

Optic flow

We want to define an optic flow metric such that the distance $d_f\left(\boldsymbol{f}_i, \boldsymbol{f}_j\right)$ is large if their orientation is widely different; if they have a similar orientation, distance should be small for vectors of similar magnitude, and increase with magnitude difference. In their survey of optic flow algorithms, Barron et al. (1994) suggested the use of the arc–cosine of normalised vectors' dot product as an error measurement for optic flow.

Given an optic flow vector $\boldsymbol{f} = (x, y)^T$, we will consider the equivalent homogeneous vector $\tilde{\boldsymbol{f}} = (x, y, 1)^T$ Then arc–cosine of the normalised dot product provides us with a suitable optic flow metric:

$$d_f(\boldsymbol{\pi}_i,\boldsymbol{\pi}_j) = \frac{1}{\pi}\arccos\left(\frac{\tilde{\boldsymbol{f}}_i \cdot \tilde{\boldsymbol{f}}_j}{||\tilde{\boldsymbol{f}}_i|| \cdot ||\tilde{\boldsymbol{f}}_j||}\right) \tag{2.13}$$

This formula effectively compares magnitude as well as orientation of the optic flow vectors: a large difference in the vectors' orientation always leads to a high distance value, irrespectively of the vectors' magnitude, which is consistent with intuition. The Fig. 2.12 illustrates the optic flow metric in equation (2.13): the height and colour of each point $(x, y)^T$ on the surface represents the distance between the two optic flow vectors $\boldsymbol{f}_j = (x, y)^T$ and $\boldsymbol{f}_i = (1, 0)^T$ (represented by the blue arrow). In this plot, red indicates large distances, blue small distances.

2.4 Discussion

In this chapter, we described the primitive based image representation first introduced by Krüger et al. (2007), that is used in the following of this thesis. This image representation shows several notable qualities:

1. The multi–modal vector encoding a visual primitive achieves a data condensation of 95%, relatively to the image area it was extracted from. The primitives' sparseness limits the redundancy inherent to natural images by directly encoding image semantics. Discarding intrinsically zero–dimensional areas further reduces the amount of data describing an image, with minimal information loss (Elder, 1999). The resulting image representation holds a dense and complete description of the edges present in the image.

2. The ambiguity of a feature matching task depends on two factors: the number of candidates, and the feature distinctiveness. Primitives effectively reduce this ambiguity on both sides: First, the rich information carried by the primitives make them more distinctive than, *e.g.*, raw pixel information, local orientation, or phase taken separately. Second, because the primitive representation of an image is sparse, the ambiguity faced by a matching algorithm is greatly reduced compared to dense methods — see chapters 4, 6, and 7.

3. The notion of intrinsic dimension provides a semantic interpretation of the local signal, embodied by the primitive descriptor. Let us emphasise that the decision to only consider intrinsically one–dimensional structures in this work was made in order to limit the scope of the research, and is in no way a limitation of the chosen image representation. Nonetheless, it would require an adaptation of the primitives' symbolic description to appropriately describe intrinsically zero– or

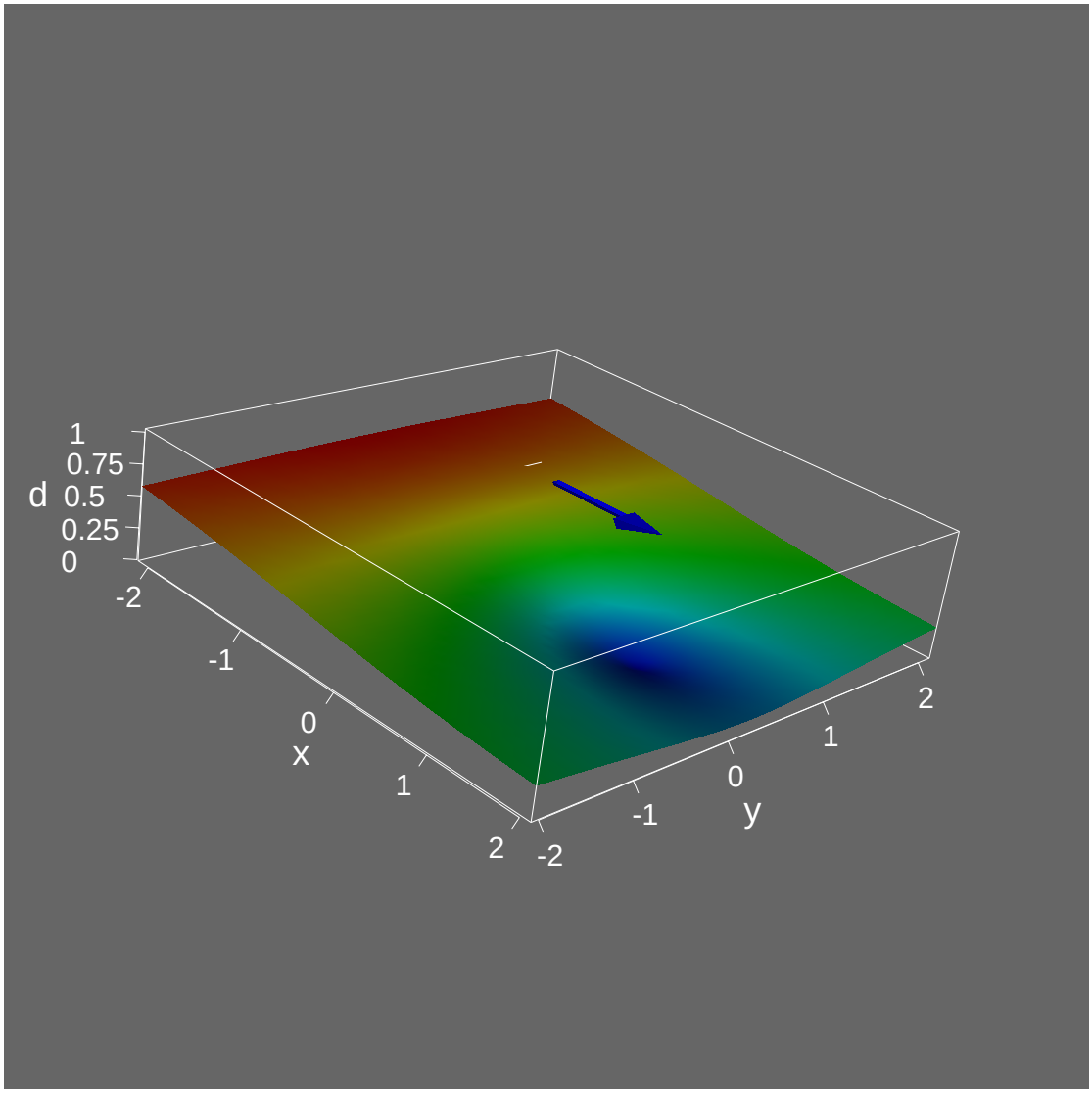


Figure 2.12: Graph of the optic flow metric used in this work. Each position in the surface represents an optic flow vector $\boldsymbol{f}_j = (x, y)^T$. The height of the surface and the colour represent the optic flow distance of this vector with an optic flow of $\boldsymbol{f}_i = (1, 0)^T$ (represented by the blue arrow). The colour red stands for large distance, blue for small distance.

two–dimensional locations.

Those three qualities are essential in order to draw the statistical relations that underly the processes presented hereafter. Indeed, the combinatorial explosion that is produced by attempting to draw complex relations at the pixel level, render such schemes practically intractable without a preliminary reduction of the feature space. Also, the lack of distinctiveness of local pixel information only allows for weak relations to be drawn, which are therefore sensitive to ambient noise. The semantics associated to the primitives are critical because they justify the relations themselves. Due to its distant relation to scene semantics, pixel information requires additional assumptions in order to draw any kind of contextual relation (often, that the scene is piecewise planar).

It is worth mentioning that there is some evidence for such a condensed, retinotipic, multi–modal processing of the visual information in the human visual system in the primary visual cortex's hyper-columns.

In the primate visual system, information gathered in the retina projects to the primary visual cortex (V1) (Wurtz and Kandel, 2000a). The structure of V1, that was investigated by Hubel and Wiesel (1962, 1969), is a retinotopic map showing a specific and repetitive pattern of substructures called hyper–columns. Hyper–columns themselves contain so–called orientation columns and blobs which are mainly involved in colour processing. However, in an orientation column, we find cells sensitive, beside orientation, to disparity (Barlow et al., 1967; Parker and Cumming, 2001), local motion (Wurtz and Kandel, 2000b), colour (Hubel and Wiesel, 1969), and phase (Jones and Palmer, 1987). Also, cells responding to junction–like structures were measured (Shevelev et al., 1995). Moreover, cells in V1 are locally densely connected. Therefore, it is believed that the visual cortex, in its early stages, processes local, multi–modal feature descriptions. For a more in–depth discussion of the analogy between early cortical connectivity and visual primitives, we refer to Krüger et al. (2004).

In the following chapters we will describe a framework building on this image representation and aiming to provide a robust and general symbolic representation of the visual information.

Chapter 3

Formalisation of the Organisation of the Primitives

No object is mysterious. The mystery is your eye.

- Elizabeth Bowen

In the previous chapter we described an image representation based on local edge descriptors we called primitives. One of the challenges of visual perception is to come from local image descriptors (pixel, corner, primitive, *etc.*), that are dependent on sampling scale, to a description of the global image structures (*e.g.*, image contours and shapes), in a manner similar to Marr's *full primal sketch* (Marr, 1982). In order to bridge this gap we need to bind similar primitives into global contours. This is one aspect, amongst others, of *perceptual grouping*: psychophysical studies have observed that the human visual system is apt at grouping together parts of a broken contour into a whole — see, *e.g.*, (Field et al., 1993).

Psychophysical studies have shown that this perceptual grouping is strongly biased, leading to so–called "visual illusions": the erroneous perception of contours or shapes in unusual configurations. The rules driving perceptual grouping were investigated by the *Gestalt* psychologists (Koffka, 1935; Wertheimer, 1935; Köhler, 1947). For example, in Fig. 3.1a) a version of the Kanisza square is drawn: the perceptual impression is that of a white square occluding four black circles, while the objective figure is only four black 'pacman' figures arranged in a regular fashion. Fig. 3.1b,c illustrate some other biases

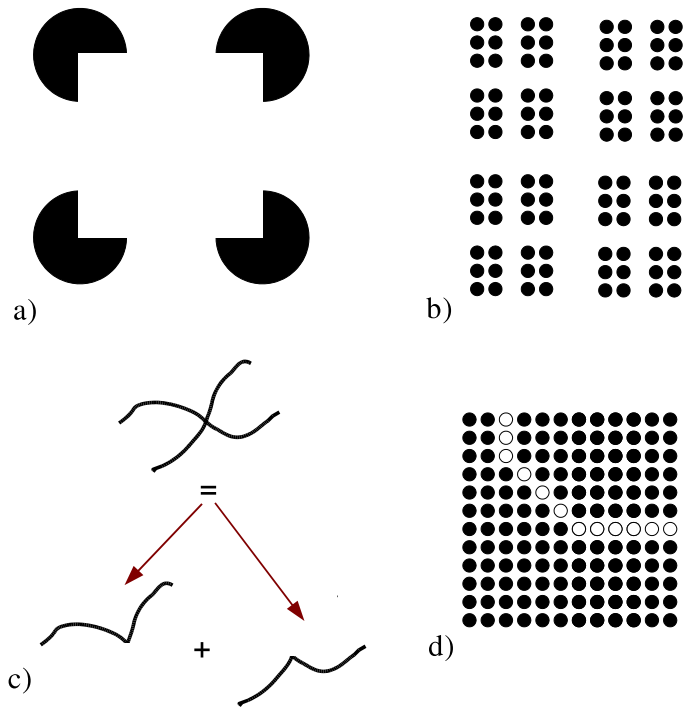


Figure 3.1: Illustration of some of the *Gestalt* laws: **a)** The Kanisza square. Here we perceive a white rectangle (that is objectively there) occluding the four black circles. **b)** Proximity: the dots that are located more tightly together are more strongly grouped. **c)** Good continuation: in this figure the visual system prefers to assume a crossing of two smooth curves, whereas it can also be two broken curves joined by perspective. **d)** Similarity: Here all the dots have the same size and are regularly positioned. The string of white dots appears to be part of a separate structure, occluding the array of black dots.

in the visual system: dots are perceived as one group when they: 1) are proximate (see Fig. 3.1b), 2) form a continuous curve (see Fig. 3.1c), or 3) have similar qualities (*e.g.*, intensity in Fig. 3.1d). These observations suggest that the visual system interprets visual information by using general rules (such as good continuation, proximity, *etc.*) to group local features together. Already in 1953, Brunswik and Kamiya suggested that these *Gestalt* laws should be a direct consequence of the statistics of natural images. This has more recently been demonstrated, in the case of perceptual grouping, by several independent research groups (Krüger, 1998b; Elder and Goldberg, 1998; Geisler et al., 2001).

Amir and Lindenbaum (1998) proposed to consider contour grouping as the combination of two different mechanisms: 1) the definition of a pairwise affinities between feature, and the construction of a relational graph, hereafter called *affinity matrix*, and 2) the clustering this space into global groups.

Affinity matrix: The first aspect of the problem is the generation of the affinity matrix between local features: Given two primitives $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$ in the image, we want the affinity between those two 2D–primitives to express the likelihood that they both describe the same contour.

Definition 3.0.1. *Two primitives $\boldsymbol{\pi}_i, \boldsymbol{\pi}_j \in \boldsymbol{\mathcal{I}}$ that describe the same contour $C \in \boldsymbol{\mathcal{I}}$ will henceforth be called a* link $g_{i,j}$*, and the likelihood of this link will be called the* affinity $p[g_{i,j}] = \boldsymbol{A}_{i,j}$.

Affinity measures in the literature commonly involve: proximity (Shi and Malik, 2000; Perona and Freeman, 1998), collinearity (Perona and Freeman, 1998), co–circularity (Parent and Zucker, 1989; Amir and Lindenbaum, 1998), common region (Sarkar and Soundararajan, 2000), or symmetry (Cham and Cipolla, 1996). On the other hand, the use of *Gestalt* law of similarity (*e.g.*, in colour, contrast, motion) has been the subject of little investigation (Sarkar and Soundararajan, 2000; Elder and Goldberg, 2002), although its statistical relevance has been shown in natural images by Krüger and Wörgötter (2002). In this work we will define the affinity as a combination of the geometric information (proximity, collinearity, and co-circularity) and the primitives' modal distances (colour, phase and optical flow, see section 2.3). It is convenient to see the result of such a process as a graph $(\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{L}})$, where the primitives $\boldsymbol{\pi}_i \in \boldsymbol{\mathcal{I}}$ are the nodes of the graph, and the links $g_{i,j} \in \boldsymbol{\mathcal{L}}$ are the edges of the graph.

Scene segmentation The second aspect of the problem is the segmentation: Given an image representation $\boldsymbol{\mathcal{I}}$, we want to obtain those subsets $\boldsymbol{\mathcal{C}} \subset \boldsymbol{\mathcal{I}}$ such as $\boldsymbol{\pi}_i \in \boldsymbol{\mathcal{C}}$ and $\boldsymbol{\pi}_j \in \boldsymbol{\mathcal{C}}$ is true if, and only if, $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$ belong to the same contour.[1] In the graph formalisation proposed earlier, such a group in $\boldsymbol{\mathcal{C}} \subset \boldsymbol{\mathcal{I}}$ is defined as a set of primitives $\boldsymbol{\mathcal{C}} \subset \boldsymbol{\mathcal{I}}$ and a set of links $\boldsymbol{\mathcal{G}} \subset \boldsymbol{\mathcal{L}}$ between those primitives, such as $(\boldsymbol{\mathcal{C}}, \boldsymbol{\mathcal{G}})$ is a *connected* sub–graph. This second problem has been widely addressed in the literature, using a variety of techniques including: graph cuts (Shi and Malik, 2000; Sarkar and Soundararajan, 2000), affinity normalisation (Perona and Freeman, 1998), dynamic programming (Sha'ashua and Ullman, 1990), probabilistic chaining (Crevier, 1999), *etc.*

In this chapter we will address the first half of the grouping problem for the 2D–primitive framework, namely the generation of the whole graph $(\boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{L}})$. It will become clear that the semi–local relations defined by $\boldsymbol{\mathcal{L}}$ are sufficient to model the existence of groups in the vicinity of a primitive. Extracting

[1] Note that this grouping of primitives over the image can alternatively be seen as a clustering problem. Indeed, when clustering datapoints one tries to define groups of points which stand closer together (for a given metric) and separated from other groups.

global contours is outside of the scope of the present work, focused on middle–level vision, but could be achieved from this graph using any of the classical methods cited above. In section 3.3 we will discuss the meaning of groups and isolated primitives in our framework. A novel, primitive–based, grouping algorithm is defined in section 3.2.4.

3.1 Literature review

Field et al. (1993) proposed a model of perceptual grouping in human vision where missing edges are inferred from an "association field" generated by all neighbouring edge points, backed by psychophysical studies of the perception of fragmentary closed contours embedded in noisy images. More recently, Elder and Goldberg (2002) proposed a Bayesian formalisation of perceptual grouping, elegantly combining cues of proximity, co–circularity, parallelism, and similarity. A similar approach including prior knowledge of the contours has been successfully used for lake contour extraction from aerial images by Elder et al. (2003).

Lowe (1987) discussed the importance of the Gestalt rules of collinearity, co–curvilinearity and simplicity for perceptual grouping. Ullman (1976) proposed a network model inferring the contour between two tangents as a pair of circular arcs meeting smoothly and minimising the total curvature. Parent and Zucker (1989) based their approach on curve consistency and co–circularity. Cham and Cipolla (1996) proposed to describe contours using *basis points* that can vary along the curve, effectively freeing the representation from the correspondence ambiguity that stems from the aperture problem. Perona and Freeman (1998) proposed an algorithm based on the factorisation of an affinity measure between local tangents. This affinity measure was effectively a mixed rule combining proximity, collinearity and co–circularity constraints. Guy and Medioni (1996) advocated a global grouping scheme based on an *extension field*. In this technique each point receives votes from all neighbourhood. Amir and Lindenbaum (1998) chose to divide the grouping problem into the two tasks of building an affinity graph, and partitioning this graph into groups using a standard clustering algorithm. Sarkar and Soundararajan (2000) used a stochastic automata onto a Bayesian framework to learn the network parameters from a set of training images.

In this chapter we present a contour grouping mechanism that takes full advantage of the multi–modal nature of the 2D–primitives. The likelihood for two primitives to be grouped is hereafter called *affinity*,

and is derived from a joint application of the *Gestalt* laws of proximity, good continuation, and similarity.

3.2 Definition of the affinity between primitives

As seen in chapter 2, a primitive is extracted from a local patch of the image therefore all of its modalities descend from this patch (*e.g.*, orientation, phase, colour, and optical flow). In the following we are defining an affinity measure (in the sense defined in the previous section) as a combination of modal metrics (see section 2.3), allowing us to estimate the likelihood for two primitives to describe the same contour. We regroup those criteria in two classes: a *geometric* criterion (implementing the *Gestalt* laws of proximity, collinearity, and co–circularity), and a *multi–modal* criterion (implementing the *Gestalt* law of similarity, applied to phase, colour and optical flow).

3.2.1 Geometric constraint

The position and orientation of the primitives are intrinsicly related: as primitives represent local contour descriptors, their positions are points along the edge, and the orientations can be seen as local tangents to the contour at these points. The Good Continuation law states that grouping is biased towards contours of smoothly changing properties, reflecting the assumption that natural object shapes are mostly made of such contours. Accordingly, the estimated likelihood of a contour is based upon the assumption that smooth structures are more likely to describe the scene's contours, and that jagged structures are more likely to be manifestations of erroneous or noisy data (or both):

Effectively, this likelihood is formulated as a combination of three basic constraints drawn upon the primitives' relative position and orientation; namely:

Proposition 3.2.1. *Law of Proximity: A contour $C \subset \mathcal{I}$ is more likely if it is described by a dense population of primitives, i.e., $\forall \boldsymbol{\pi}_i, \boldsymbol{\pi}_{i+1} \in C$, we have $d(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{i+1}) < \epsilon$, with ϵ a small quantity.*

Large gaps in the primitives' description of the contour are an indication that the contour might be, in fact, two contours collinear yet distinct. The proximity constraint $c_p\left[g_{i,j}\right]$, applied between two primitives $\boldsymbol{\pi}_i, \boldsymbol{\pi}_j \in \mathcal{I}$, is defined by the following equation:

$$c_p\left[\boldsymbol{\pi}_i, \boldsymbol{\pi}_j\right] = 1 - e^{-\sigma h\left(1 - \frac{d_E(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j)}{\lambda r}\right)}. \tag{3.1}$$

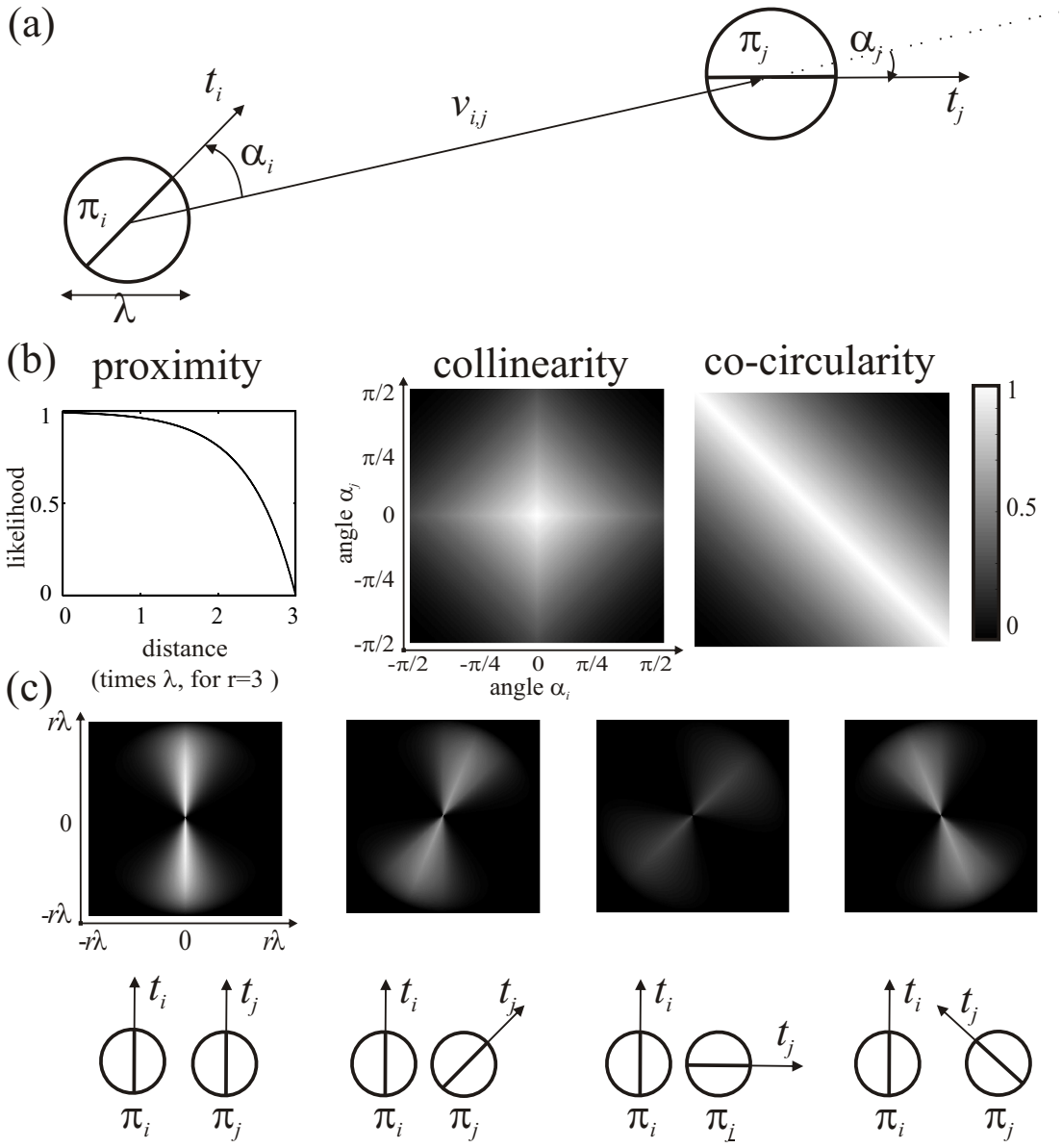


Figure 3.2: Geometric affinity between two primitives $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$. (a) illustration of the parameters: $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$ represent the orientation vectors of each primitive ($\|\boldsymbol{t}_i\| = \|\boldsymbol{t}_j\| = 1$). Note that the orientation being π–periodic, t and $-t$ are equivalent. The vector $\boldsymbol{v}_{i,j}$ joins the two centres of both primitives, such that $\|\boldsymbol{v}_{i,j}\| = d_E(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j)$, and α_i, α_j are the angles between $\boldsymbol{v}_{i,j}$ and the respective orientation of $\boldsymbol{\pi}_i, \boldsymbol{\pi}_j$ respectively. Note that those angles being resultant of the orientation, they are also π–periodic. (b) Left: link likelihood relative to the distance between two primitives (λ is the size of a primitive at this scale). Middle: CoCollinearityriterion, depending on the angles α_i and α_j (as defined in a)). Right: Co-circularity criterion as a function of α_i and α_j. (c) In those examples, the shade at each location shows the strength of the geometric affinity between $\boldsymbol{\pi}_j$ at this point and $\boldsymbol{\pi}_i$ at the centre of the square, for the shown orientations — respectively 0^o, 45^o, 90^o and -45^o. In those graphs white stands for high likelihood, and black for low.

There, σ is the steepness parameter of the function, set to 5 in our system, λ is the size of a primitive in pixels (the so–called line/edge bifurcation distance, see chapter 2), and λr is radius of the neighbourhood considered for the grouping. The value $d_E(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j)$ is the Euclidian distance in pixels separating the centres of the two primitives. This is illustrated in Fig. 3.2(b), left panel. The function $h(x)$ is defined such that:

$$h(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{otherwise} \end{cases} \tag{3.2}$$

Proposition 3.2.2. Collinearity: *A contour C is more likely if its local curvature is small, i.e., $\forall \boldsymbol{\pi}_i, \boldsymbol{\pi}_{i+1} \in C$ the local curvature is small.*

A sharp curve might be an indication of two intersecting or occluding contours. The collinearity constraint $c_{co}\left[\boldsymbol{\pi}_i, \boldsymbol{\pi}_j\right]$, applied between two primitives $\boldsymbol{\pi}_i, \boldsymbol{\pi}_j \in \boldsymbol{\mathcal{I}}$, is given by the following equation:

$$c_{co}\left[\boldsymbol{\pi}_i, \boldsymbol{\pi}_j\right] = 1 - \left|\sin\left(\frac{|\alpha_i| + |\alpha_j|}{2}\right)\right|, \tag{3.3}$$

where α_i and α_j are the angles between the line joining the two primitives centres and the orientation of $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$, respectively — see Fig. 3.2(a). A map of this function depending on α_i and α_j, is drawn in Fig. 3.2(b), central panel.

Proposition 3.2.3. Co–circularity: *A contour C is more likely if it is piecewise circular, and therefore if it has a locally constant curvature, i.e., $\forall \boldsymbol{\pi}_i, \boldsymbol{\pi}_{i+1} \in C$ there exist a circle Ω such that $\boldsymbol{\pi}_i, \boldsymbol{\pi}_{i+1}$ are tangent to Ω.*

The co–circularity constraint $c_{ci}\left[g_{i,j}\right]$, applied between two primitives $\boldsymbol{\pi}_i, \boldsymbol{\pi}_j \in \boldsymbol{\mathcal{I}}$, is given by the following equation:

$$c_{ci}\left[\boldsymbol{\pi}_i, \boldsymbol{\pi}_j\right] = 1 - \left|\sin\left(\frac{\alpha_i + \alpha_j}{2}\right)\right| \tag{3.4}$$

A map of this function, depending on α_i and α_j, is drawn in Fig. 3.2(b), right panel.

The different geometric configurations possible for a pair of primitives are illustrated in Fig. 3.3. Note that it is possible to have two primitives perfectly co–circular, but with a very high curvature (*e.g.*, Fig. 3.3(d) and therefore with a low collinearity rating. Conversely, it is possible to have two primitives nearly collinear but with incompatible curvature (*e.g.*, Fig. 3.3(a): this is the case for two parallel primitives, for example. Therefore although collinearity and co–circularity express two similar concepts they

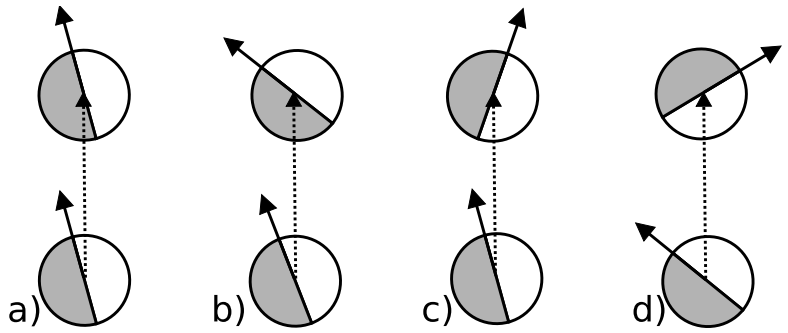


Figure 3.3: Illustration of the different geometric configuration for a pair of primitives: (a) parallel lines ($c\,[co]$ is high and $c\,[ci]$ is low), (b) changing curvature ($c\,[co]$ and $c\,[ci]$ are both low), (c) smooth curve ($c\,[co]$ and $c\,[ci]$ are both high), and (d) high local curvature ($c\,[co]$ is low and $c\,[ci]$ is high).

are nonetheless different — see also Fig. 3.2(b).

The combination of those three criteria forms the *geometric* affinity measure, as illustrated in Fig.3.2(c) and in the following formula:

$$G(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j) = \sqrt[3]{c_e\left[\boldsymbol{\pi}_i, \boldsymbol{\pi}_j\right] \cdot c_{co}\left[\boldsymbol{\pi}_i, \boldsymbol{\pi}_j\right] \cdot c_{ci}\left[\boldsymbol{\pi}_i, \boldsymbol{\pi}_j\right]} \tag{3.5}$$

Here $G(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j)$ is the geometric affinity between two primitives $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$, where $G(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j) = 1$ indicates the certainty that two primitives are linked, and $G(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j) = 0$ indicates the certainty that they are not. This measure is illustrated in Fig. 3.2(c), for different configurations.

3.2.2 Primitive orientation and switching

As described in section 2.2.4, the orientation property of a primitive is inscribed between $[0, \pi[$ leading to an ambiguity between two equivalent yet distinct interpretations — see Fig. 3.4.

1. a direction of θ
2. a direction of $\theta + \pi$

When comparing two proximate primitives, it is necessary to disambiguate which side (local 'left' or 'right') of each primitives is to be compared with which. As the primitives' orientations define lines, the maximal orientation difference is reached for two orthogonal primitives, yielding an orientation difference of $\frac{\pi}{2}$. As shown in Fig. 3.4, an orientation difference of more than $\frac{\pi}{2}$ implies that the primitives' 'left'

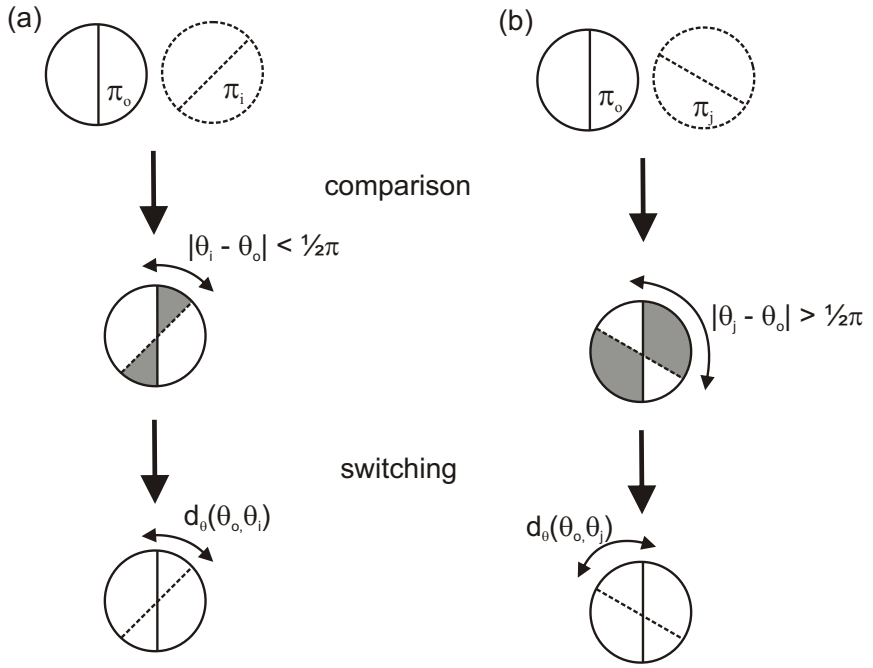


Figure 3.4: When comparing two primitives' colour and phase modalities, it is necessary to decide explicitly which side of the contour is compared with which. Yet, as the orientation of the two primitives is different, the 'left' and 'right' sides of both primitives — as defined in section 2.2.4 — do not fully overlap. Hence, we need to choose which area is compared (in white here) and which is neglected (shown in grey), such that the comparison reflects as much as possible of the overlapping areas. In (a), the difference between orientation is $|\theta_i - \theta_o| < \frac{\pi}{2}$ and the compared area is larger than the neglected one: no switching is required. In (b), the difference $|\theta_j - \theta_o| > \frac{\pi}{2}$, and the neglected area becomes larger than the compared one. In order for the comparison to be meaningful: an orientation switching is required — see also section 2.2.4.

and 'right' areas overlap only marginally. This implies than the colour distance computed between such two 2D–primitives compares only small areas of 2D–primitives's areas, whilst the larger area is ignored. This issue is resolved by applying direction switching as defined in section 2.2.4. Thus, the 'left' side of the first primitive is compared with the 'right' side of the second and thereby the compared areas overlap by more than half.

3.2.3 Modality Consistency

If the geometric constraint offers a suitable estimation of the likelihood of the curve described by the pair of primitives, other modalities allow inferring more about the qualities of the physical contour they

represent. The colour, phase, and optical flow of the primitives further define the properties of the contour. Thus, consistency constraints can also be enforced over those modalities.

Proposition 3.2.4. Similarity Law: *a contour C is more likely if its properties are continuously changing, i.e., $\forall \boldsymbol{\pi}_i, \boldsymbol{\pi}_{i+1} \in C$, we have $d_m(\boldsymbol{\pi}_i, \boldsymbol{\pi}_{i+1}) < \epsilon$ for a small ϵ, where d_m is a modal distance between two primitives.*

Multi–modal affinity: The multi–modal affinity is a weighted sum of the three modalities criteria.

$$M(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j) = 1 - \sum_{m \in \{\omega, \boldsymbol{c}, \boldsymbol{f}\}} w_m d_m(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j) \tag{3.6}$$

Where $d_m(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j)$ stands for the distance between the primitives $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$ in their modality m. Each of the three w_ω, $w_{\boldsymbol{c}}$ and $w_{\boldsymbol{f}}$ is the relative scaling for each modality, with $w_\omega + w_{\boldsymbol{c}} + w_{\boldsymbol{f}} = 1$. As all distance metrics defined in chapter 2 are bounded between $[0, 1]$, $\boldsymbol{M}_{i,j}$ is also bounded between 0 (for dissimilar primitives) and 1 (for identical primitives).

3.2.4 Primitive Affinity

The overall affinity between all pairs of primitives in an image is formalised as the matrix $\boldsymbol{A}$, where $\boldsymbol{A}_{i,j}$ holds the affinity between the primitives $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$. From equations (3.5) and (3.6) we define an overall affinity between the primitives that encompasses all information carried by these primitives:

$$\boldsymbol{A}_{i,j} = \sqrt{\alpha G^2(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j) + (1 - \alpha) M(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j) \cdot G(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j)} \tag{3.7}$$

Here α is the weighting of geometric (*i.e.*, proximity, collinearity, and co–circularity) against multimodal (*i.e.*, phase, colour, and optical flow) information in the affinity. A setting of $\alpha = 1$ implies that only geometric information is used and $\alpha = 0$, that geometric and multimodal information are evenly mixed.

This affinity is also a valid estimate of the likelihood for $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$ to be part of the same contour C. In the following, we will consider that a link $g_{i,j}$ between two primitives exists if $\boldsymbol{A}_{i,j}$ is large enough, with an associated confidence $c\left[g_{i,j}\right] = \boldsymbol{A}_{i,j}$. Fig. 3.5 shows the links extracted, along with the different modal affinities. The links extracted for different thresholds τ_A on the affinity are shown in Fig. 3.6. In the following, we will consider links such that $\boldsymbol{A}_{i,j} > 0.5$.

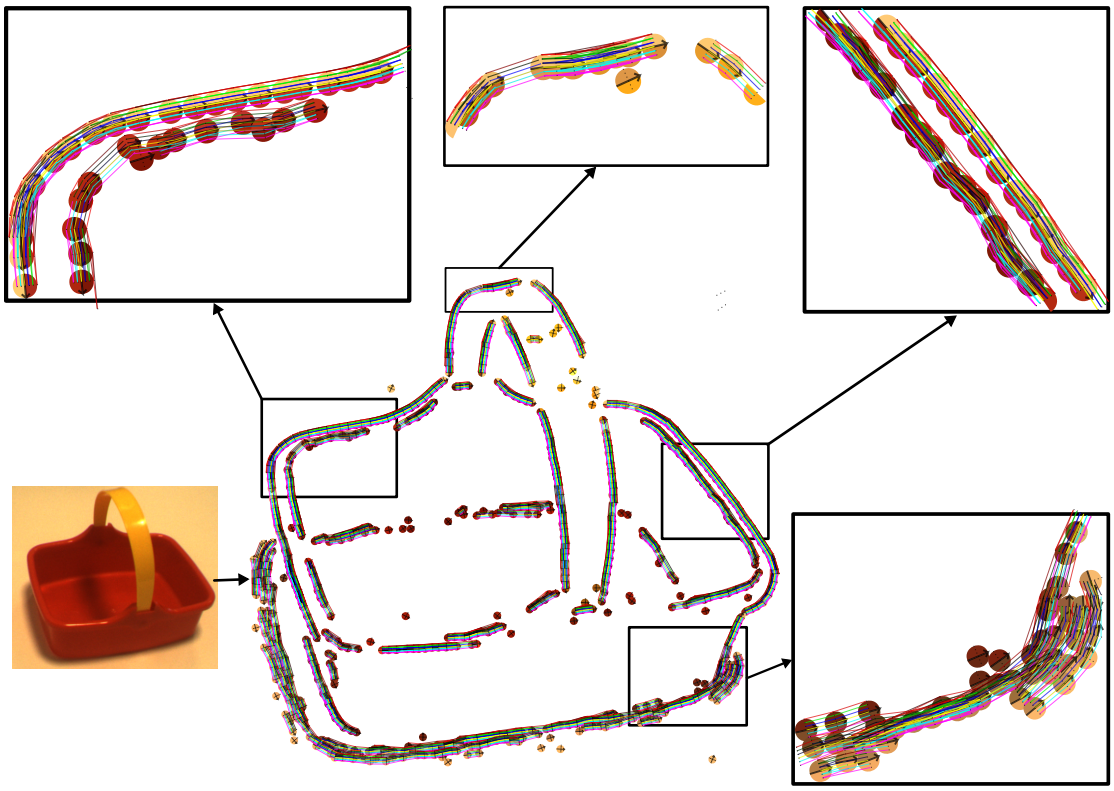
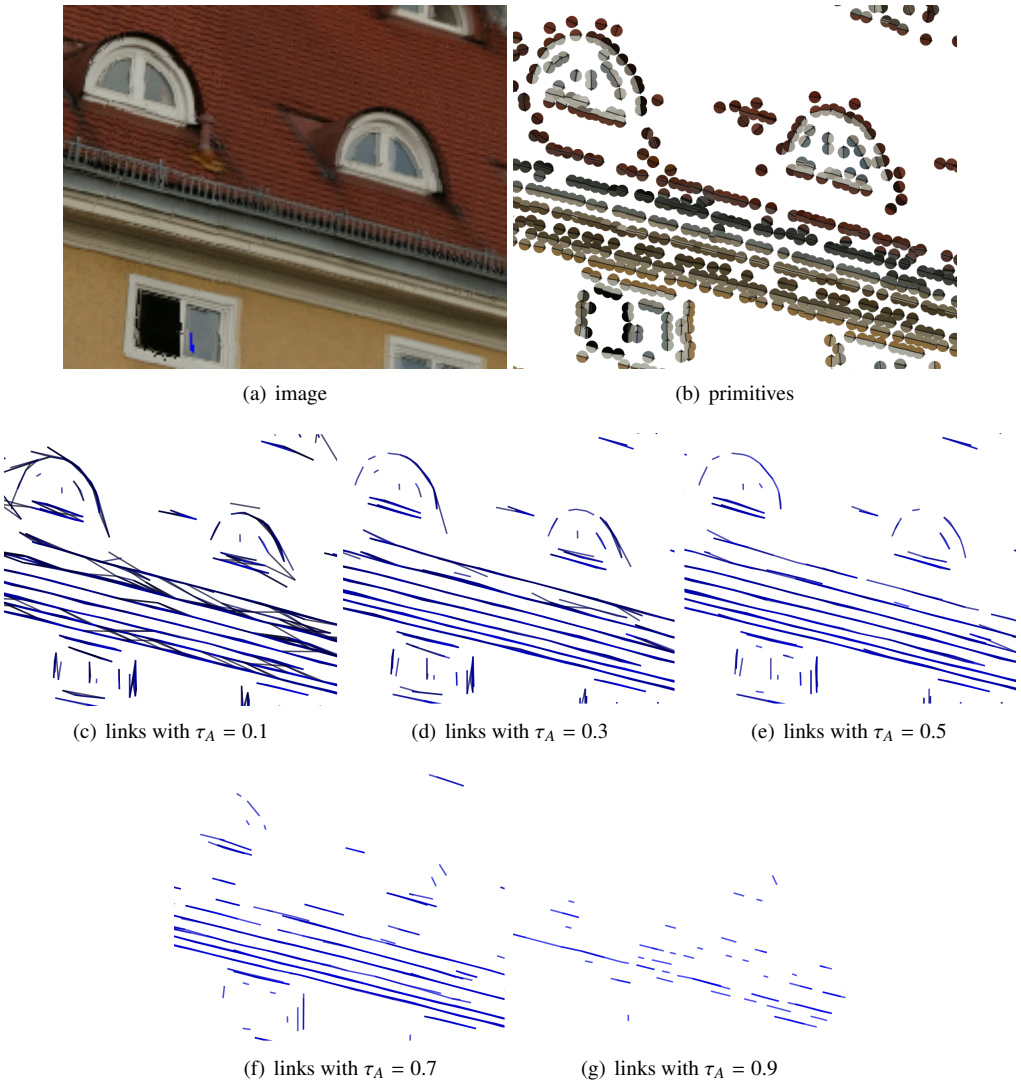


Figure 3.5: Illustration of the affinities between 2D–primitives. In this figure, the 2D–primitives are linked by coloured lines, where a brighter colour stands for a stronger affinity. Red stands for collinearity, Green for phase, Blue for colour, and Yellow for optical flow affinity.



(a) image (b) primitives

(c) links with $\tau_A = 0.1$ (d) links with $\tau_A = 0.3$ (e) links with $\tau_A = 0.5$

(f) links with $\tau_A = 0.7$ (g) links with $\tau_A = 0.9$

Figure 3.6: Illustration of the links extracted for different affinity thresholds — for τ_A values of (c) 0.1, (d) 0.3, (e) 0.5, (f) 0.7, and (g) 0.9 — using a radius $r = 10$. The blue lines represent the links, where more saturated lines stand for higher affinity values.

3.3 Isolated Primitives and Information

One advantage of considering local estimators of image structures (such as the 2D–primitives introduced in chapter 2) is their redundancy along image contours: any such contour will be encoded by a string of primitives in the image representation.

Hence, these contours are *scene* descriptors that are not directly dependent on the sampling, while 2D–primitives are *image* descriptors, and thereby a direct result of the image sampling. Thus, contours are an important step from signal description towards scene representation. Furthermore, we argue that 2D–primitives that are not part of a contour are bits of information that, even though they can be accurate, are purely local and sampling dependent, and thus do not carry useable scene information. Isolated 2D–primitives hold information conflicting with — or simply unsupported by — their context, and that is likely to be the inaccurate or erroneous product of a noisy signal.

Thus, we will differentiate between two classes of 2D–primitives: first, the isolated 2D–primitives, which have low affinity with their neighbours, hence are not part of a contour; second, strings of 2D–primitives sharing consistent information, and describing a scene contour. Note that in this sense, the primitives can be seen as part of a whole, formalised here as contours, which properties descent directly from the quality of this whole (the 3D contour): the 2D–primitives' modalities describe local qualities of the 3D contour they infer. This property is in the line of the *Gestalt* theories.

3.4 Correction of 2D–primitives using interpolation

As we only considered primitives that describe contours, they are expected to show almost always smoothly changing properties (*cf.* proposition 3.2.4). In this section we propose to use the same smoothness assumption to correct 2D–primitives that belong to a contour. If we consider three primitives $\boldsymbol{\pi}_i$, $\boldsymbol{\pi}_j$ and $\boldsymbol{\pi}_k$, that are all part of the same contour $C \subset \mathcal{I}$; furthermore, if we consider that $\boldsymbol{\pi}_i$ lies in between $\boldsymbol{\pi}_j$ and $\boldsymbol{\pi}_k$, then we call $\left(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j, \boldsymbol{\pi}_k\right)$ a *triplet*. Moreover, if we consider two primitives $\boldsymbol{\pi}_j$ and $\boldsymbol{\pi}_k$, it is possible to interpolate between them the contour they describe. Hence, given a triplet $\left(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j, \boldsymbol{\pi}_k\right)$, it is possible to correct the modalities of $\boldsymbol{\pi}_i$ using the contour interpolated between $\boldsymbol{\pi}_j$ and $\boldsymbol{\pi}_k$.

3.4.1 Cubic Hermite spline interpolation

We interpolate the curve between two primitives using Cubic Hermite splines. Those have the advantage of taking the orientation of the control points into consideration, in addition to their localisation (Wikipedia, 2007).

The Hermite interpolation at a location $s \in [0, 1]$ between two primitives $\boldsymbol{\pi}_j$ and $\boldsymbol{\pi}_k$, with positions $\boldsymbol{x}_j$ and $\boldsymbol{x}_k$, and local tangents (defined by the primitives' orientations) of $\boldsymbol{t}_j$ and $\boldsymbol{t}_k$, respectively, is calculated as follows:

$$\widehat{\boldsymbol{x}}(s) = \begin{pmatrix} s^3 & s^2 & s & 1 \end{pmatrix} \cdot \boldsymbol{H} \cdot \begin{pmatrix} \boldsymbol{x}_j \\ \boldsymbol{x}_k \\ \boldsymbol{t}_j \\ \boldsymbol{t}_k \end{pmatrix}, \tag{3.8}$$

Note that $\widehat{\boldsymbol{x}}(s)$ is a vector of the same dimension than $\boldsymbol{\pi}_j$ and $\boldsymbol{\pi}_k$ (2 or 3). The tangent at this point is computed by derivating the polynomial:

$$\widehat{\boldsymbol{t}}(s) = \frac{\partial \widehat{\boldsymbol{x}}(s)}{\partial s} = \begin{pmatrix} 3s^2 & 2s & 1 & 0 \end{pmatrix} \cdot \boldsymbol{H} \cdot \begin{pmatrix} \boldsymbol{x}_j^\top \\ \boldsymbol{x}_k^\top \\ \boldsymbol{t}_j^\top \\ \boldsymbol{t}_k^\top \end{pmatrix}, \tag{3.9}$$

where the position $s = 0$ is $\boldsymbol{x}_j$, the position $s = 1$ is $\boldsymbol{x}_k$, and $\boldsymbol{H}$ is the matrix formulation for the Hermite polynomials:

$$\boldsymbol{H} = \begin{pmatrix} 2 & -2 & 1 & 1 \\ -3 & 3 & -2 & -1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} \tag{3.10}$$

3.4.2 Linear interpolation of modalities

The other modalities are interpolated by assuming that those modalities change linearly between $\boldsymbol{\pi}_j$ and $\boldsymbol{\pi}_k$:

Phase: The phase modality of the primitive interpolated at $s \in [0, 1]$ is computed as follows:

$$\widehat{\omega}(s) = \arctan\left(\frac{(1-s)\sin(\omega_j) + s\sin(\omega_k)}{(1-s)\cos(\omega_j) + s\cos(\omega_k)}\right), \tag{3.11}$$

Colour: The colour components of the primitive interpolated at $s \in [0, 1]$ are computed using the following equation:

$$\widehat{\boldsymbol{c}}(s) = (1-\alpha)\boldsymbol{c}_j + \alpha\boldsymbol{c}_k. \tag{3.12}$$

The optic flow is presently not interpolated.

3.4.3 Primitive correction

For position and colour[2] information, we can correct the *extracted* primitive with the *interpolated* primitive, using the same weighted mean formula that was used for the interpolation, namely:

$$\bar{m} = (1-\alpha)m + \alpha\widehat{m} \tag{3.13}$$

where $m \in \{\boldsymbol{x}, \boldsymbol{c}\}$ is the extracted modality value, $\widehat{m}$ is the interpolated value, $\bar{m}$ is the corrected value, and α is the correction rate (in our case 0.1).

For orientation[3] and phase $m \in \{\theta, \omega\}$, we have:

$$\bar{m} = \arctan\left(\frac{(1-\alpha)\sin(m) + \alpha\sin(\widehat{m})}{(1-\alpha)\cos(m) + \alpha\cos(\widehat{m})}\right). \tag{3.14}$$

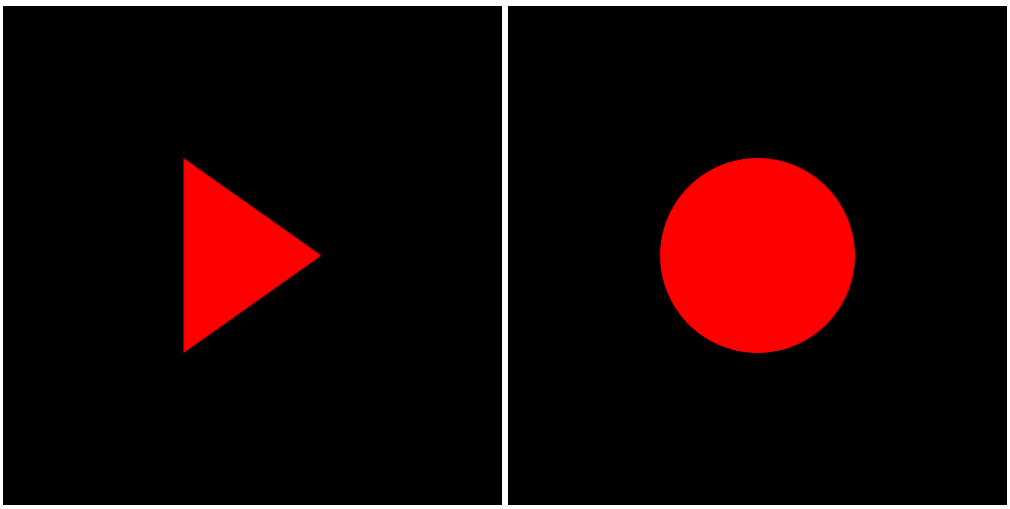
Currently, the optic flow information is not corrected.

3.4.4 Results

We evaluated the performance of this scheme on two simple artificial scenes, illustrated in Fig. 3.7(a), with the primitives extracted drawn in Fig. 3.7(b). Fig. 3.7(c), (d), (e), show the results of 10 iterations of the correction process for the localisation, orientation, and phase of the 2D–primitives, in the triangle (full line), and the circle (dashed line) scenarios. The horizontal axis shows the number of iterations of

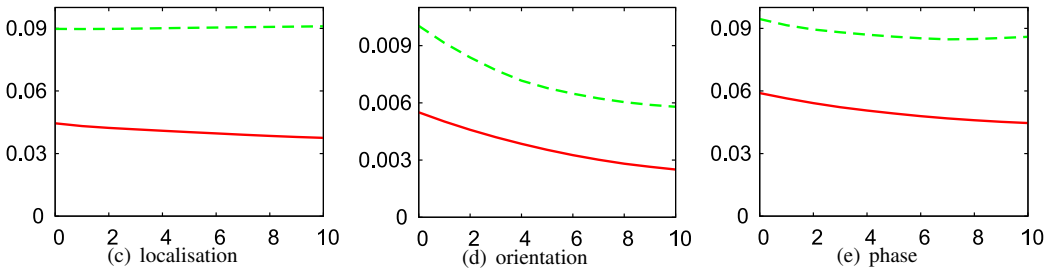
[2] Note that by colour we mean *all* components of the colour information, on *both* sides of the primitive.

[3] Note that if $|\widehat{\theta} - \theta| \leq \frac{\pi}{2}$, we need to switch the primitive's direction interpretation as described in section 2.2.4, before correcting orientation, colour, or phase.



(a) images

(b) primitives



(c) localisation (d) orientation (e) phase

Figure 3.7: (a) Images used for the quantification of the 2D–primitives correction process. (b) Primitives extracted for each image. (c,d,e) Accuracy of the 2D–primitive (c) localisation, (d) orientation and (e) phase after several iterations of the correction process, for the triangle (full line) and circle (dashed line) scenarios. The horizontal axis shows the number of iterations of the correction process and the vertical axis shows the error for (c) in pixels, and for (d) and (e) in radians.

the correction process and the vertical axis shows the mean error for the 2D–primitives. Note that the error is measured in pixels for the localisation and in radians for the orientation and the phase.

First, before any correction, note that sub–pixel accuracy is lower for the circle scene, due to the curvature of the contour: because primitives are local *line* descriptors, the quality of the curved contours' description they provide decreases with local curvature. Conversely, because the primitives' sub–pixel localisation assumes a linear model, it performs better with linear structures. Nonetheless, the accuracy is high in both cases: less than one tenth of a pixel for the localisation and less than one hundredth of a radian for the orientation.

When looking at the localisation results, in Fig. 3.7, we note that interpolation leads to mixed results depending on the modality: We see a distinct improvement of the localisation for the triangle scene but not for the circle. This is due to two sources: In the case of the circle, Hermite interpolation is used. This interpolation makes use of the orientation of the control points in addition to their position. Hence, the interpolated curve is sensitive to errors in orientation. Furthermore, although Hermite polynomials are an efficient model for describing general curves, they do not allow a perfect interpolation of an arc. Thus large curvatures lead to a reduced quality of the interpolations. Nonetheless, the accuracy obtained by interpolation is sufficient for improving the sub–pixel localisation.

Concerning orientation we see a clear improvement of ~ 0.003 radians for both scenarios ($\sim 50\%$ and $\sim 30\%$ for the triangle and circle scenarios).

Phase shows a clear if slight improvement in both cases. The triangle scenario sees an improvement of ~ 0.015 ($\sim 25\%$), whereas the circle scenario sees an improvement of ~ 0.01 ($\sim 11\%$).

Those results show that the 2D–primitives correction, using the contours described in this chapter, lead to an improvement of the primitives' accuracy. The localisation improvement is slight, but the original sub–pixel accuracy is already high. On the other hand, we obtain a bigger improvement when correcting other modalities — in this case orientation and phase.

3.5 Conclusion

In this chapter we have proposed a simple definition of primitive affinity in the image domain, based on the *Gestalt* principles of proximity, good continuation, and similarity. This affinity measure provides us with a simple way to form contours from local 2D–primitives.

We evaluated that the inferred contours have a high likelihood to be conserved if observed from another viewpoint (in the case of this chapter, the view given by another camera, in a fronto–parallel set–up). This means that these contours, more than merely describing the image, are actual descriptions of the contours in the scene. In chapter 5 we will extend this definition to the 3D domain using stereopsis.

In the last part we proposed a method to interpolate contours between the primitives describing it, making use of Hermite (or linear in the case of perfectly collinear primitives) interpolation. Using this interpolation as a predictor, we proposed to correct the extracted 2D–primitives using the primitives predicted at this location by pairs of neighbour primitives (neighbour meaning here a proximate primitive that has a direct with the corrected one). This interpolation significantly improves the accuracy of the primitives' modalities This correction is a local process and a first example of the inter–process feedback mechanisms advocated in the introduction.

Part II

Stereopsis and 3D reconstruction $\mathcal{S}$

Chapter 4

Using Primitives for Stereo–reconstruction

One sees great things from the valley,
only small things from the peak.
- G. K. Chesterton

The 2D–primitive based image representation described in section 2 provides a good description of images in terms of low–level symbolic entities. Yet, the purpose of vision is not to merely infer knowledge about an image, but rather about the 3D scene that produced it. For this, the reconstruction of depth information, and thus of full 3D shape of the scene, is essential.

In this chapter we will focus on the depth cue that was most successfully applied to the computer vision problem, namely stereo–reconstruction. The premise is as follows: Given two (or more) calibrated cameras viewing one scene, if we can identify corresponding points in each image, it is possible to reconstruct the corresponding 3D–point. Here *calibrated* means that the cameras' projective parameters are known — see appendix C. We refer to (Faugeras, 1993; Hartley and Zisserman, 2000) for a comprehensive review of the geometric problems involved.

We propose to implement stereo, using the primitive based framework described in the previous chapters. Marr (1982) suggested that edge features are a good base for stereo, and Grimson (1993) discussed psychophysical experiments showing that human stereo vision is blind to constant gradient depth. This

strongly advocates for the existence of an edge based depth estimation mechanism — although it is likely that the human visual system adopts a mixed strategy (Mayhew and Frisby, 1981).

Consider a stereo–pair of calibrated cameras, labelled 'left' and 'right' for convenience (although any other kind of physical arrangement of the cameras is possible); from the produced pair of images I^r, I^l we extract two sets of 2D–primitives $\boldsymbol{\mathcal{I}}^l$ and $\boldsymbol{\mathcal{I}}^r$, as described in chapter 2, that are hereafter called *image representations*. Our intent in this chapter is to match 2D–primitives between those two image representations, and to infer from such correspondences the spatial equivalent of 2D–primitives henceforth called 3D–primitives.

A pair of corresponding 2D–primitives provides considerably more information than, *e.g.*, two corresponding points: First, we know that corresponding primitives are projections of the *same* 3D feature. Hence they share similar (up to the projective distortion induced by the viewpoint difference) properties in terms of orientation, colour, phase, and optic flow. Second, the multi–modal information held by both 2D–primitives can in turn be used to infer equivalent spatial information about the scene. Therefore a 3D–primitive is more than a position in space.

Considering a stereo–pair of images I^l and I^r of a given scene, and their respective image representations $\boldsymbol{\mathcal{I}}^l$ and $\boldsymbol{\mathcal{I}}^r$, if $\boldsymbol{\pi}^l \in \boldsymbol{\mathcal{I}}^l$ and $\boldsymbol{\pi}^r \in \boldsymbol{\mathcal{I}}^r$ are two corresponding primitives, we define the *reconstruction* of a 3D–primitive from a stereo–pair of 2D–primitives as the following relation:

$$\mathcal{R} : (\boldsymbol{\pi}^l, \boldsymbol{\pi}^r) \longrightarrow \boldsymbol{\Pi} \tag{4.1}$$

In this formula $\boldsymbol{\Pi}$ encodes a 3D entity, spatial equivalent of a stereo–pair of primitives. Ideally, we want $\mathcal{R}$ such as the reverse operation

$$\mathcal{P} : \boldsymbol{\Pi} \longrightarrow \left(\widehat{\boldsymbol{\pi}}^l, \widehat{\boldsymbol{\pi}}^r\right) \tag{4.2}$$

is feasible, with $\widehat{\boldsymbol{\pi}}^l$ and $\widehat{\boldsymbol{\pi}}^r$ holding the same information than respectively $\boldsymbol{\pi}^l$ and $\boldsymbol{\pi}^r$. We refer to this operation as the *reprojection* of a 3D–primitive onto an image plane.

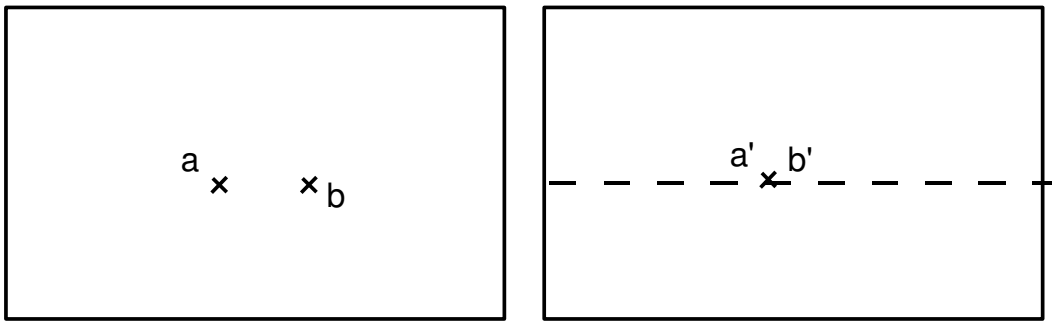
Stereopsis faces one of the most difficult problems of artificial vision — namely: "how to match the primitives extracted from the first image with those extracted from the second one." This *correspondence problem* is difficult because of the re–occurrence of similar structures in natural images. It implies that, in an image, several primitives will have the exact same properties (think, for example, of a carpet with

repetitive patterns). The problem is further complicated by the fact that two manifestations of the same spatial structure viewed from different perspectives can be quite different.

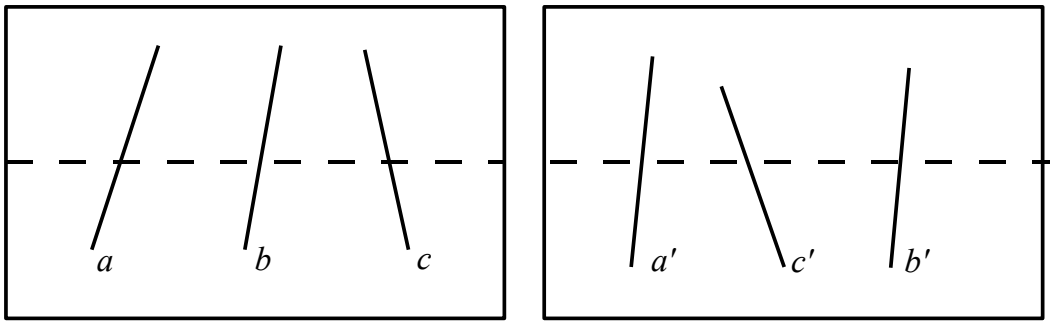
Although there exists no general, local solution to this problem, and the impossibility of such a solution has been demonstrated by Burns et al. (1992), different local matching algorithms were proposed, and achieved some measure of success. Mayhew and Frisby (1981); Grimson (1985) used the sign of the zero–crossings and their orientation for local matching; this is similar to our use of orientation and phase distances (Kovesi, 1999). Ayache and Faverjon (1987) used the length and orientation of line segments. The length of the line segment is not a robust criterion, because 1) it depends on the perspective under which the line segment is observed, as observed by (Ogale and Aloimonos, 2006), and 2) under partial occlusions the visible length of the segment can vary. The orientation is reliable for small baselines and for distant objects, but, as explained in chapter 2, the orientation distortion between two views increase drastically for larger baselines or closer objects. Lee and Leou (1994) used the orientation similarity and the overlapping factor of two line segments — stated as follows: "assuming horizontal epipolar geometry, how much do two segments vertically overlap ?". They proposed a global matching approach, that forms a relational graph between the line segments and uses dynamic programming to find the maximum weighted path through this graph. Kim and Bovik (1988) tried to match line segments' end–points. This approach performance is very dependent on the reliability with which these end–points are extracted, located, and matched; it fails in the case of partially occluded contours. Schmid and Zisserman (1997, 2000) proposed to compute the normalised cross–correlation between the pixels surrounding lines (or curves). In the present work we make use of the multi–modal information carried by the 2D–primitives to design a robust matching criterion. The use of multi–modal, symbolic information provides some measure of robustness to projective distortion. This is described in details in section 4.2.

Because matching local image patches across viewpoints is an unsolvable problem, it is common to use additional global constraints in order to simplify the matching problem:

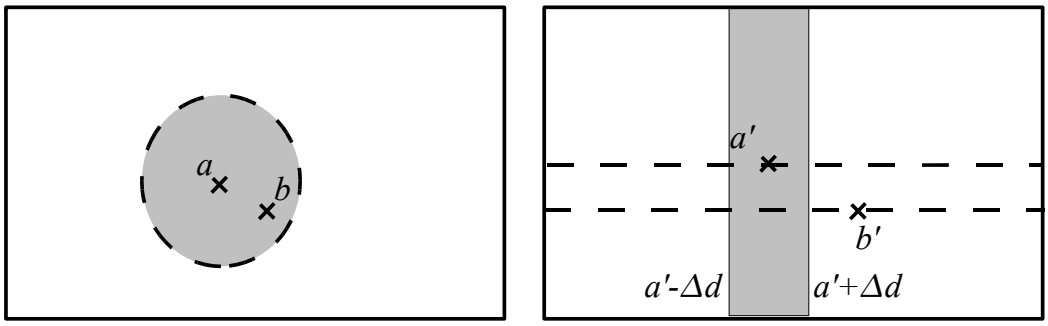
Epipolar constraint: The epipolar constraint states that the correspondence of a point in the left image must lie on a line that is the projection on the right image of the optical ray back–projected by this point and containing all its possible origins in space — see, *e.g.*, (Faugeras, 1993; Hartley and Zisserman, 2000). This constraint allows to reduce the correspondence search to a one–dimensional



(a) Uniqueness



(b) Ordering



(c) Gradient

Figure 4.1: Illustration of commonplace global constraints for stereo–matching. In these cases we assume horizontal epipolar lines, and no vergence. The dashed line shows the epipolar line in each case. (a) Uniqueness constraint: if a and b are two points in the left image, then their two correspondences a' and b' must be distinct. (b) Ordering constraint: if we consider 3 lines a, b and c, crossing the horizontal line in this order, their three correspondences need to cross the horizontal line in the same order: a', b' then c'. (d) Gradient constraint: if we consider two proximate points a and b in the left image, their disparity must be similar, and thus a' and b' must also be proximate.

manifold, called the *epipolar line*. This epipolar line is displayed as the dashed lines in the examples of Fig.4.1.

Uniqueness constraint: The uniqueness constraint states that one point in the left image can at most correspond to one single point in the right image, and conversely. This constraint is generally false when considering contour features: a scene contour is sampled by a different number of pixels in each image, and can therefore lead to a different number of 2D–primitives — see, *e.g.*, (Ayache and Faverjon, 1987). This forbids the case illustrated in Fig. 4.1(a).

Ordering constraint: The ordering constraint states that the horizontal ordering of features in the left image should be preserved in the right image — see (Baker and Binford, 1981; Ohta and Kanade, 1985) This effectively forbids the case illustrated in 4.1(b).

Gradient constraint: The gradient constraint (also called continuity constraint) is based on the following statement: "Matter is cohesive, it is separated into objects." (Marr and Poggio, 1976); it enforces that proximate points in the left image should have a similar disparity, and therefore their correspondences should also be proximate — see, *e.g.*, (Ayache and Faverjon, 1987; Kim and Bovik, 1988). This forbids the case shown in Fig. 4.1(c).

Figural continuity: The figural continuity was proposed by Mayhew and Frisby (1981), and suggests that the primal sketch should be conserved across stereo.

These global constraints can be enforced using various forms of optimisation over the whole image: *e.g.*, dynamic programming (Lee and Leou, 1994), graph operations like maximal clique (Horaud and Skordas, 1989), belief propagation (Sun et al., 2002), non–linear diffusion (Scharstein and Szeliski, 1998).

In the following we will not use any global optimisation, and only use the epipolar constraint, and this for two reasons: First, global constraints (or global optimisation processes) enforce a certain bias in the interpretation of the scene. Although this bias is acceptable in the general case, we believe that it is preferable to postpone such global operations to a later stage, when visual information is interpreted in terms of world knowledge, and contextual information is available. Second, the aim of this work is to investigate how local interactions and inter–processes feedback mechanisms can provide disambiguation already at a local level. In the following we will present a classical, local stereo–matching algorithm

making use of the primitive–based image representation we presented in chapter 2 and of the epipolar constraint.

First, section 4.1 presents the implementation of the epipolar constraint used in this work, and the finding of potential correspondences. Second, section 4.2, describes the multi–modal confidence rating of those potential correspondences. This can be followed by a standard winner–take–all mechanism. The performance of such a scheme is evaluated in section 4.3. The reconstruction of points (Liu et al., 2005) and lines (Wolff, 1989) is well known, and we address the reconstruction of a 3D–primitive from a stereo–pair of corresponding 2D–primitives in section 4.4. Finally, we will briefly present the re–projection process of a 3D–primitive onto an image plane in section 4.5. This will prove useful in later chapters for the implementations of feedback loops between 2D and 3D entities.

4.1 Finding putative matches for a primitive

If the computation of the depth of a 3D point, from its projection onto both image planes, is well known and understood (see, *e.g.*, (Faugeras, 1993)), to identify the image projections of one 3D point under different viewpoints, the so–called *correspondence problem*, is an open problem (see (Burns et al., 1992)). The problem we face can be rephrased as follows: "Given a 2D–primitive in the first image, which 2D–primitive in the second image is the projection of the same 3D feature ?".

In this section we will propose a simple algorithm to select plausible pairs of 2D–primitives from the two images, called henceforth *putative correspondences*. We will use a combination of geometric constraints (the so–called *epipolar constraint* described in appendix C) and of a similarity measure between the two primitives (described in the next section).

When applying the epipolar constraint to 2D–primitives, one important consideration is that a 2D–primitive is not located at a single point, but represents a whole image patch. Consequently, we need to loosen slightly the classical epipolar constraint, insofar that we will consider as putative correspondence of a 2D–primitive in the first image any 2D–primitive in the second image that lie nearby the epipolar line. Accordingly, the position of the centre $\boldsymbol{x}^r_j$ of a 2D–primitive in the right image is estimated relatively to the epipolar line, in terms of its tangential and normal components — see Fig. 4.2. The *normal* component is the Euclidian distance from the centre of the receptive field to the epipolar line $d_{\text{norm}}(\boldsymbol{\pi}^r, \boldsymbol{\pi}^l) = d(\boldsymbol{x}^r, l^r_{\boldsymbol{x}^l})$ and the *tangential* component is the distance between the two points $d_{\tan}(\boldsymbol{\pi}^r, \boldsymbol{\pi}^l) = d(\boldsymbol{x}^r_j, \boldsymbol{x}^r_\infty)$, where $\boldsymbol{x}^r_\infty$

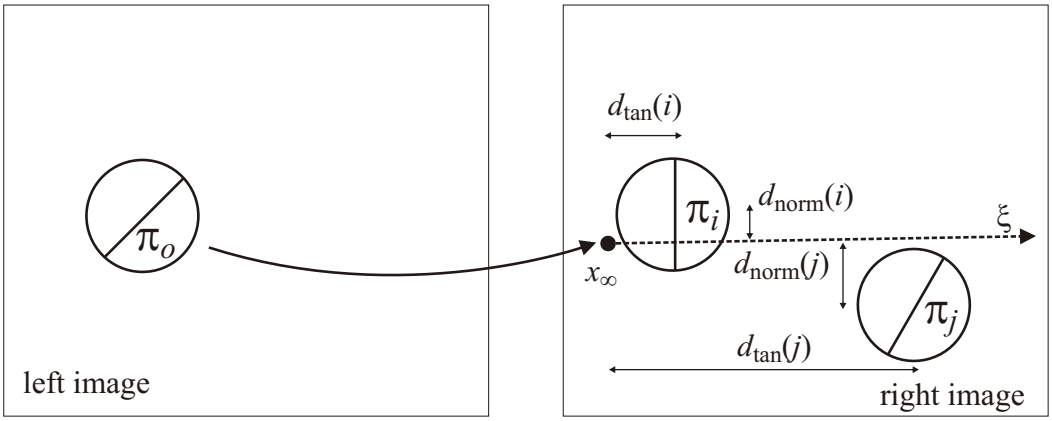


Figure 4.2: This figure shows how the epipolar constraint is enforced during the stereo–matching. First, because the primitives are extracted in a sparse way, it is unlikely to find a primitive on the right image that lie exactly on the epipolar line; therefore we consider as putative correspondences primitives that lie within a certain distance from the epipolar line We call this distance the *normal disparity*, and set it to one and half times the primitive's size. The smaller is this value, the more accurate is the 3D–reconstruction. Second, a left image 2D–primitive's correspondence in the right image for an object infinitely far is called *projection at infinity*. The distance between this point and the putative correspondence is called the tangential disparity. Larger tangential disparities stands for closer 3D entities.

is the re–projection at infinity of $\boldsymbol{x}^l$ — see appendix C. If the epipolar line crosses the image patch of the second 2D–primitive far from its centre the 3D origin of the two 2D–primitives only overlaps marginally; thus, they only share a small part of their 3D information. Hence, the normal component could also be used as a measure of *inaccuracy* (and will be used to enforce this loose epipolar constraint), while the tangential is the ideal disparity: the disparity assuming that the centre of the matched primitive would lie precisely on the epipolar line. Some imprecision is unavoidable at this stage, due to the sparseness of the representation; we will show that it can be corrected during reconstruction — see section 4.4. If the distance $d_{\text{norm}}(\boldsymbol{\pi}^r_j, \boldsymbol{\pi}^l_i) < \varepsilon\lambda_j$, (with $\varepsilon = 1.5$, and λ is the size of the primitive $\boldsymbol{\pi}^r_j$) then the 2D–primitive $\boldsymbol{\pi}^r_j$, located at $\boldsymbol{x}^r_j$, is considered as a putative correspondence of $\boldsymbol{\pi}^l_i$. Hence, any 2D–primitive $\boldsymbol{\pi}^l_i$ in the left image has a set of competing (as we know that *at most one* correspondence can be correct), putative correspondences $\{\boldsymbol{\pi}^r_j\}$ in the right image, leading to the inference of different 3D structures. The corresponding stereo–hypotheses are written $\{s_{i\to j}\}$.

Note that, by keeping record of all hypotheses rather than selecting straight away one candidate, we keep the possibility to use contextual knowledge available at later processing stages to revise this early decision. This will be developed in the following chapters.

4.2 Evaluation of the putative correspondences: multi–modal similarity

Apart from cases of occlusion — where there is no 2D–primitive $\boldsymbol{\pi}_j^r$ in the right image such as $s_{i\to j}$ is the correct stereo — the problem becomes to select the correct correspondence out of those competing hypotheses. In order to distinguish the 2D–primitives, we will associate to each hypothesis a confidence $c\left[s_{i\to j}\right]$ based on the similarity between the corresponding 2D–primitives in the left and right image.

The similarity function used is akin to the one defined in section 3.2.3 in the perceptual grouping context. The notable difference is that, because neither collinearity nor co–circularity rules apply between stereo–pairs of 2D–primitives, the geometric constraint reduces to a difference in direction between the pair of primitives — illustrated in Fig. 4.3. Note that, in the general case, the *correct* correspondence is expected to be somewhat different from the original 2D–primitive, due to the difference in viewpoints, the sparseness of the representation, and noise. Similarity between stereo–pairs of 2D–primitives is not an exact mapping of the correctness of this correspondence assumption; however, we will show that it is an efficient criterion for identifying the correct correspondence over spurious ones.

4.2.1 Switching in the stereo case

In a similar manner than in the grouping context, the orientation–direction ambiguity needs to be resolved in order to compare two 2D–primitives over stereo. In this case, the constraint one can apply on the interpretation of the two 2D–primitives is a three–dimensional one. The two different cases are illustrated in Fig. 4.3: if the orientation of both 2D–primitives point on the *same* side of the epipolar line defined by the left 2D–primitive, no switching is required. On the other hand, if the orientations points to *different* sides of this line, the two interpretations are incompatible; hence, the second 2D–primitive is switched — as defined in section 2.2.4.

4.2.2 Geometric constraint in the stereo case

We stated before that the similarity measure used for stereo–matching is similar to the one used for the perceptual grouping of 2D–primitives. The main difference lies in the interpretation of the geometric relationship between two 2D–primitives: in the grouping context, we used a geometric constraint that

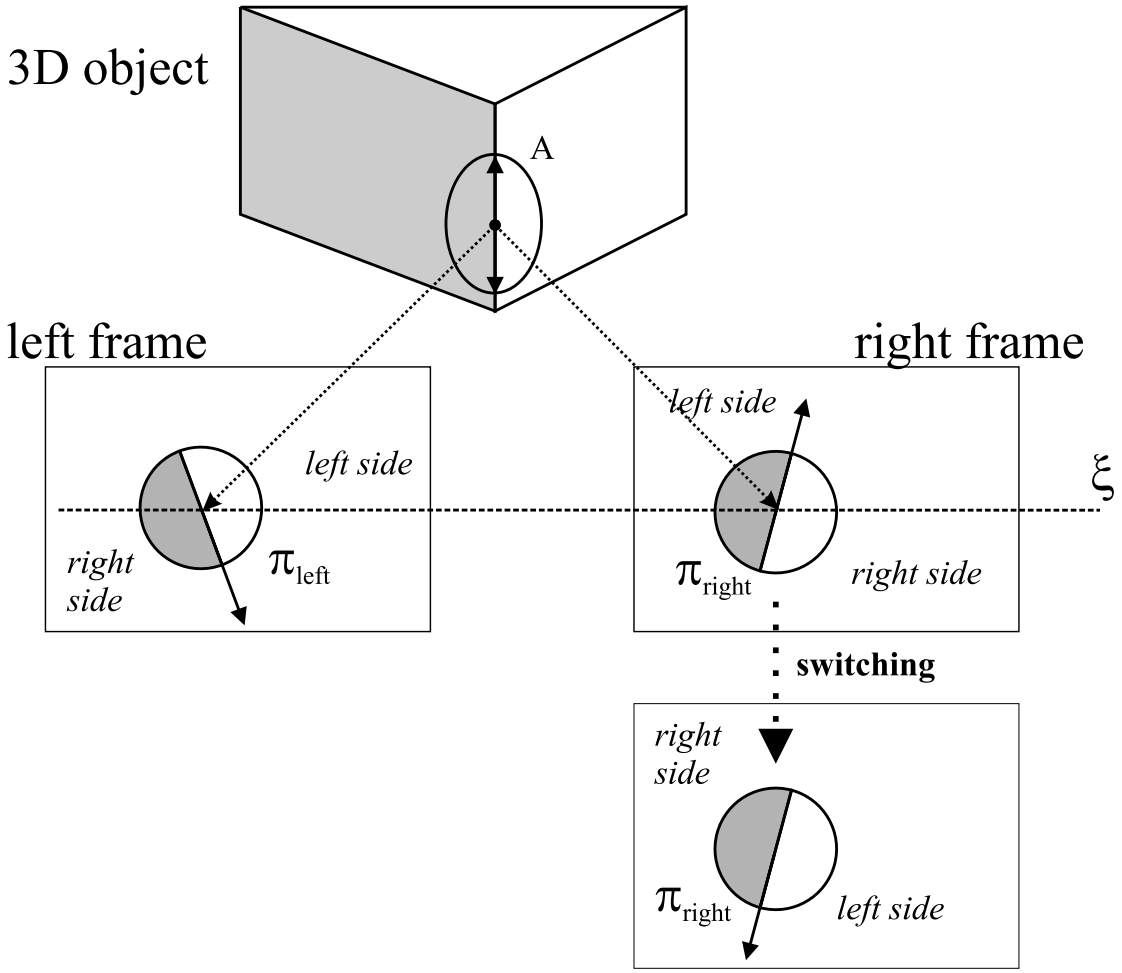


Figure 4.3: Illustration of the 3D consistency constraint applied on the interpretation of the orientation of primitives in the stereo case. The orientation of the local contour is arbitrary, but has to be chosen consistently between the left and right primitives to ensure that each side of the contour is unambiguously identified. Here we see that if the orientation of the primitive is above (respectively below) the epipolar line in the left image, it should also be above (below) in the right image. If not, the two 2D–primitives' interpretations are inconsistent: the right 2D–primitive need to be *switched* — as explained in section 2.2.4.

assessed the relative position and orientation of the 2D–primitives in terms of proximity, collinearity, and co–circularity. In the stereo context, there is no equivalent to the good continuation rule; therefore we will simply consider the 2D–primitives' orientation difference. Because the two cameras view the scene from a different perspective, the orientation of two corresponding 2D–primitives is expected to be somewhat different. Nevertheless, using orientation similarity for stereo–matching yields a performance significantly above chance, as shown in section 4.3.

In the stereo case the geometric constraint is reduced to the normalised angular distance between the 2D–primitives' orientations:

$$c_\theta\left[s_{i\to j}\right] = 1 - \arctan\left(\frac{\sin(|\theta_j - \theta_i|)}{\cos(|\theta_j - \theta_i|)}\right) \tag{4.3}$$

4.2.3 Multi–modal stereo confidence

The three other modal metrics (*i.e.*, phase, colour and optical flow) used in the grouping context (see section 2.3) are used in the same way in the stereo–matching context. Note that, before computing the similarity, the orientation consistency needs to be ensured — as described in section 4.2.1.

We define a multi–modal similarity between a stereo–pair of 2D–primitives as a weighted combination of the individual modal metrics, as follows:

$$c\left[s_{i\to j}\right] = \boldsymbol{w} \cdot \begin{pmatrix} c_\theta\left[s_{i\to j}\right] \\ c_\omega\left[s_{i\to j}\right] \\ c_c\left[s_{i\to j}\right] \\ c_f\left[s_{i\to j}\right] \end{pmatrix} \tag{4.4}$$

with $\boldsymbol{w} = (w_\theta, w_\omega, w_c, w_f)$ the weighting of the modalities distances between the two 2D–primitives so that $w_\theta, w_\omega, w_c, w_f \in [0, 1]$ and $w_\theta + w_\omega + w_c + w_f = 1$.

4.2.4 Limits of the epipolar constraint

If we consider a contour which orientation in the right image is nearly parallel to the epipolar line (*e.g.*, an horizontal line), then all 2D–primitives in the right image along such a contour are putative correspondences. Furthermore, as they are all extracted from the same contour they hold very similar properties. This makes it nigh impossible (or at best, unreliable) to identify the true correspondence amongst them

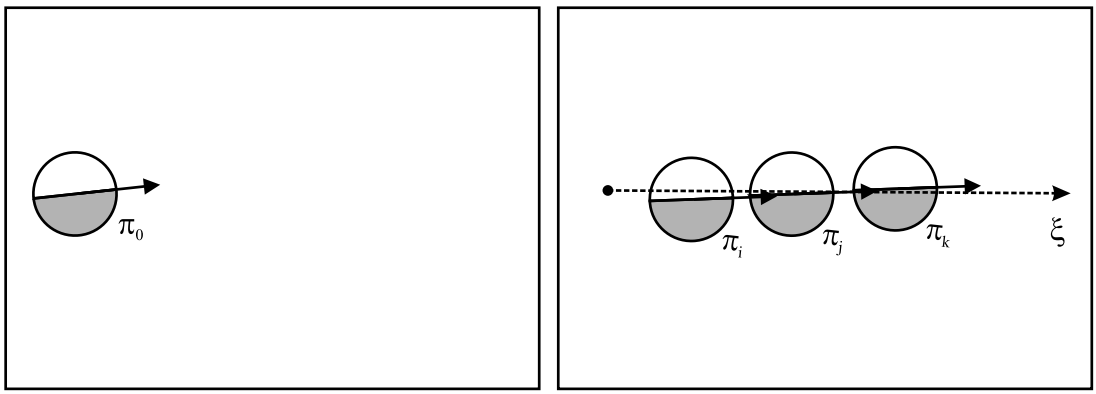


Figure 4.4: Limits of the epipolar constraint. Consider a primitive $\boldsymbol{\pi}_0$, in the left image, and its epipolar line ξ in the second image. Three 2D–primitives $\boldsymbol{\pi}_i, \boldsymbol{\pi}_j, \boldsymbol{\pi}_k$ are crossed by ξ, designating them all as putative correspondences of $\boldsymbol{\pi}_0$. Those three primitives are all manifestations, in the right image, of the *same* contour of the scene; consequently, they are similar in all modalities, and the multi–modal constraint do not allows to choose reliably between those three candidates. Hence, when an image contour's orientation is similar to epipolar line's orientation, several 2D–primitives along this contour will be candidates and the correspondence problem cannot be reliably solved by local means.

— illustrated in Fig. 4.4. This is the worst case of stereo–ambiguity that can be found in general scenes, and it is unfortunately common (indeed horizontal structures are fairly common in natural scenes).

This ambiguity cannot be overcome using solely local information. One common strategy is to enforce an *ordering constraint* on the stereo–correspondences. This requires to identify accurate endpoints on those segments, and is unstable in the event of occlusion. Furthermore, because this study focuses on local edge primitives, the endpoints are not directly available to us, and such an approach is unpractical. Effectively, we chose to disregard primitives with an orientation differing with the one of the epipolar line by less than 10 degrees.

4.3 Quantification of the multimodal stereo

In their survey of dense two frame stereo–matching algorithms, Scharstein and Szeliski (2002) used the root mean square (RMS) error of the disparity error and the percentage of pixels with an absolute disparity error of more than 0.5 pixels in order to evaluate the quality of the stereo–match found. The RMS error is known to be sensitive to outliers. This was not critical in (Scharstein and Szeliski, 2002) because their experiments only considered a disparity range of 20 pixels, but it makes this measure unsuitable in

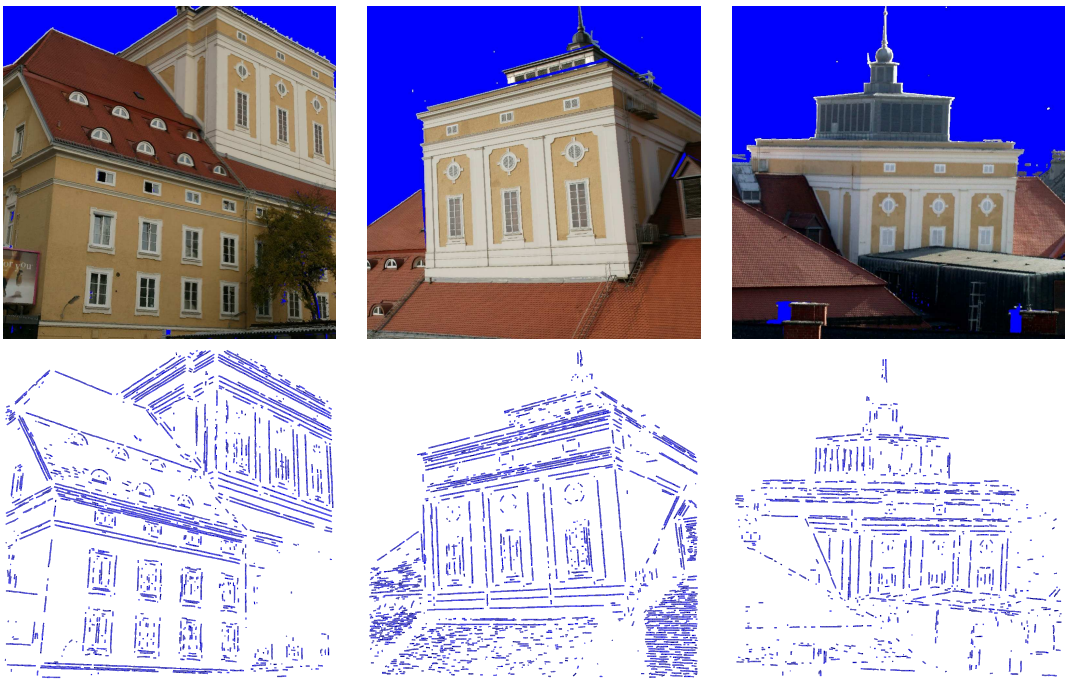


Figure 4.5: Three sequences with ground truth used for the stereo–matching evaluation.

the general case, where the disparity is unconstrained. We therefore evaluated the performance of our stereo–matching algorithms by considering its reliability: how many of the matches found were correct and how many were erroneous — similar to the second measure proposed by (Scharstein and Szeliski, 2002). For this purpose, we defined a stereo–match to be correct if its disparity error is smaller than the 2D–primitive's size λ; hence, there could not be another more accurate correspondence in the image. Note that the imprecision of the disparity can be either due to the sparsity of the 2D–primitives' sampling, or due to its inaccuracy. In the first case, this imprecision is corrected during the reconstruction process; in the second case, the inaccuracy of the 2D–primitives directly impacts the accuracy of the reconstructed 3D–primitives.

For the purpose of quantification, we use three different calibrated stereo sequences, illustrated in figure 4.5. For the purpose of quantification we use outdoor high resolution images of buildings, with associated depth ground truth recorded using a range scanner. These images were provided by the company Riegl. The right image is interpolated from the left image using the depth ground truth, leading to high quality semi–artificial images.

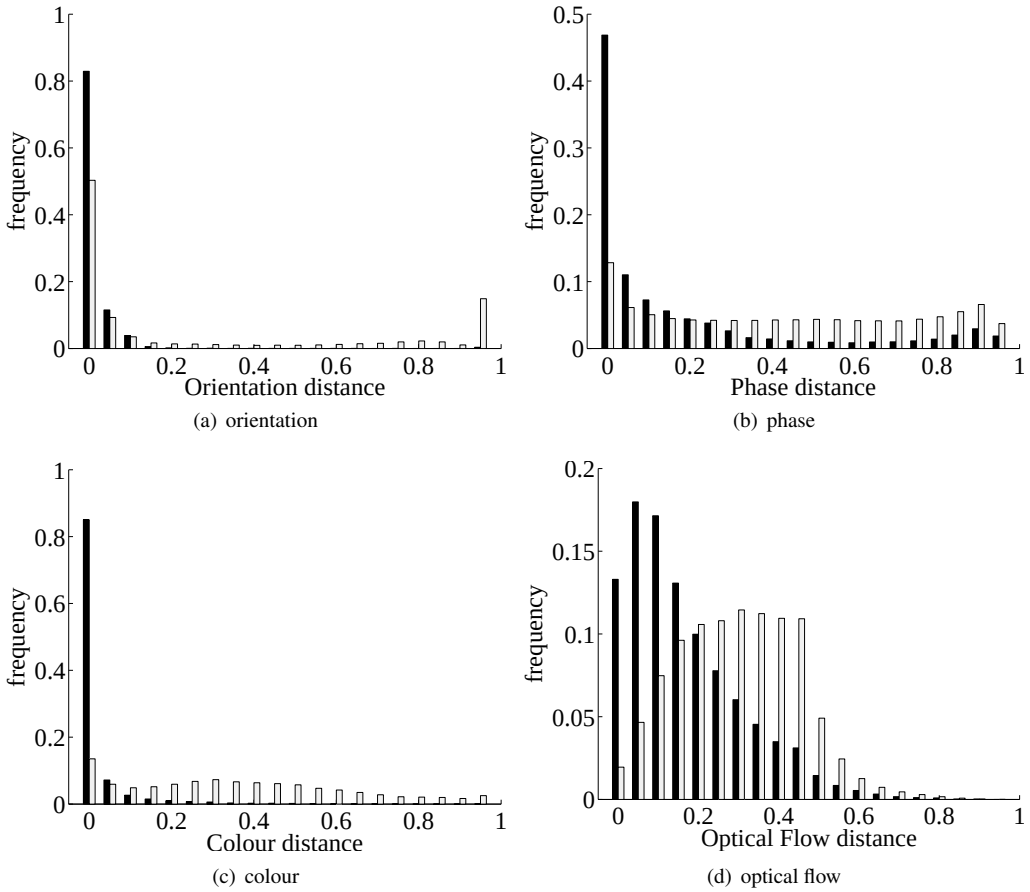


Figure 4.6: Histograms of the distances in different modalities (0 for identity and 1 for dissimilar items), for correct (black bars) and false (white bars) correspondences. The vertical axis shows this distance value's frequency of occurrence, between 0 and 1. These results were obtained from 10 frames of the sequences shown in Fig. 4.5A, B, and C.

4.3.1 Performance of different modalities

Fig. 4.6 shows histograms of the modal distances between pairs of 2D–primitives that satisfy the epipolar constraint. All histograms show a separation between the distributions of correct (black) and false (white) correspondences. In the phase (in 4.6(b)) and colour (in 4.6(c)) histograms, correct correspondences form a sharp peak for a distance of zero; false correspondences show an even distribution for all distance values between $[0, 1]$. In Fig. 4.6(a), the large peak at zero distance for false correspondences is explained by the presence of parallel structures in the image: if one draws an horizontal line in an image, this line will cross numerous parallel contours with very similar local orientations. The optical flow distribution shown

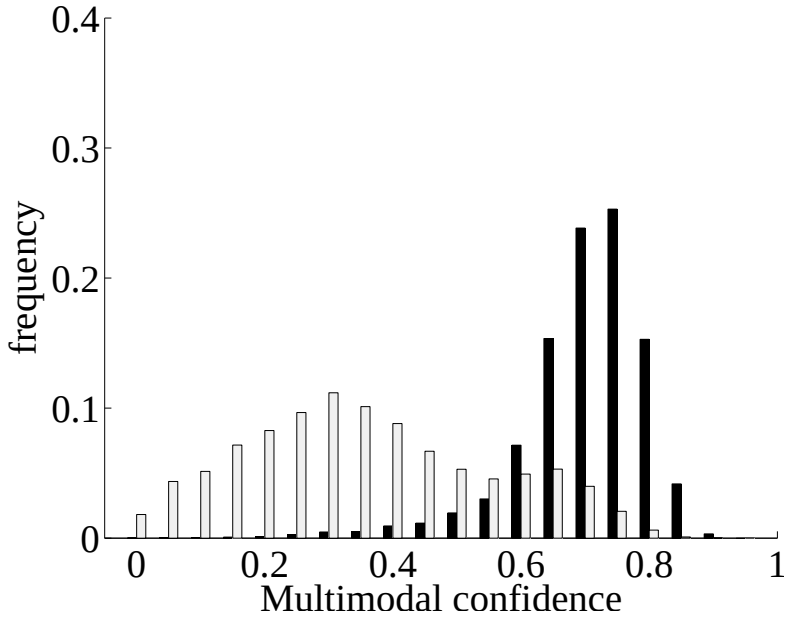


Figure 4.7: Histograms of multimodal similarity between two putative stereo–correspondences (1 for identity and 0 for dissimilar items), for correct (black bars) and false (white bars) correspondences. The vertical axis shows this value's frequency of occurrence, between 0 and 1. This is measured for an optimal weighting. These results were obtained from 10 frames of the sequences shown in Fig. 4.5A, B, and C.

in 4.6(d) feature a Gaussian–like distribution centred at a distance of 0.1 for the correct correspondences, with a long tail until 0.6. This is due to the noisiness of optical flow data. The false correspondences also show a clearly peaked distribution, centred at a distance of 0.3. In spite of this large overlap, optical flow allows some discrimination between correct and erroneous candidates. The histograms in Fig. 4.3.1 show the distribution of the overall multimodal similarity. There, the discrimination between the two classes (correct and false putative correspondences) is significantly better than the one in the previous, unimodal histograms. The quality of this discrimination is evaluated in section 4.3 using a standard ROC analysis (see appendix A).

4.3.2 Receiver Operating Characteristic (ROC) analysis

We processed a Receiver Operating Characteristic (ROC) analysis of the results (see appendix A) to those numbers, in order to obtain a threshold independent understanding of the performance of the different criteria. Fig. 4.9 shows the ROC curves of the stereo selection, using each modal similarity and the optimal multi–modal mixing. All modal similarities produce a stereo–matching better than chance, and

the multi–modal criterion performs best.

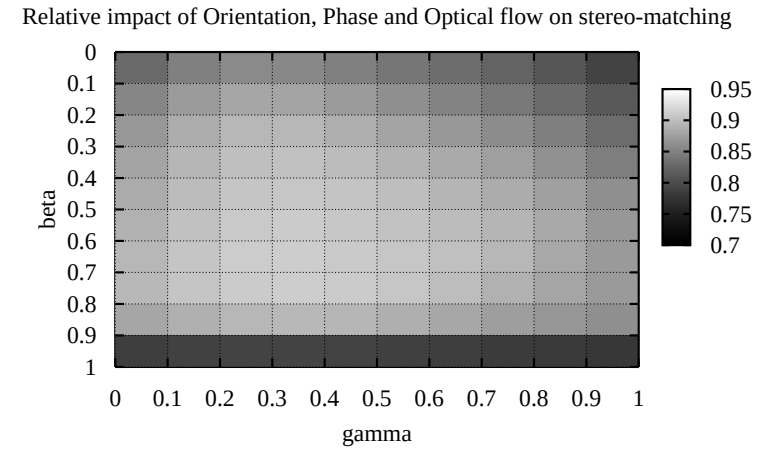
The impressive performance of the colour modality in this figure is explained by two facts: first, the colour modality of a primitive is a complex value — 6 to 9–dimensional, depending on the phase and the orientation, see 2. Consequently, the colour modality encompasses a larger amount of information than the other modalities. Second, the relative artificialness of the stereo sequences with ground truth we used for this evaluation makes the the images' colour to be perfectly identical between two views of the scene, whereas discrepancies in the two cameras, sampling differences, and lighting oddities can lead to significant colour variations in real scenarios.

We conducted a rough exploration of the parameter space of equation (4.4), to assess the relative impact of each modality on the similarity based stereo–matching, depending on the modality weighting. The results gathered over 30 stereo–frames, for which ground truth was available, are reported in Fig. 4.8. In this figure, the modalities are weighted relatively to the parameters α, β, and γ — defined as follows:

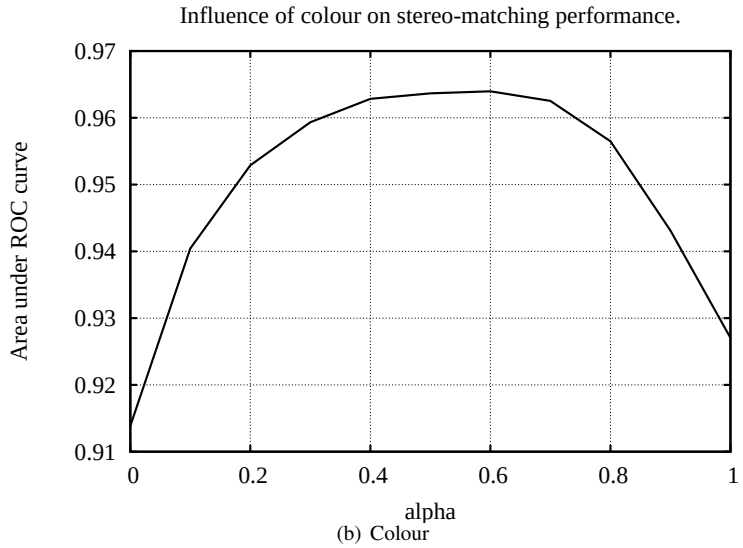
$$\begin{cases} w_{\boldsymbol{c}} &= \alpha \\ w_{\theta} &= (1-\alpha)\cdot\beta \\ w_{\omega} &= (1-\alpha)\cdot(1-\beta)\cdot\gamma \\ w_{\boldsymbol{f}} &= (1-\alpha)\cdot(1-\beta)\cdot(1-\gamma) \end{cases} \tag{4.5}$$

Hence, α represents the relative weighting of colour versus the other modalities, β represents the weighting of the orientation modality against the others, and γ is the weighting of phase against optical flow. This weighting has been chosen to compare the role of colour with the other modalities — as it appeared to be the strongest cue for stereo–matching as shown in Fig. 4.9.

In these graphs we see that the best performance (indicated by the largest area under the ROC curve), is obtained for a mix of the different modalities: the peak performance was obtained for $\alpha = 0.5$ $\beta = 0.6$ and $\gamma = 0.3$. Moreover, the performance curves are smooth, indicating that the system is robust to small parameters change.



(a) Orientation, Phase and Optical Flow



(b) Colour

Figure 4.8: Area under the stereo similarity ROC curve, as a function of the weights. a) β represents the weight of the orientation modality against the others, and γ the weighting of phase against optical flow. b) α represents the relative weight of colour against the other modalities. This curve is drawn for the optimal values $\beta = 0.6$ and $\gamma = 0.3$. These results were obtained from 10 frames of the sequences shown in Fig. 4.5A, B, and C.

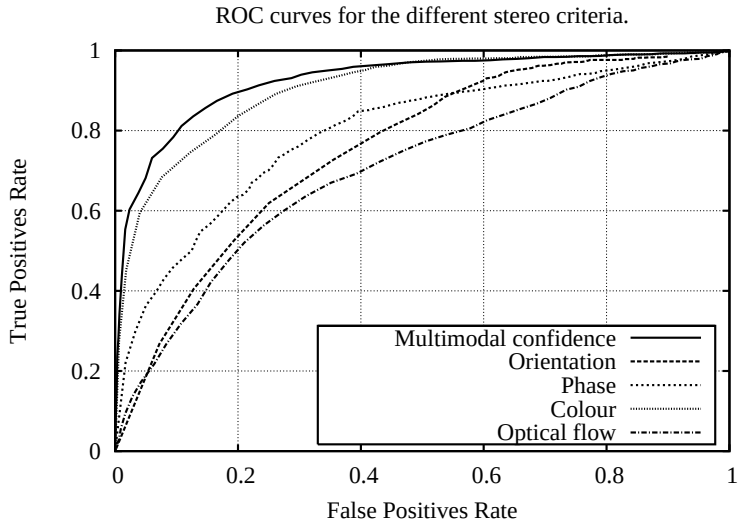


Figure 4.9: ROC analysis of the stereo, over 30 stereo images. These results were obtained from 10 frames of the sequences shown in Fig. 4.5A, B, and C.

4.4 Reconstruction

As the 2D–primitives are defined as *local* image descriptors, the result of a stereo–reconstruction from a given stereo–pair of 2D–primitives is also a *local* 3D entity, and it holds analogous multi–modal information for the 3D area it describes. The geometrical properties of a stereo–pair of primitives are naturally reconstructed: the position becomes a location in space and the orientation becomes a 3D line orientation — see section 4.4.1. In contrast to purely geometrical properties like position and orientation, appearance descriptors, like colour and phase, do not have a straightforward geometric meaning that can be reconstructed in 3D. In order to extend the two–dimensional appearance based information into the 3D domain, we need to define a reference plane onto which they apply — the reader is directed to (Mundy et al., 1996) for a discussion of geometric versus appearance descriptors. We define such a plane, based upon contextual knowledge of the stereo set–up geometry, in section 4.4.2.

Therefore, our 3D–primitive is defined by a local 3D surface patch, a line tangent to this surface, and the multimodal qualities of this 3D surface, on both sides of that line, inferred from the pair of 2D–primitives. We will define our 3D–primitive as the vector:

$$\mathbf{\Pi} = (\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\Gamma}, \Omega, \boldsymbol{C}, \Lambda)^T \tag{4.6}$$

Such a 3D–primitive can be seen in Fig. 4.12. The set of all reconstructed 3D–primitive is called the *Spatial Representation*. The modalities of this 3D–primitive are extensions of the 2D–primitive's:

- The 3D–primitive's location in space is encoded in a vector $\boldsymbol{X} = (x, y, z)^T$. The reconstruction of this vector is exposed in section 4.4.1.
- The 3D–primitive's orientation in space is two dimensional, and can be represented, either by the two angles (Φ, Ψ), either by the direction vector Θ. In the following we will use one or the other, depending on which is more convenient. In a similar way that we defined a 2D–primitive's orientation between 0 and π to remove ambiguity, we will define the orientations in space in the half sphere towards positive z, such that: $\boldsymbol{\Theta} \equiv (\Phi, \Psi) \in [0, \frac{\pi}{2}] \times [0, 2\pi[$ — *cf.* Fig.4.10 and section 4.4.1.
- The vector $\boldsymbol{\Gamma}$ is a contextual vector generated from the combined viewpoints from where the 3D–primitive was extracted; it is required to define the reference plane in section 4.4.2.
- The size of the 3D–primitive, Λ, is the size of the 3D surface patch that projects onto the image patches of both 2D–primitives.
- The phase Ω of a 3D–primitive holds the contrast transition across the 3D surface patch it describes, in a manner similar to the phase ω of a 2D–primitive. Because this definition depends on the 3D surface patch, the correctness of the phase modality depends on the definition of the vector $\boldsymbol{\Gamma}$. Its computation is described in section 4.4.2.
- The colour $\boldsymbol{C} = (\boldsymbol{C}_{\boldsymbol{\Gamma}}, \boldsymbol{C}_m, \boldsymbol{C}_{\overline{\boldsymbol{\Gamma}}})$ of a 3D–primitive encodes the colour transition on the 3D surface patch, across the contour, similarly to the 2D–primitive's colour modality $\boldsymbol{c} = (\boldsymbol{c}_l, \boldsymbol{c}_m, \boldsymbol{c}_r)$ 3D–primitive— the relation between these two vectors is explained in section 4.4.2. Because it is relative to the 3D surface patch, it also depends on the definition of the vector $\boldsymbol{\Gamma}$.

We will neglect the optical flow modality of the 2D–primitive, $\boldsymbol{f}$: estimating 3D motion requires a more complex formulation, like the Rigid Body Motion presented in chapter 6.

4.4.1 Geometric reconstruction of 3D–primitive

Wolff (1989) discussed that a greater accuracy is obtained by reconstructing lines directly rather than reconstructing pairs of points, and then inferring the line. Indeed, because the chance for two 3D lines

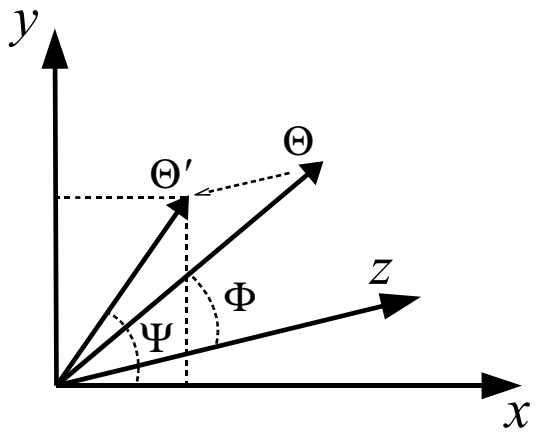


Figure 4.10: Here $\Phi \in [0, \frac{\pi}{2}]$, and $\Psi \in [0, 2\pi[$ define the half–sphere containing the direction vectors Θ with positive z values. Φ represents the angle between the orientation vector Θ and the orientation of the z–axis. Ψ represents the angle between the projection Θ' of Θ onto the xy–plane, and the x–axis. Therefore, a value pair of $(\Phi, \Psi) = (\frac{\pi}{2}, 0)$ is the orientation of the axis x; a value pair of $(\Phi, \Psi) = (\frac{\pi}{2}, \frac{\pi}{2})$ encodes an orientation along the axis y; a value pair $(\Phi, \Psi) = (0, \alpha)$ for any value of α signify an orientation collinear to the z–axis.

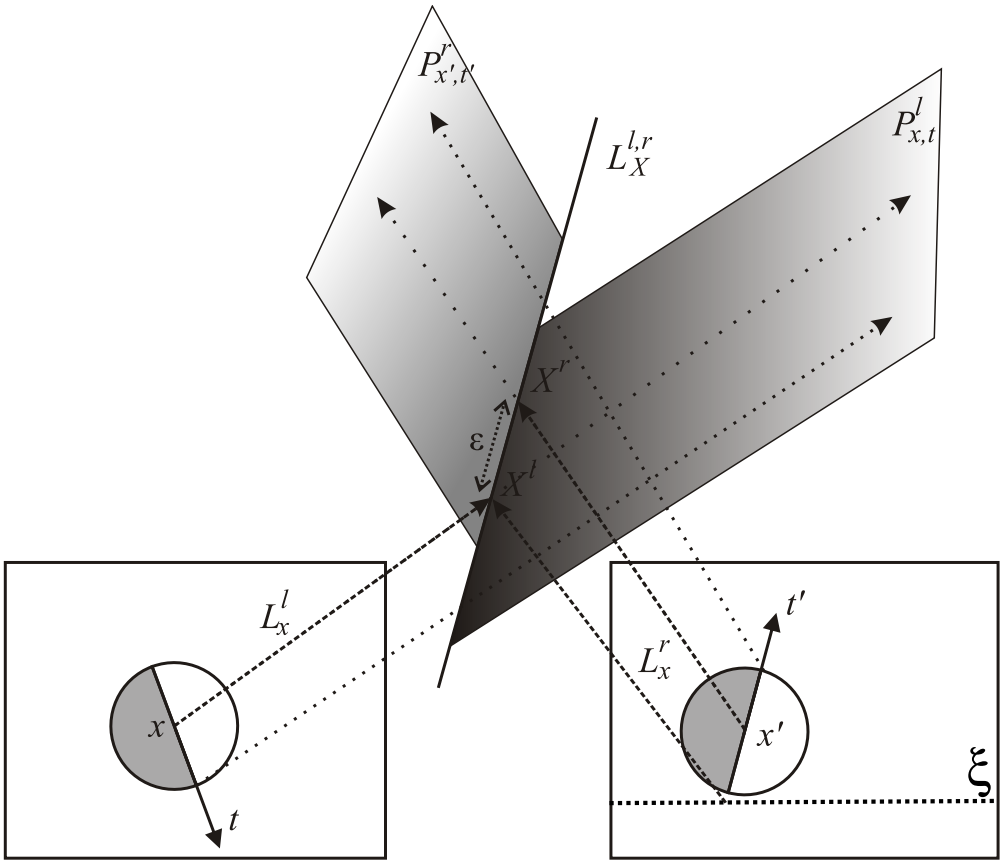


Figure 4.11: Example of the reconstruction of the 3D point M from a pair of primitives $\boldsymbol{\pi}^l$ and $\boldsymbol{\pi}^r$. The two 2D–primitives' possible origins in space draw two optical rays, respectively $L^l_{\boldsymbol{x}}$ and $L^r_{\boldsymbol{x}}$. When taking their respective orientation into account, we obtain two planes $P^l_{\boldsymbol{x}}$ and $P^r_{\boldsymbol{x}}$. We know the reconstructed point must lie on the intersection $L^{l,r}_{\boldsymbol{X}}$ between those two planes. Consequently, we reconstruct the point $\boldsymbol{X}^l$ at the intersection between $L^l_{\boldsymbol{x}}$ and $P^r_{\boldsymbol{x}}$. The distance $\varepsilon = d(\boldsymbol{X}^l, \boldsymbol{X}^r)$ is a good estimation of the imprecision of the reconstruction process.

to intersect is vanishingly small, the reconstruction of 3D points require an approximation step, (Liu et al., 2005). Moreover, due to a 2D–primitive's size and the sparse sampling, its stereo–correspondence is generally not centred on the epipolar line (see Fig. 4.2), which is an additional source of inaccuracy. Therefore, we reconstruct 3D–primitives in space by first reconstructing the 3D line in space that projects on both 2D lines defined by the stereo–pair of 2D–primitives; then we position the 3D–primitive on this line using the position of the 2D–primitivein the left image.

Reconstruction of line structures: If we consider two corresponding primitives $\boldsymbol{\pi}^l_{\boldsymbol{x}} \in \boldsymbol{\mathcal{I}}^l$ and $\boldsymbol{\pi}^r_{\boldsymbol{x}'} \in \boldsymbol{\mathcal{I}}^r$ with orientations collinear to, respectively, $\boldsymbol{t}$ and $\boldsymbol{t}'$. Using equation (C.14) from annex C, the possible reconstructions are contained in the line at the two planes intersection:

$$L^{l,r}_{\boldsymbol{X}} \equiv P^l_{\boldsymbol{x},\boldsymbol{t}} \cap P^r_{\boldsymbol{x}',\boldsymbol{t}'} \tag{4.7}$$

Note that 2D–primitives are only local line descriptors; hence, a line reconstructed from a stereo–pair of 2D–primitives is only meaningful nearby the 3D location designated by those two 2D–primitives' positions. The reconstruction of this location is explained in the next paragraph. The orientation of the line $L^{l,r}_{\boldsymbol{X}}$ provides the 3D–primitive's orientation. In order to handle the orientation–direction ambiguity (see section 2.2.4), we bind the orientation vectors in the half sphere of positive z coordinates — see Fig. 4.10.

Reconstruction of position: We reconstruct the 3D–primitive's position $\boldsymbol{x}$ by intersecting the optical ray $L^l_{\boldsymbol{x}}$ (*cf.* formula C.11), generated by the location of $\boldsymbol{\pi}^l_{\boldsymbol{x}}$, with the plane $P^r_{\boldsymbol{x}',\boldsymbol{t}'}$, generated by the position and orientation of $\boldsymbol{\pi}^r_{\boldsymbol{x}'}$ — reconstructed using equation (C.12).

$$\boldsymbol{X} \equiv L^l_{\boldsymbol{x}} \cap P^r_{\boldsymbol{x}',\boldsymbol{t}'} \tag{4.8}$$

The coordinates of this point are given by equation (B.30). Because the optical ray is built using the exact same points as the plane, we have ensured that the point $\boldsymbol{X}$ falls exactly on the line $L^{l,r}_{\boldsymbol{X}}$.

Reconstruction uncertainty: Second, the geometric reconstruction's uncertainty can be estimated by the distance from the points reconstructed from left and right optical rays

$$\varepsilon = d(\boldsymbol{X}^l, \boldsymbol{X}^r) \tag{4.9}$$

If $\varepsilon = 0$, the two optical rays intersect and the reconstructed position is the same as in appendix C.4.

Reconstruction of the size: The size of a 3D–primitive is the area of the 3D surface patch that is described by the stereo–pair of 2D–primitives. This also indicates the volume of the space that re–projects onto the receptive fields of both 2D–primitives in the stereo–pair. Effectively, this volume is dependent on the orientation and on the depth of the reconstructed 3D–primitive. In this work we only need an approximation of this size — for display purposes.

In order to obtain this approximation we consider the point $y = \boldsymbol{x} + \lambda \boldsymbol{t}$ (where $\boldsymbol{t}$ is a unit vector with orientation θ) in the left image plane, and compute the reconstruction Y of this point as we did for $\boldsymbol{x}$ in equation 4.8:

$$\boldsymbol{Y} \equiv L^l_y \cap P^r_{\boldsymbol{x}',t'} \tag{4.10}$$

Therefore, the 3D–primitive's size Λ in space is

$$\Lambda = ||\boldsymbol{Y} - \boldsymbol{X}|| \tag{4.11}$$

Equations (C.14) and (4.8) enable us to reconstruct the position and orientation of the 3D–primitive. In the next section we will explain the reconstruction of the other modalities.

4.4.2 Reconstruction of colour and phase

A position in space can be computed from two corresponding points, and an orientation in space can be computed from two 2D–primitives' positions and local orientations. The 3D reconstruction of a 3D–primitive's colour and phase modalities is not as straightforward. Consider the 2D–primitive as a local descriptor of an image contour, its phase models the intensity transition in the image across the contour; its colour models the hue and saturation on both sides and along the contour. Hence they represent appearance based information. In the 2D domain, the local orientation divides the image plane and

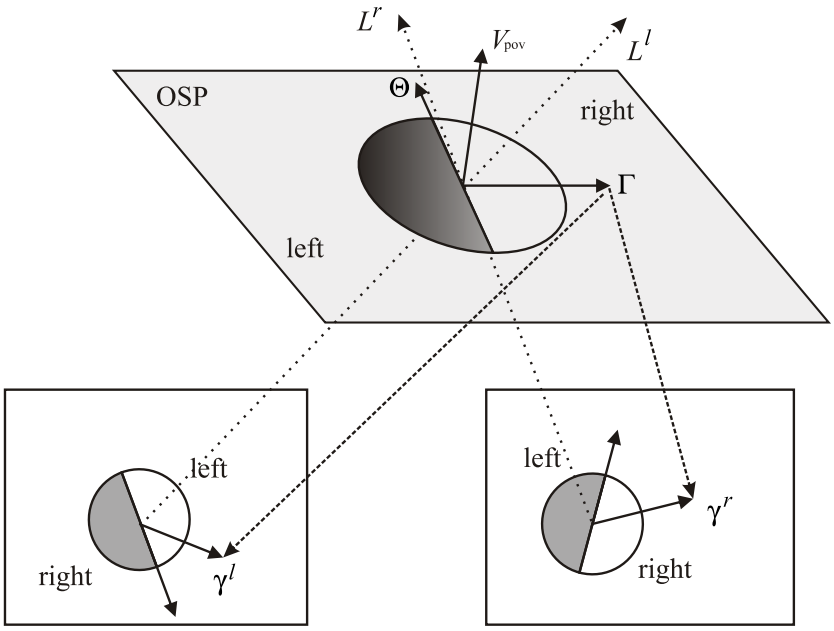


Figure 4.12: Illustration of the importance of the OSP for the reconstruction of the 3D–primitive. The OSP is defined by the two vectors Θ, the reconstructed 3D orientation, and $\mathbf{\Gamma}$, the the viewpoint–dependent component. The area of the 2D–primitives in the left and right image where the projection $\boldsymbol{\gamma}$ of the vector $\mathbf{\Gamma}$ falls is sampled from both primitives, and is called the 'left' side of the 3D–primitive. Conversely, the other side is called the 'right' side. This definition allows to faithfully re–project the image's properties across the contour.

gives an unambiguous meaning to the phase transition and the colour separation. In the 3D domain, the orientation is insufficient to divide uniquely the space. This creates an ambiguity in the meaning of those two modalities. To resolve this ambiguity, we need to define a reference plane in space, onto which they apply. The information held by a stereo–pair of 2D–primitives does not allow to infer this plane uniquely. However, it is possible to define a plane that suitably divides the space using the pose of the cameras from which the 3D–primitive is reconstructed. This plane is defined relatively to those viewpoints, and is an accurate approximation for any viewpoint close to the two it has been computed from. We call this approximation the Optimal Surface Patch (OSP).

Definition of the OSP: One point and two vectors in space are required to define a plane; for the OSP definition, the point and the first vector are given by the 3D–primitive's position and orientation (as

defined in the previous section). Therefore, we need an additional vector to completely define the OSP, that we compute using the pose information from the cameras' projection matrices. This vector is called $\mathbf{\Gamma}$ and defined by:

$$\mathbf{\Gamma} = \mathbf{\Theta} \times \boldsymbol{V}_{\text{pov}}, \tag{4.12}$$

Thus, the surface is defined to be normal to the viewpoint dependent vector $\boldsymbol{V}_{\text{pov}}$, computed from both cameras' projection matrices:

$$\boldsymbol{V}_{\text{pov}} = \frac{1}{2}\left(\overrightarrow{C_L\boldsymbol{X}} + \overrightarrow{C_R\boldsymbol{X}}\right), \tag{4.13}$$

where $\overrightarrow{C_L\boldsymbol{X}}$ and $\overrightarrow{C_R\boldsymbol{X}}$ are the two optical rays joining the location of the primitive $\mathbf{\Pi}$ with the optical centre of the left (C_L) and right (C_R) camera. The vector $\mathbf{\Gamma}$ not only allows us to define a 3D surface but also to identify uniquely both sides of the 3D–primitive, and to associate them to sides of the 2D–primitives — see Fig. 4.12.

Switching during reconstruction: As stated in section 2.2.4, a given primitive has two different interpretations that both define the same image patch. This orientation–direction ambiguity was addressed several times in this work, in different contexts. Here we address it in the context of stereo–reconstruction.

In the 3D domain, the OSP is divided by the 3D line that was reconstructed from the 2D–primitives' orientations. Because (by definition) $\mathbf{\Gamma}$ lies into the OSP, and because it is orthogonal to the line's orientation $\mathbf{\Theta}$, it identifies each side of the OSP: the 'left' side of the 3D–primitive is defined as the half–plane where $\mathbf{\Gamma}$ falls, the 'right' side is the other half–plane. In order to reconstruct the 3D–primitive's phase and colour modalities, we re–project $\mathbf{\Gamma}$ onto both image planes I^l and I^r, into the vectors $\boldsymbol{\gamma}^l$ and $\boldsymbol{\gamma}^r$, respectively, defining how the 'left' and 'right' sides of the 2D–primitives relate to the 'left' and 'right' sides of the reconstructed 3D–primitive.

Phase reconstruction : The 3D phase value is estimated as the average between the phases of the two 2D–primitives from which it is reconstructed.

$$\Omega = \arg\left(\frac{e^{i\omega^l} + e^{i\omega^r}}{2}\right) \tag{4.14}$$

If the vector $\boldsymbol{\gamma}$ does not point in a 2D–primitive's 'left' half, this 2D–primitive's phase is switched prior to this averaging — see 3.2.2.

Colour reconstruction : The colour of the 'left' side of the 3D–primitive is the average between both 2D–primitives' colour on the side designated by $\boldsymbol{\gamma}$, and conversely for the 'right' side. We write $\boldsymbol{C}_{\boldsymbol{\Gamma}}$ the colour of the 3D–primitive on the 'left' side, $\boldsymbol{C}_{\overline{\boldsymbol{\Gamma}}}$ the colour on the 'right' side, and $\boldsymbol{C}_{\boldsymbol{x}}$ the colour reconstructed on the line. Similarly, we call $\boldsymbol{c}_{\boldsymbol{\gamma}}$ the colour of the primitive $\boldsymbol{\pi}^s$ on the side designated by the projection $\boldsymbol{\gamma}$ of the vector $\boldsymbol{\Gamma}$, $\boldsymbol{c}_{\overline{\boldsymbol{\gamma}}}$ the colour of the other side, and $\boldsymbol{c}_{\boldsymbol{x}}$ the colour on the line,

$$\left\{\begin{array}{rcl} \boldsymbol{C}_{\boldsymbol{\Gamma}} & = & \frac{1}{2}\left(\boldsymbol{c}^l_{\boldsymbol{\gamma}} + \boldsymbol{c}^r_{\boldsymbol{\gamma}}\right) \\ \boldsymbol{C}_{\overline{\boldsymbol{\Gamma}}} & = & \frac{1}{2}\left(\boldsymbol{c}^l_{\overline{\boldsymbol{\gamma}}} + \boldsymbol{c}^r_{\overline{\boldsymbol{\gamma}}}\right) \\ \boldsymbol{C}_m & = & \frac{1}{2}\left(\boldsymbol{c}^l_m + \boldsymbol{c}^r_m\right) \end{array}\right. \tag{4.15}$$

Note that step–edges, for phase values within $\frac{\pi}{4} \leq |\omega| < \frac{3\pi}{4}$, do not have a middle colour component $\boldsymbol{C}_m$, and therefore colour is only sampled over the 'left' and 'right' sides — *cf.* section 2.2.4. The quality of colour and phase modalities' re–projections depends on the quality of the OSP: an inaccurate OSP will tend to provoke erroneous switching of the 'left' and 'right' colour values when re–projecting.

In Fig. 4.13, details of the 3D–primitives reconstructed in a simple scenario are shown. Although this is a relatively simple scenario, there are serious cases of occlusion (the basket's handle), shadows, and reflections. Therefore, although the 3D–primitives describe accurately parts of the objects, some parts are inaccurate or missing. This is unavoidable due to the local nature of our algorithm. In the rest of this thesis, we will address these problems using feedback from other vision processes.

4.5 3D–primitives reprojection and error measurement

In the following sections, we will see that comparing primitives in 2D instead of 3D allows to circumvent the noise and imprecision due to 3D reconstruction. Hence, we want to compare two 3D–primitives in the image planes. A straightforward way to achieve this is to *re–project* the 3D–primitives onto both image planes, thus generating a pair of 2D–*pseudo*–primitives [1] — henceforth called *re–projected primitives*. The 2D–primitives reprojected by two 3D–primitives can then be used to compare them in the image plane.

[1] Pseudo– in the sense that they do not carry actual information from the image, like the genuine 2D–primitives, but are inferred from a 3D–primitive

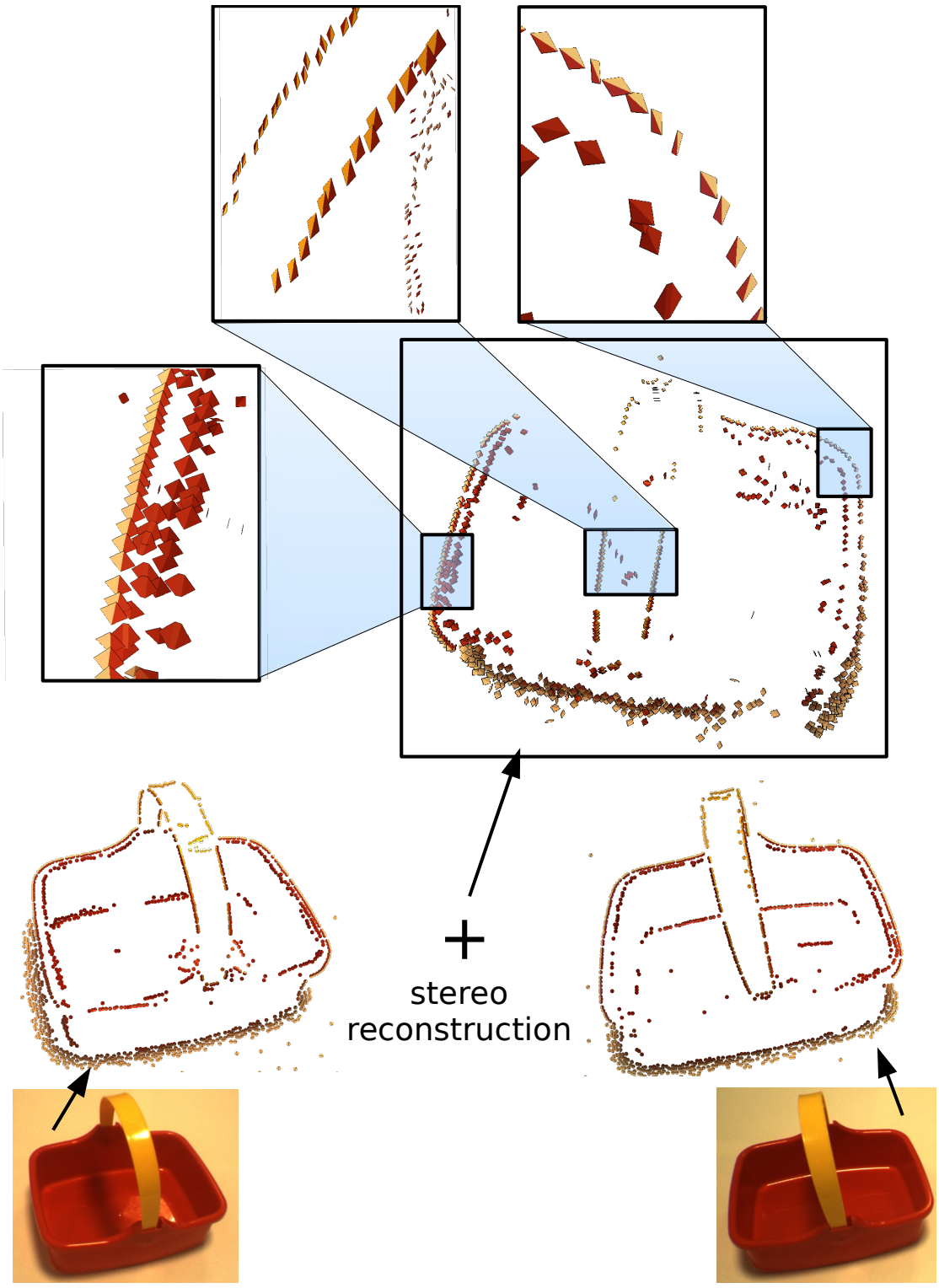


Figure 4.13: Example of the 3D–primitive reconstruction

We can obtain those reprojected primitives from the projection matrices: we will consider the 3D–primitive $\mathbf{\Pi} = (\boldsymbol{X}, \boldsymbol{\Theta}, \boldsymbol{\Gamma}, \Omega, \boldsymbol{C}, \Lambda)^T$, for which we want to have the reprojection $\boldsymbol{\pi}$ on an image plane $\boldsymbol{\mathcal{I}}$, using a projection matrix $\boldsymbol{P}$. From appendix C we know that the position is:

$$\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{P}}\tilde{\boldsymbol{X}} \tag{4.16}$$

where tilde indicate the use homogeneous coordinates — see appendix C. Obtaining the orientation of the reprojected primitive from the orientation of the 3D–primitive is just as straightforward (note that we use the orientation vector in this formula, and not the angular representation)

$$\tilde{\boldsymbol{t}} = \left(\tilde{\boldsymbol{P}}\left(\tilde{\boldsymbol{X}} + \tilde{\boldsymbol{\Theta}}\right) - \tilde{\boldsymbol{P}}\tilde{\boldsymbol{X}}\right) \tag{4.17}$$

At this point, we need to address once more the orientation ambiguity in infer a valid orientation from the vector $\boldsymbol{t}$ (*i.e.*, an angle $\theta \in [0, \pi[$). Because other orientation–dependent modalities (namely, colour and phase) have not yet been assigned to the re–projected primitive, casting the orientation in the valid range is sufficient to resolve the ambiguity. We will address this ambiguity a second time when assigning those orientation–dependent modalities to the re–projected primitive.

The colour and phase modalities are assigned directly from the 3D–primitive's values, where 'left' and 'right' sides are associated using the reprojected of the Γ vector, that is obtained through the classical projection formula:

$$\tilde{\boldsymbol{\gamma}} = \left(\tilde{\boldsymbol{P}}\left(\tilde{\boldsymbol{X}} + \tilde{\boldsymbol{\Gamma}}\right) - \tilde{\boldsymbol{P}}\tilde{\boldsymbol{X}}\right) \tag{4.18}$$

Where $\tilde{\boldsymbol{\gamma}}$ is an homogeneous form of the vector $\boldsymbol{\gamma}$. If the angular orientation of $\boldsymbol{\gamma}$ is greater than the re–projected orientation θ, the colour from the 3D–primitive's 'left' and 'right' sides and its phase are assigned to the re–projected primitive:

$$\begin{cases} \boldsymbol{c}_l &= \boldsymbol{C}_\Gamma \\ \boldsymbol{c}_m &= \boldsymbol{C}_m \\ \boldsymbol{c}_r &= \boldsymbol{C}_{\bar{\Gamma}} \\ \omega &= \Omega \end{cases} \tag{4.19}$$

Otherwise, the 'left' and 'right' colours are switched and the phase value is set to its opposite:

$$\begin{cases} \boldsymbol{c}_l &= \boldsymbol{C}_{\bar{\Gamma}} \\ \boldsymbol{c}_m &= \boldsymbol{C}_m \\ \boldsymbol{c}_r &= \boldsymbol{C}_{\Gamma} \\ \omega &= -\Omega \end{cases} \tag{4.20}$$

Because the optic flow does not have a 3D representation, it is not re–projected — the re–projected primitive's optic flow is set to the null vector.

The size λ of the reprojected primitive is obtained from the size of the 3D–primitive using the following formula:

$$\lambda = \| \left(\tilde{\boldsymbol{P}} \left(\tilde{\boldsymbol{X}} + \Lambda \tilde{\boldsymbol{\Theta}} \right) - \tilde{\boldsymbol{P}} \tilde{\boldsymbol{X}} \right) \| \tag{4.21}$$

We have defined the re–projection as the reverse operation to the 3D–reconstruction, and thus we can now consider 3D–primitives either in the 3D domain, or in selected image planes. This duality will prove a valuable asset for later processes, where the imprecision of reconstructed 3D information is problematic. One example is the good continuation property, defined in chapter 3 for the 2D domain. The imprecision of 3D–reconstruction, makes it unlikely for the reconstructed 3D–primitives to form a smooth curve. In the next chapter, we will make use of the grouping defined in the 2D domain to improve stereo–matching and 3D reconstruction.

4.6 Discussion

In this section we proposed a simple, local stereo matching algorithm adapted to the primitive–based image representation presented in chapter 2. This matching algorithm uses the multiple modalities associated to the 2D–primitives in order to draw a strong similarity constraint, and proposes an adaptation of the classical epipolar constraint to fit the sparseness of our image representation. This matching process is therefore a purely local one, where the only global knowledge that is assumed is the projective properties of the cameras. These properties are fixed in our scenario, and can be easily computed. Being a purely local matching process, this algorithm's aim is not to generate a flawless disparity map, but rather to provide the plausible disparity values for each 2D–primitive. Because we do not discard less likely

correspondences at this stage, it is then possible to correct the original assumption using contextual information, when it is available. This contextual information can be derived, for example, from perceptual grouping, motion, or object knowledge. In this context, the most likely stereo–correspondences provided by this stereo system serve to initiate some of the higher level processes, like motion estimation (see chapter 6) or temporal accumulation of visual information (see chapter 7).

In a second part, we use these correspondences to reconstruct the three–dimensional equivalent of a 2D–primitive, that we called 3D–primitive. We proposed solutions to reconstruct each modality held by the stereo–pair of 2D–primitives (with the exception of optic flow) into the 3D domain. This reconstruction required the assumption of additional knowledge (the so–called Optimal Surface Patch) that we inferred from the projection matrices parameters. Here again, the estimated OSP is not expected to be true to the 3D structure of the scene, but it is a first approximation, optimal for the available information. It can be corrected at later processing stages, for example during the inference of object surfaces (as done, *e.g.*, by Kalkan et al. (2007a)). The duality between stereo–pair of 2D–primitives and 3D–primitives is useful, in the sense that it allows to handle each visual sub–problem in the domain that is most adequate. For example, 3D information suffers from imprecision due to the reconstruction process. In the previous chapter, we defined a grouping mechanism based on the *Gestalt* laws of good continuation and similarity. To define good continuation in the image plane is fairly simple, but it is difficult in the 3D space: due to the imprecision of the reconstructed depth, contours that are smooth in 2D are less likely to be smoothly reconstructed in space. Conversely, while the 2D appearance of objects viewed in different poses changes broadly, its 3D representation is more stable because distances and angles are preserved. Therefore, in the following we will make use of the 3D–primitives, either directly in the 3D space, either through their reprojection in some image plane.

Finally, it would be straightforward to augment the current scheme using some of the global constraints and optimisations discussed in the introduction of this chapter. This is recommended if no post–processing is applied, and if the desired output of the system is just a final depth map, and if accuracy is primordial.

Chapter 5

Spatial Consistency Constraint Applied to Stereo

The true mystery of the world is the visible, not the invisible.

- Oscar Wilde

In the previous section we exposed how 2D–primitives can be matched over two stereo images, and used to infer structures of the 3D scene witnessed. Despite their strong semantic content, the 2D–primitives still suffer from local ambiguity, and instability of the signal, and therefore can generate an inexact reconstruction of the scene.

We face two kinds of errors when interpreting visual information. First, the ambiguity in the matching process creates a first source of error. Erroneous matches will lead to the reconstruction of wrong 3D–primitives. Second, even correct matches can lead to inaccurate reconstructions, if the original 2D–primitive extraction was unprecise, if the the 3D–primitive is reconstructed far from the camera, or if the 2D–primitive's orientation is close to the epipolar line's.

This chapter proposes to make use of the groups defined in chapter 3 to improve the stereo–reconstruction described in chapter 4. Improvements in the resulting scene representation $\mathcal{S}$ are evaluated using three measures of quality: First, we call *reliability* of the representation, the proportion of correct stereo–matches. Second, the reconstruction of those *correct* stereo–matches is subject to errors due to the

sampling and local signal imprecision; we call *accuracy* of the representation, the correctness of the reconstructed 3D–primitives— compared to 3D ground truth. Third, we call *density* of the representation, the number of 3D–primitives inferred by the system relatively to the number of 2D–primitives originally extracted. The higher the requirements in reliability and accuracy, the lower becomes the density, and vice versa. Hence, different compromise are preferable for different tasks.

Intuitively, a group defines a contour of the scene, conjointly described by the primitives it contains. This is important as this group is the first scene descriptor which is not dependent on the primitive sampling: depending on the viewpoint of the camera, the primitive extraction in the image of a given 3D contour can vary. Nevertheless, they will always form a group describing this contour regardless of the viewpoint. In this sense, grouping allows a more robust description of the image's structures than primitives. Note that a 3D contour can be partially (or even completely) occluded in one specific viewpoint. Also the contour might fall out of the field of view for this viewpoint. In those cases, the contour will not be (fully) represented in the groups extracted from this viewpoint. Nevertheless, assuming a small enough displacement between two viewpoints, we can assume that all groups in one frame have a corresponding group in other frames — both groups describing the same 3D contour.

Mayhew and Frisby (1981) proposed the so–called *figural continuity* constraint for stereopsis:

Definition 5.0.1. *Disparity along an edge contour changes smoothly, i.e., there should be no disparity discontinuities along a contour.*

They justified the use of this constraint with psychophysical experiments. In this chapter we propose a local implementation of this constraint to the primitives' framework presented in the previous chapters.

Various works in stereopsis proposed to use some amount of local consistency to improve the reliability of stereo–matching: Ohta and Kanade (1985) enforced intra–scanline constraints on the disparity of zero–crossings, effectively ensuring that disparity is continuous along contours. Herman and Kanade (1986) proposed a hierarchical symbolic representation of the scene, using prior geometric knowledge. Boyer and Kak (1988) proposed a structural stereo–matching using a cost function derived from information theory. Horaud and Skordas (1989) proposed a stereo–matching algorithm, based on a relational graph, which nodes are contours extracted in three steps: edge detection, edge linking, and piecewise segmentation. They argue that feature grouping is an essential step because it reduces combinatorial explosion. Mohan et al. (1989) implemented the figural continuity constraint while making the distinction

between local errors (that can be resolved by figural continuity) and global errors (that cannot). Hoff and Ahuja (1989) simultaneously addressed the problems of edge matching and surface interpolation. This is effectively an improvement on the disparity gradient and figural continuity constraints. Their model assumes that all surface slope discontinuities and occluding boundaries are located at image edges, and fit quadratic surfaces between them. Chung and Nevatia (1991, 1995) proposed to base stereo–matching on hierarchical features; high level (more abstract) features reduce the correspondence ambiguity, and the feature hierarchy is checked against the different views. Two interesting qualities of their approach is their explicit handling of occlusion, and their formalisation of curved surfaces (limb boundaries). This approach is limited by the performance of the monocular grouping, and the authors, and therefore seems inadequate for highly textured or unstructured scenes. Venkateswar and Chellappa (1995) proposed a hierarchical, feature based, stereo–matching algorithm. Their hierarchy consists of lines, vertices (junctions), edges (contours), and edge–rings (groups of contours). Features in the hierarchy are hypotheses that are checked against a truth maintenance system.

In chapter 3, we proposed a pairwise evaluation of affinity between primitives, and used it to define a graph of grouping relations $(\mathcal{I}, \mathcal{L})$ in the image representation $\mathcal{I}$. In this chapter we will use the local link structure in the vicinity of a 2D–primitive in order to ensure the figural continuity of potential stereo–correspondences, and thus improve the reliability of the stereopsis. This is different from the disparity gradient in the sense that the disparity is not explicitly constrained; also, because this process applies on contours, cases of occlusions are implicitly handled. Moreover, the grouping relation we use here stems completely from the *Gestalt* laws of good continuation and similarity, does not require the explicit definition of higher level features. Indeed, we intend to produce a 3D representation of the scene using minimal assumptions about the scene, to leave the interpretation to a higher, more informed level of processing.

Section 5.1 presents a scheme to re–evaluate the confidence in a stereo–match using proximate linked primitives. This re–evaluation lead to a better selection of the potential correspondences by the winner–take–all mechanism, and to an improved reliability of the stereo reconstruction. reliability. Section 5.2 extends the primitive interpolation scheme exposed in section 3.4 to reconstructed 3D–primitives, thus improving the accuracy thereof.

5.1 Perceptual grouping constraints to improve stereopsis

Because we consider only intrinsically one–dimensional primitives (lines and step–edges), the resulting representation is very redundant along contours. This redundancy allows us to use perceptual grouping to derive the following two constraints for the matching process:

Isolated primitives are likely to be unreliable: A contour is encoded in our representation by a string of primitives. Therefore an *isolated* primitive is either 1) a correct primitive that failed to be grouped, 2) a correct primitive describing a small feature of the scene, which does not generate more primitives, or 3) an erroneous primitive, extracted from texture or noise. Note that the cases 2) and 3) are two interpretations of the same case: the distinction between texture and small structure is merely a question of scale. In all three cases, though, the primitive, on its own, will not be useable by higher level processes, and can therefore be discarded.

Stereo consistency over groups: If a set of primitives forms a contour in the first image, the *correct correspondences* of these primitives in the second image also form a contour (apart from occlusions). This is illustrated in Fig. 5.1: the primitive $\boldsymbol{\pi}_i$ is the one most similar (according to equation 4.4) to $\boldsymbol{\pi}_2$ (mainly due to very similar orientation). Hence, this stereo–correspondence $s_{2\to i}$ holds a higher local confidence than does, *e.g.*, $s_{2\to j}$. Nonetheless, only the putative correspondence $\boldsymbol{\pi}_j$ forms a link $g_{s,j}$ with $\boldsymbol{\pi}_s$, conserving the link $g_{1,2}$ between $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$.

In the following we will make use of these two properties to re–evaluate the confidence in potential stereo–correspondences, in order to reduce the number of outliers in the 3D model of the scene generated by our stereo system.

5.1.1 Basic Stereo Consistency Event (BSCE)

As explained in chapter 2, 2D–primitives represent local estimators of image contours. A constellation of those 2D–primitives describes the contour as a whole. Those contours are consistent over stereo, with the notable exception of partially occluded contours — see Fig. 4.5, bottom row. In chapter 3, we defined the likelihood for two 2D–primitives to describe the same contour as the affinity between these two 2D–primitives. Hence, we can rewrite the previous statement as:

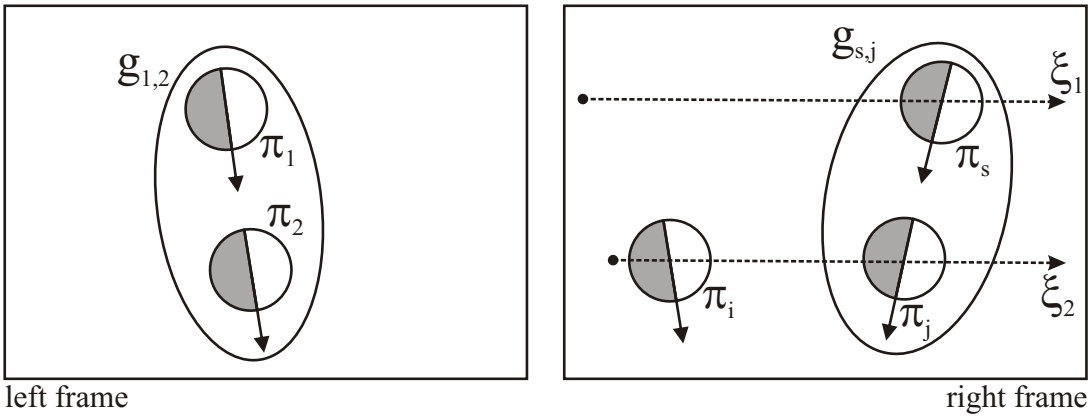


Figure 5.1: The BSCE criterion: Let $\boldsymbol{\pi}_1$ be a primitive in the left image, linked with a second primitive $\boldsymbol{\pi}_2$. $\boldsymbol{\pi}_2$ has a stereo–correspondence $\boldsymbol{\pi}_s$ in the right image. $\boldsymbol{\pi}_1$ has two possible correspondences $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$, lying on its epipolar line ξ_1 in the right image. In this case, the correspondence $\boldsymbol{\pi}_i$ is clearly more similar to $\boldsymbol{\pi}_1$, and yields a higher multimodal confidence than $\boldsymbol{\pi}_j$. On the other hand, when considering the BSCE criterion, only the correspondence $\boldsymbol{\pi}_j$ forms a link $g_{j,s}$ with $\boldsymbol{\pi}_s$, conserving the relation $g_{1,2}$ between $\boldsymbol{\pi}_1$ and $\boldsymbol{\pi}_2$.

Definition 5.1.1. *Given two 2D–primitives $\boldsymbol{\pi}_i^l$ and $\boldsymbol{\pi}_j^l$ in $\mathcal{I}^l$ and their respective correspondences $\boldsymbol{\pi}_n^r$ and $\boldsymbol{\pi}_p^r$ in a second image $\mathcal{I}^r$; if $\boldsymbol{\pi}_i^l$ and $\boldsymbol{\pi}_j^l$ belong to the same group in $\mathcal{I}^l$ then $\boldsymbol{\pi}_n^r$ and $\boldsymbol{\pi}_p^r$ should also be part of a group in $\mathcal{I}^r$. — see Fig. 5.1.*

We call the conservation of the link between a pair of 2D–primitives by the stereo–correspondences of those 2D–primitives, the *Basic Stereo Consistency Event* (BSCE). This condition can then be used to test the validity of a stereo–hypothesis. Consider a 2D–primitive $\boldsymbol{\pi}_i^l$, and a stereo–hypothesis:

$$s_{i\to n} : \boldsymbol{\pi}_i^l \to \boldsymbol{\pi}_n^r \tag{5.1}$$

and consider a neighbour $\boldsymbol{\pi}_j^l \in \mathcal{N}(\boldsymbol{\pi}_i^l)$ of $\boldsymbol{\pi}_i^l$ such that the two 2D–primitives share an affinity $c\left[g_{i,j}\right]$. For this second 2D–primitive a stereo–correspondence $\boldsymbol{\pi}_p^r$ with a confidence of $c\left[s_{j\to p}\right]$ exists. We can then estimate how well the stereo–hypothesis $s_{i\to n}$ preserves the BSCE by:

$$E(g_{i,j}, s_{i\to n}) = \begin{cases} +\sqrt{c\left[s_{j\to p}\right] \cdot c\left[g_{i,j}\right]} & \text{if } c\left[g_{n,p}\right] > \varepsilon \\ -\sqrt{c\left[s_{j\to p}\right] \cdot c\left[g_{i,j}\right]} & \text{else} \end{cases} \tag{5.2}$$

In other words, considering a stereo–pair of 2D–primitives: the BSCE of a 2D–primitive in the first

image with one of its neighbours is high if they share a strong affinity and if this second 2D–primitive creates a stereo–hypothesis such that the correspondences in the second image of both 2D–primitives *also* share a strong affinity. It is low if the stereo–correspondences of this 2D–primitive, and others part of the same group, do not form a group in the second image. This naturally extends the concept of group that was defined in chapter 3 into the stereo domain.

5.1.2 Neighbourhood consistency Confidence

Building on the formula (5.2), we can define how *the whole neighbourhood* of a 2D–primitive is consistent with a given stereo–hypothesis.

Equation (5.2) tells us how a 2D–primitive's stereo–correspondence is consistent with our knowledge of plausible stereo–hypotheses for a second 2D–primitive in its neighbourhood. Now, if we consider a 2D–primitive $\boldsymbol{\pi}_i^l$ and an associated stereo–correspondence $s_{i\to n}$, we can integrate this BSCE confidence over the neighbourhood of the 2D–primitive $\mathcal{N}(\boldsymbol{\pi}_i^l)$ — as defined in section 3.2.4.

$$c_{ext}[s_{i\to n}] = \frac{1}{\#\left(\boldsymbol{\mathcal{N}}_i^l\right)} \sum_{\boldsymbol{\pi}_k^l \in \boldsymbol{\mathcal{N}}_i^l} E(g_{i,k}, s_{i\to n}) \tag{5.3}$$

where $\#\left(\boldsymbol{\mathcal{N}}_i^l\right)$ is the size of the neighbourhood — *i.e.*, the number of neighbours of $\boldsymbol{\pi}_i^l$ that have a link with $\boldsymbol{\pi}_i^l$. We call this new confidence the *external confidence* in $s_{i\to n}$, as opposed to the confidence given by the multi–modal similarity between the 2D–primitives — equation (4.4).

In Fig. 5.2, the correct (black) correspondences have mostly positive external confidences, while incorrect (white) ones have mainly negative values (sharp peak at -0.9). The small peak of correct correspondences for negative external confidence is due to the few cases where most 2D–primitives on a contour have an erroneous correspondence; therefore, the few correct ones are strongly contradicted by their neighbours. The application of a threshold on the external confidence removes stereo–hypotheses that are inconsistent with their neighbourhood, and thus reduces the stereo–matching ambiguity. Note that selecting a threshold of zero implies the removal of all the isolated 2D–primitives — because isolated 2D–primitives have an external confidence of zero by definition.

Fig. 5.3 shows ROC curves of the performance for varying thresholds τ_m on the multi–modal similarity. Each curve shows the performance of the stereo–matching process described in chapter 4 after

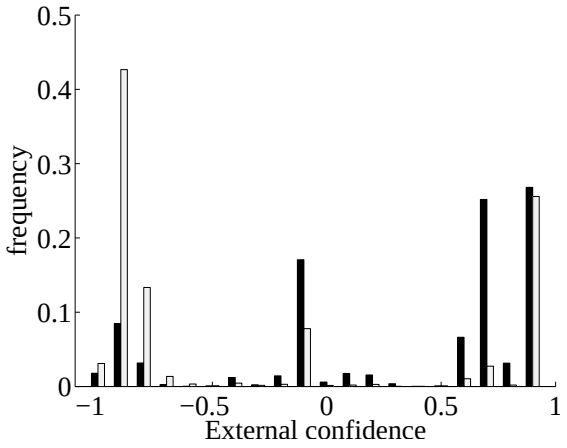


Figure 5.2: Histogram of the external confidence rating for correct (black bars) and false (white bars) correspondences. Those curves represent the statistics over 30 stereo images with ground truth. These results were obtained from 10 frames of the sequences shown in Fig. 4.5A, B, and C.

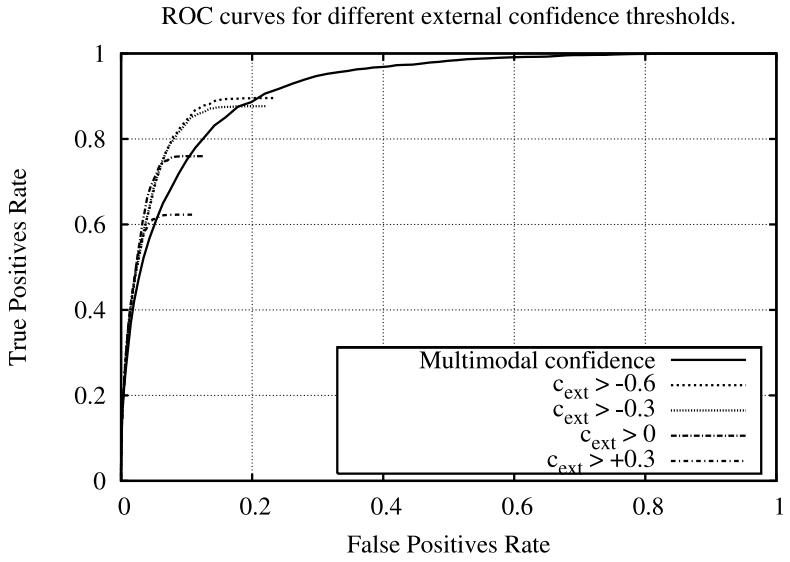


Figure 5.3: Each curve stands for a the application of a different threshold over the external confidence, prior to the ROC analysis. Those curves represent the statistics over 30 stereo images with ground truth. These results were obtained from 10 frames of the sequences shown in Fig. 4.5A, B, and C.

sequence	τ_m	τ_e	correct c	false f	$\frac{c-f}{c+f}$
A	0.8	-1.0	3633	498	0.76
A	0.8	-0.1	3582	456	0.77
B	0.8	-1.0	2205	1178	0.30
B	0.8	-0.1	1915	447	0.62
C	0.8	-1.0	906	276	0.53
C	0.8	-0.1	804	167	0.66

Table 5.1: Performance of the stereopsis with and without external confidence threshold. These results were obtained from 10 frames of the sequences shown in Fig. 4.5A, B, and C.

the application of different thresholds on the external confidence — respectively, for threshold values of $\tau_e \in \{-0.6, -0.3, 0, +0.3\}$, and without threshold. We can see from those results that the reduction of the ambiguity resulting from the application of a threshold on the external confidence improves the accuracy of the stereo–matching significantly. Depending on the type of scene representation desired (very selective and reliable, or more lax but denser) different threshold values will be chosen. The best overall improvement seems to be reached for a threshold of $\tau_e = -0.3$ over the external confidence. Nonetheless, if very high reliability is required, a threshold of $\tau_e = 0$ (meaning discarding all primitives which are part of no group) may be preferred. Because a threshold is applied to the external confidence *prior* to the ROC analysis, the resulting curve does not reach the $(1, 1)$ point of the graph. This is normal as the threshold already removes some stereo–hypotheses even before the multi–modal confidence is considered. Fig. 5.5(c) shows the 3D–primitives reconstructed with a threshold on external confidence of $\tau_e = -0.1$. When comparing Fig. 5.5(b) and Fig. 5.5(c) we can see that a large number of outliers are discarded from the reconstructed 3D–primitives, leading to a cleaner description of the scene.

5.2 Interpolation in space

One issue when reconstructing 3D structures from stereopsis is that the accuracy of the reconstructed depth is decreasing with the distance to the cameras — see (Wolff, 1989). Fig. 5.5(c) shows the reconstruction of the tree (along with the road markings) in an outdoor scenario. There, we can see that, although all 3D–primitives describe the contour of the tree from the same point of view, their exact position and orientation in space vary: they do not form a smooth contour in space. Nevertheless, the 2D perceptual grouping mechanism presented in chapter 3, applied to both stereo images, tells us that these

	localisation error		orientation error	
	mean	variance	mean	variance
before	0.03524	0.00392	0.01712	0.00082
after	0.02426	0.00221	0.01434	0.00056

Table 5.2: Effect of the correction process on the localisation and orientation in space of the primitives reconstructed from the triangle scenario.

form one perceptual group in both stereo images (as defined in section 5.1 and Fig. 4.5 bottom row). Thus, they do describe a smooth, continuous contour in space, and the variance in their re–constructed position and orientation is due to noise.

We propose to remove some of this noise by extending the 2D–primitives interpolation mechanism described in section 3.4 to 3D–primitives. Like in the 2D case, we will consider *triplets* of 3D–primitives, constituted of a central primitive $\mathbf{\Pi}_i$, and of two supporting primitives $\mathbf{\Pi}_j$ and $\mathbf{\Pi}_k$. These two 3D–primitives are both linked to the central one, such that the central 3D–primitive lies in between the two supporting 3D–primitives. The 3D–grouping is defined as follows:

Definition 5.2.1. *Two 3D–primitives $\mathbf{\Pi}_i, \mathbf{\Pi}_j \in \boldsymbol{S}$ are linked iff. their projections $\boldsymbol{\pi}_i^x$ and $\boldsymbol{\pi}_j^x$ in both image planes are linked, such that $g_{i,j}$ exists in both image representations ($x \in l, r$).*

For each iteration n of the smoothing, the central 3D–primitive's position $\boldsymbol{X}$, and orientation $\mathbf{\Theta}$, are corrected using the curve interpolated between the two supporting 3D–primitives:

$$\boldsymbol{X}_i^{(n)} = \frac{1}{2}\left(\boldsymbol{X}_i^{(n-1)} + \widehat{\boldsymbol{X}}_i^{(n-1)}\right), \tag{5.4}$$

and

$$\mathbf{\Theta}_i^{(n)} = \frac{1}{2}\left(\mathbf{\Theta}_i^{(n-1)} + \widehat{\mathbf{\Theta}}_i^{(n-1)}\right), \tag{5.5}$$

where $\widehat{\boldsymbol{X}}_i^{(n)}$ and $\widehat{\mathbf{\Theta}}_i^{(n)}$ refer to the position and orientation interpolated at iteration n from the triplet $(i, j, k)^{(n)}$.

This scheme was evaluated on the triangle sequence shown in Fig. 2.10 and resulted in a reduction of the localisation error by $\sim$ 30%. The orientation error was reduced by $\sim$ 16% — see table 5.2 and Fig. 5.4, solid curves. When applying the same scheme to the circle scenario, the localisation error was reduced by $\sim$ 20%. The orientation error was also reduced by $\sim$ 20% — see table 5.3 and Fig. 5.4, dashed curves.

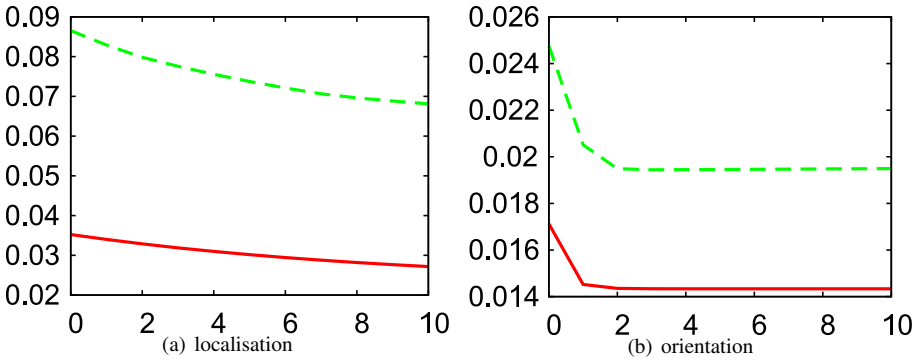


Figure 5.4: Error of the (a) localisation and (b) orientation of the reconstructed 3D primitives after several iterations of the correction process. The full line shows the errors for the triangle scenario and the dashed line for the circle scenario. The horizontal axis shows the number of iterations of the correction process and the vertical axis shows the error in (a) arbitrary units and (b) radians.

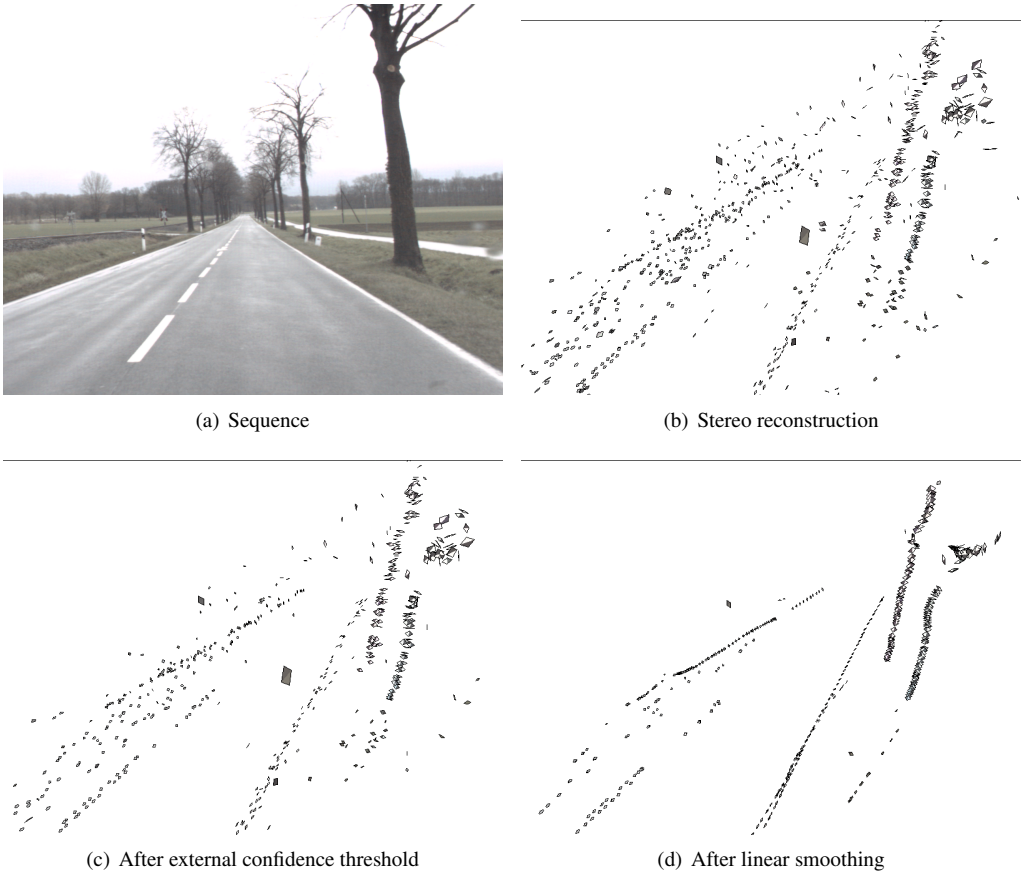
	localisation error		orientation error	
	mean	variance	mean	variance
before	0.08653	0.01188	0.02476	0.00071
after	0.06868	0.00882	0.01955	0.00046

Table 5.3: Effects of the correction process on the localisation and orientation in space of the primitives reconstructed from the circle scenario.

Furthermore, we qualitatively evaluated the improvement in accuracy obtained by this correction in more complex scenarios. Figure 5.5 illustrates the reconstructed 3D–primitives from an outdoor sequence. Note that it is necessary to choose a point of view sufficiently different from the one of the camera in order to highlight the reconstruction errors, while being sufficiently similar for the shapes of the scene to be recognisable. We chose a point of view located high on the right side of the scene, looking downwards at the road. Figure 5.5(d) shows the same part of the scene after 10 iterations of the Hermite smoothing. The 3D–primitives form the contour of the tree and the road markings are now smoothly aligned. Figure 5.6 shows the effect of this smoothing on selected details in an indoor scene.

5.3 Conclusion

In chapter 4, a multi–modal similarity measure was used to rate the potential correspondences between 2D–primitives in a stereo pair of images. The proposed stereo algorithm was purely local and therefore does not make use of global constraints beside the epipolar geometry or optimisation scheme —



(a) Sequence (b) Stereo reconstruction

(c) After external confidence threshold (d) After linear smoothing

Figure 5.5: Reconstruction of 3D–primitives from stereo–matches obtained from a real life outdoor sequence — see Fig. 4.5D. (b) shows the reconstruction resulting from a stereo–matching done using only the multi–modal stereo approach (with a threshold of 0.4 on the multi–modal confidence). (c) shows the reconstruction obtained when an additional threshold of -0.1 is applied to the external confidence. (d) shows the corrected entities, after 3 iterations of the linear smoothing process.

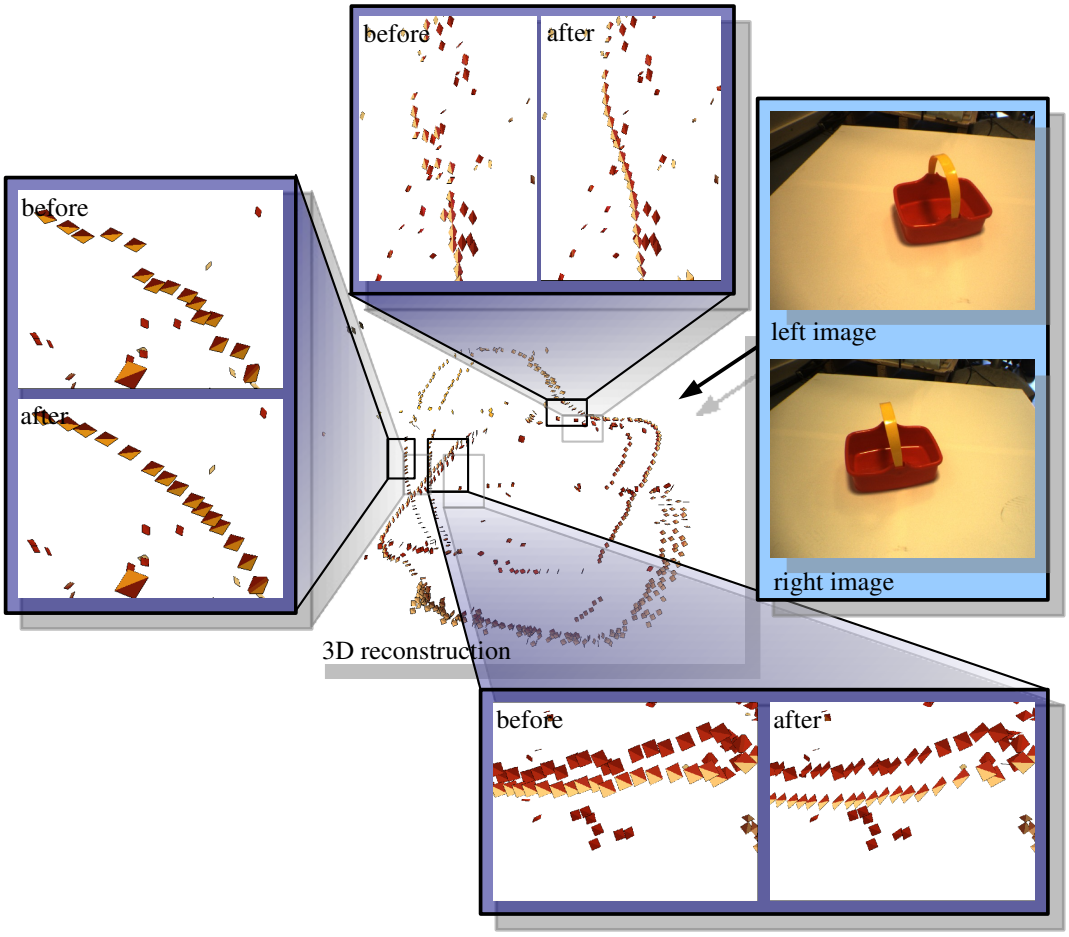


Figure 5.6: Illustration of the effects of the 3D–primitives' correction using interpolation.

cf. introduction of chapter 4), Such global optimisations generally improve the performance of local stereo–matching schemes, and therefore could be applied to this system to further improve the quality of the representation. This chapter proposed two mechanisms using feedback from contour grouping information (as described in chapter 3) in a 2D–primitive's neighbourhood to improve the quality of stereo–reconstruction — without the need for global constraints or prior knowledge about the scene.

Section 5.1 describes a local scheme, integrating the contour grouping and stereopsis to improve the reliability of the latter. The external confidence defined there is comparable in effect to the averaging over a local neighbourhood of a disparity gradient constraint along contours (Kim and Bovik, 1988). Because it applies only along contours, it is also comparable to a local implementation of the figural continuity property (Mayhew and Frisby, 1981), implemented into the primitive framework. In a similar fashion, Mohan et al. (1989) enforced that the disparity changes linearly along line segments. The main difference of the present approach is that: 1) because the line descriptors are local they permit handling generic curved contours; and 2) because the external confidence is based on perceptual grouping, there is no need to set an explicit threshold on the disparity, or its gradient. The definition of the BSCE provides a meaningful indicator, that stems directly from the good continuation definition, of which neighbour has a positive, or a negative, contribution to an hypothesis' confidence — and this using only the perceptual grouping defined in chapter 3. This constraint was evaluated in section 5.1.2, and was shown to significantly improve the stereo matching reliability.

Moreover, in section 3.4, it was showed that this relation can be used to interpolate contours between pairs of linked 2D–primitives. This was then used to correct 2D–primitives to the contour interpolated from its neighbours. In this chapter, we extended the link definition to 3D–primitives, and, in section 5.2, we used a similar method to interpolate 3D–primitives. When interpolating 3D–primitives, we reduced the localisation error by more than 20% and the orientation error by more 15%. This happened consistently for amount of noise varying from 0 to 10%. Therefore this interpolation step proved to be a robust manner to improve the representation's accuracy, both in 2D and 3D. Because the scheme is local, there is no *a priori* assumption that the whole contours comply with a certain mathematical description (we only assume that the contour is smooth between two proximate primitives, and model that using Hermite interpolation). This interpolation is a simple and local method to reduce the noise induced by the reconstruction of 3D–primitives, from stereo–correspondences of 2D–primitives.

We show here that using such mutual feedback between mid–level, local processes allows to disambiguate them without the need for additional contextual knowledge. Thereby, we provide a reliable and accurate 3D representation of the shapes in the scene, that can then be used for higher level visual operations, where contextual knowledge may be available.

Part III

Temporal integration $\mathcal{A}$

Chapter 6

Ego–motion Estimation

> It takes a little talent to see clearly what lies under one's nose, a good deal of it to know in which direction to point that organ.
>
> - W. H. Auden

Different kinds of motion exist in the natural world, some of them complex combinations of simpler motions. For example, the motion of a bird is a combination of its displacement in the sky, the movement of its wings, the deformation of its feathers due to the air pressure, *etc.*

In this chapter we will focus on a considerably simpler, and better defined, class of motion, namely the motion of the observer between two instants t and $t + \delta t$. Furthermore, we will assume a rigid stereo set–up, where the position and orientation of the cameras relative to one another is constant. Under this general assumption, the motion we are interested in is the transformation between the pose of the cameras at instant t and their new pose at instant $t + \delta t$ This is called *ego–motion*.

Estimating the ego–motion is an important problem for autonomous systems, *e.g.*, for navigation, time to impact estimation, obstacles avoidance, and more generally to integrate 3D sensory information over time. The stereo reconstruction presented in section 4.4 is relative to the observer position at this instant: thus the coordinate system where visual information is reconstructed at time t and at time $t + \delta t$ are different; therefore, those two 3D representations are not comparable. In order to compare and integrate them over time, we need a way to cast a spatial representation $\mathcal{S}^t$, reconstructed at time t, into the coordinate system of a spatial representation $\mathcal{S}^{t+\delta t}$, extracted at a later instant $(t + \delta t)$. Such a

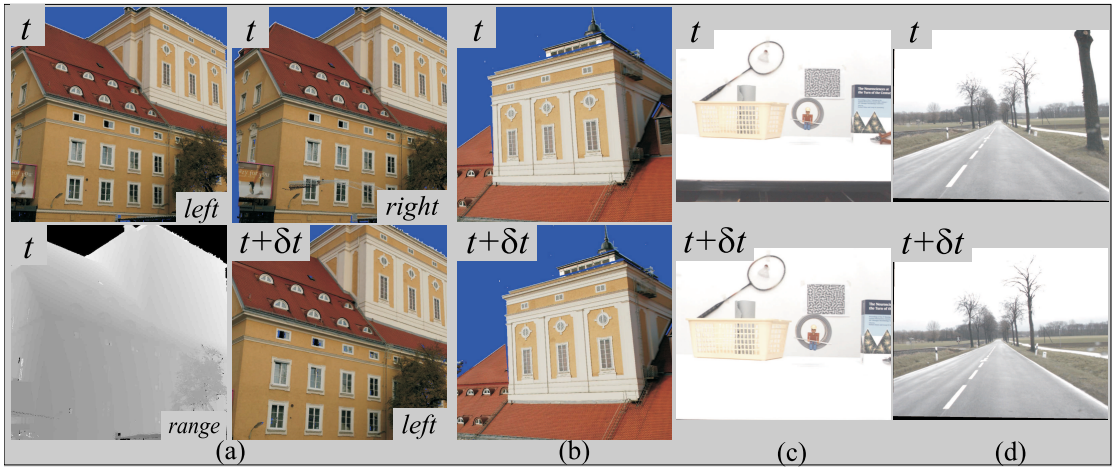


Figure 6.1: Four video sequences that were used to test the ego–motion estimation. For sequences (a) and (b), the disparity ground truth was obtained using a range scanner; the motion was (a) a pure translation, and (b) a pure rotation. For sequence (c), the motion was carefully measured, and was a pure forward translation. Finally sequence (d) shows an outdoor scenario, where the car's motion was roughly estimated.

transformation can be done if the ego–motion between t and $t + \delta t$ is known.

Because we assume that the cameras' relative position and orientation are constant, this ego–motion belongs to a class of motion called Rigid Body Motions (RBMs). Hence, the problem we address in this chapter is: "how to compute this RBM from two stereo pairs of images ?". If we consider a moving camera in an otherwise static world, *e.g.*, a car driving on an empty road (see Fig. 6.1(d)), then the motion of the camera $\boldsymbol{M}_c$ is exactly the inverse of the apparent world motion $\boldsymbol{M}_w$:

$$\boldsymbol{M}_w = (\boldsymbol{M}_c)^{(-1)}. \tag{6.1}$$

One could object that, even in an empty road, wind in the leaves and clouds in the sky always add some kind of independent motion; yet, most spurious motions can be discarded processing at a coarse enough scale, and we will show that they can be treated as another source of noise for the purpose of ego–motion estimation. The relation (6.1) is useful because, although the camera's motion is generally unknown, the world's motion can be estimated from the differences between the stereo pairs of image representations $\left(\boldsymbol{\mathcal{I}}^{l,t}, \boldsymbol{\mathcal{I}}^{r,t}\right)$ and $\left(\boldsymbol{\mathcal{I}}^{l,t+\delta t}, \boldsymbol{\mathcal{I}}^{r,t+\delta t}\right)$ (in the following we will use the first exponent to identify the camera, the second to express the instant in time). In this chapter, we propose a method to estimate the ego–motion between two frames, making use of the image representation (introduced in chapter 2),

the stereo matching (chapter 4) and the reconstruction (section 4.4) presented earlier.

Note that if other moving objects are visible in the scene, those objects' motions can also be RBMs, independent from the ego–motion. Hence, they are generally referred to as independent moving objects (IMO). Provided that one can segment those objects from the rest of the images, it is theoretically possible to estimate their apparent motion in the same way as we estimate the apparent world motion. Their true motion can be computed as follows:

$$\tilde{\boldsymbol{K}}^{t+\delta t} = \boldsymbol{M}_w \cdot \boldsymbol{M}_o \cdot \tilde{\boldsymbol{K}}^{t} \tag{6.2}$$

where $\tilde{\boldsymbol{K}}$ represents the pose of the object at instants t and $t + \delta t$, $\boldsymbol{M}_o$ is the apparent motion of the object, and $\boldsymbol{M}_w$ is the apparent world motion (hence the inverse of the ego–motion) during the same time. Herein, we will focus on the ego–motion estimation without any prior image segmentation. Consequently the IMO (only in scene (d)) are treated as spurious motions and are neglected during the estimation of the ego–motion.

In appendix C, we define the geometric properties of a camera as a combination of its intrinsic parameters (defining the optic of the camera), and its extrinsic parameters (defining the *pose* of the camera). The pose of an entity codes its position and orientation, relative to the global coordinate system, or, alternatively, the RBM moving the global coordinate system onto the camera coordinate system, expressed as the pose matrix $\tilde{\boldsymbol{K}}^{t}$. Therefore, the transformation moving the cameras from the pose $\tilde{\boldsymbol{K}}^{t}$ to a later pose $\tilde{\boldsymbol{K}}^{t+\delta t}$ can be formalised as an RBM $\boldsymbol{M}_{t\to t+\delta t}$, such as:

$$\tilde{\boldsymbol{K}}^{t+\delta t} = \tilde{\boldsymbol{K}}^{t}(\boldsymbol{M}_{t\to t+\delta t})^{(-1)}, \tag{6.3}$$

where $(\boldsymbol{M}_{t\to t+\delta t})^{(-1)}$ is the inverse of the motion of the camera. Hence, we can also rewrite the projection matrix of the camera at time $t+\delta t$ function of its projection matrix at time t and the motion between these two instants as:

$$\tilde{\boldsymbol{P}}_{t+\delta t} = \tilde{\boldsymbol{P}}_t \boldsymbol{M}_{t\to t+\delta t} \tag{6.4}$$

This means that, provided that we have the pose information at every instant t, we can compare the visual information reconstructed at instants t and $t + \delta t$ using traditional geometry exposed in appendix C. Conversely, if we have the relative transformation between the pose of the camera at instants t and $t + \delta t$, we can compute the camera's pose at $t + \delta t$, in the coordinate system of the camera at time t. We will

present a way to use this information to generate a robust and reliable scene description in chapter 7.

The estimation of the ego–motion from camera images is an important but complex problem in computer vision (see, *e.g.*, (Faugeras, 1993; Hartley and Zisserman, 2000)). It requires addressing three sub-problems:

Correspondence problem: The constraints used to estimate the RBM are drawn from multiple correspondences between visual entities, over stereo and time. Hence, the constraint equations depend on the visual entities that provided those correspondences. Therefore, those correspondences should be as reliable and precise as possible.

Mathematical formulation: A RBM can be written, and estimated, using different mathematical formulations, *e.g.*, matrices (Faugeras, 1993), quaternions (Faugeras and Hébert, 1986) or dual quaternions (Phong et al., 1995; Shevlin, 1998). The precision and stability of those methods have been discussed by, *e.g.*, Lorusso et al. (1995).

Outliers: Although we would like the correspondences to be as reliable as possible, a significant subset of those correspondences is bound to be erroneous. Hence, methods are required to select the correct correspondences, and neglect spurious ones. A prominent example is RANSAC (Fischler and Bolles, 1981).

Those three problems are deeply intertwined. For example, the choice of a mathematical formalisation depends on the kind of constraint that can be drawn, which is directly linked to the type of entity chosen — see Fig. 6.2. The type of entity chosen directly impact on the reliability of the correspondences which can be found between those entities.

In chapters 4 and 5, we have seen that a stereo–correspondence between 2D–primitives can be found reliably, and that such a correspondence allows to reconstruct a position in space (a 3D–point). Therefore, the RBM estimation problem, in our case, reduces to the pose estimation problem: provided a (partial) 3D model of the scene, and one image (taken from a similar viewpoint) of this scene, we want to compute the pose of the camera that captured this image. Furthermore, as those 2D–primitives are fundamentally local line descriptors, we face the aperture problem, and we can only draw 3D–point/2D–line correspondences over time. In section 6.1, we will explain how we constrain the RBM from such correspondences. Then in section 6.2, we will describe how those correspondences are actually found. Section 6.4 will focus on

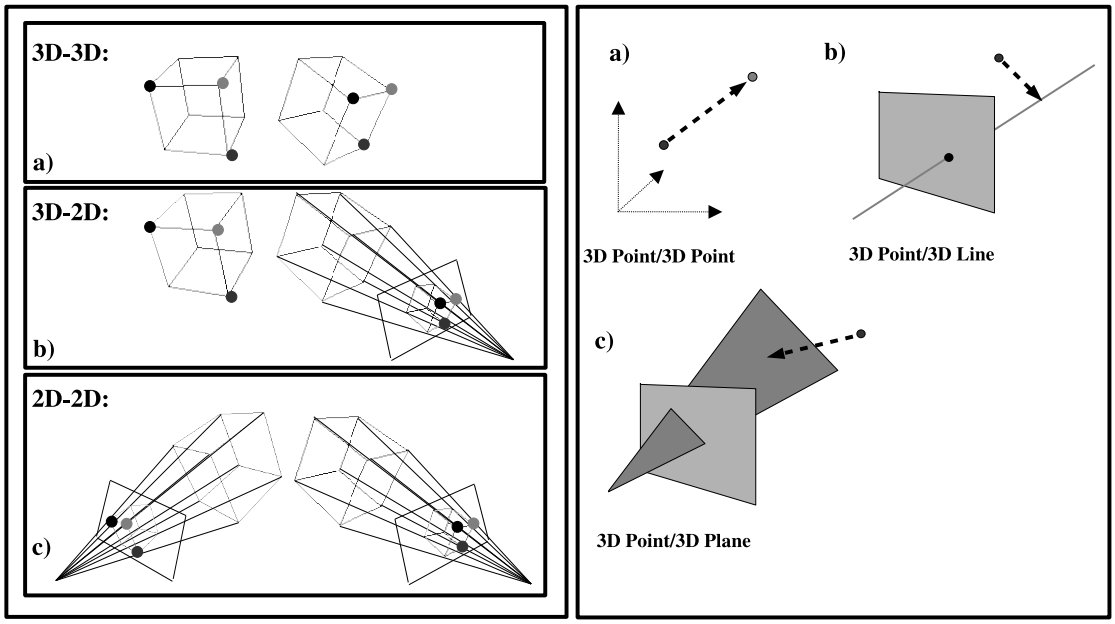


Figure 6.2: The different types of correspondence that can be used for the pose estimation. The left hand side of the figure illustrates the kind of correspondences that can be used, in terms of dimensionality: (a) 3D/3D, (b) 3D/2D, and (c) 2D/2D correspondences are The right hand side shows the different kinds of entities that can be matched, in (a), a 3D–point/3D–point correspondence provides 3 constraints; in (b), a 3D–point/3D–line correspondence provides 2 constraints); in (c), a 3D–point/3D–plane correspondence provides only one constraint. Figure reproduced from (Krüger and Wörgötter, 2004).

how to choose a good set out of this pool of correspondences, and the performance of this approach on our six working sequences is presented in section 6.5.

6.1 Mathematical framework and constraint equations

There are numerous formulations of the RBM estimation problem, using different mathematical formulations: matrices, quaternion, dual quaternions, twists, *etc.* In this chapter we use the formulation proposed by Rosenhahn et al. (2001b), which is based on an exponential formulation of the motion, called a *twist* (Bregler and Malik, 1998). Without entering into details, this formulation has several appreciable qualities (as outlined, *e.g.*, by Krüger and Wörgötter (2004)):

Searching the space of RBM: it leads to a set of equations acting directly on the RBM parameters.

Geometric interpretation: the constraint equation give measure in terms of Euclidean distances.

Mixing of different entities: this formulation allows for a conjoint use of 3D–point/2D–point correspondences, alongside 3D–point/2D–line correspondences. Although we are limiting ourselves to the latter in this work, it offers the possibility to include corner and junction features into the scheme, without changing the formulation.

We will only briefly introduce the main concepts here, and we refer the reader to (Rosenhahn, 2003; Rosenhahn et al., 2001a) for a detailed description of the algorithm and the constraint equations.

6.1.1 Twists formulation

We stated earlier that a RBM is the combination of a translation t and a rotation; the vector ω is the rotation's axis, and α the rotation's angle around this axis. We refer to (Bregler and Malik, 1998) for an explanation of the twist formulation; in short, the 6 parameters of a RBM can be written as a twist $\xi = (v_1, v_2, v_3, \omega_1, \omega_2, \omega_3)$, which matrix formulation is

$$\hat{\xi} = \begin{pmatrix} 0 & -\omega_3 & \omega_2 & v_1 \\ \omega_3 & 0 & -\omega_1 & v_2 \\ -\omega_2 & \omega_1 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{6.5}$$

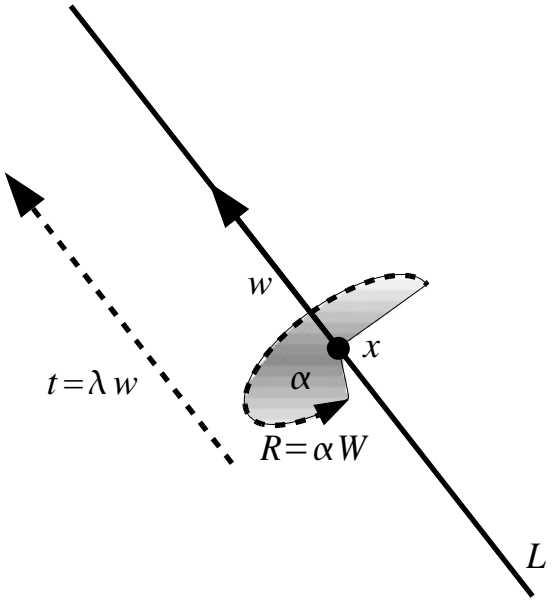


Figure 6.3: Twist formulation of the RBM. The motion is the combination of a rotation around the axis $\boldsymbol{w}$ of angle α, and a translation of $\lambda\boldsymbol{w}$.

According to Chasles theorem, any RBM can be expressed as the combination a rotation and a translation — see, *e.g.*, (Murray et al., 1994). The rotation of a point $\boldsymbol{x}$ is parametrised by its angle α and its axis: a line L defined in Plücker coordinates by its direction vector $\boldsymbol{w}$ and its moment $\boldsymbol{w} \times \boldsymbol{x}, \boldsymbol{x} \in L$ (*cf.* section B). This rotation is combined with a translation of a magnitude λ along the axis $\boldsymbol{w}$ (with $\boldsymbol{w}$ a unit vector such that $\|\boldsymbol{w}\| = 1$). This is illustrated in Fig. 6.3 We will write $\boldsymbol{W}$ the 3×3 matrix

$$\boldsymbol{W} = \begin{pmatrix} 0 & -w_3 & w_2 \\ w_3 & 0 & -w_1 \\ -w_2 & w_1 & 0 \end{pmatrix} \tag{6.6}$$

such as $\alpha\boldsymbol{W}$ stands for a rotation of an angle α around the axis $\boldsymbol{w}$. Then we call a twist the following 4×4

matrix:

$$\hat{\xi} = \begin{pmatrix} \boldsymbol{W} & -\boldsymbol{W}\boldsymbol{x} + \lambda \boldsymbol{w} \\ 0 & 0 \end{pmatrix} \tag{6.7}$$

$$= \begin{pmatrix} 0 & -w_3 & w_2 & w_3x_2 - w_2x_3 + \lambda w_1 \\ w_3 & 0 & -w_1 & w_1x_3 - w_3x_1 + \lambda w_2 \\ -w_2 & w_1 & 0 & w_2x_1 - w_1x_2 + \lambda w_3 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{6.8}$$

$$= \begin{pmatrix} 0 & -w_3 & w_2 & v_1 \\ w_3 & 0 & -w_1 & v_2 \\ -w_2 & w_1 & 0 & v_3 \\ 0 & 0 & 0 & 0 \end{pmatrix} \tag{6.9}$$

with

$$\boldsymbol{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} w_3x_2 - w_2x_3 + \lambda w_1 \\ w_1x_3 - w_3x_1 + \lambda w_2 \\ w_2x_1 - w_1x_2 + \lambda w_3 \end{pmatrix} \tag{6.10}$$

The matrix formulation of the RBM can then be obtained by computing the exponential of the twist matrix, which can be simply done by using the Taylor expansion:

$$RBM = e^{\alpha\hat{\xi}} = \sum_{n=0}^{\infty} \frac{1}{n!}(\alpha\hat{\xi})^n = I_{4\times4} + \alpha\hat{\xi} + \frac{\alpha\hat{\xi}^2}{2!} + \frac{\alpha\hat{\xi}^3}{3!} \cdots \approx I_{4\times4} + \alpha\hat{\xi} \tag{6.11}$$

Consequently, this 4×4 matrix is fully defined by the parameters α, $\boldsymbol{w}$ and $\boldsymbol{v}$. Because $\|\boldsymbol{w}\| = 1$, this yields exactly six parameters for six degrees of freedom, avoiding degenerate cases. In the following section, we will formalise constraints on those parameters.

6.1.2 3D–point/2D–line constraint

In this work we are interested in line structures and contours. Consequently the kind of features that we can match across frames are lines. The aperture problem makes it impossible to match a specific point along a line, in two different frames; yet, it is possible to match the lines. Hence, for any primitive $\boldsymbol{\Pi}$ on

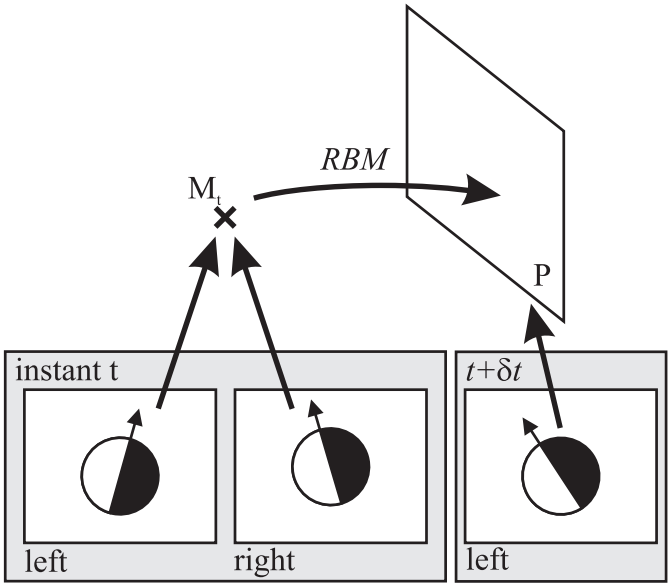


Figure 6.4: Illustration of the 3D–point/2D–line constraint.

a line L^t, reconstructed at time t, we can find the matching line $l^{t+\delta t}$ in the image representation extracted at time $t + \delta t$. This means, we will need to estimate the RBM from 3D–point/2D–line correspondences — each providing one effective constrain.

Consider the 3D–point $\boldsymbol{X}_t$ reconstructed by stereo from a pair of image representations $\boldsymbol{\mathcal{I}}^{l,t}$ and $\boldsymbol{\mathcal{I}}^{r,t}$, respectively produced by the left and right camera, at time t. The motion of this point between instants t and $t + \delta t$, is constrained such that it reprojects onto the line $l_i^{l,t+\delta t}$ — defined by the 2D–primitive $\boldsymbol{\pi}_i^{l,t+\delta t} \in \boldsymbol{\mathcal{I}}^{l,t+\delta t}$. The line $l_i^{l,t+\delta t}$ back–projects in space as the plane $P_i^{l,t+\delta t}$; therefore, the position $\boldsymbol{X}_{t+\delta t}$ is constrained to the plane $P_i^{l,t+\delta t}$ — see Fig. 6.4. If we now write the 3D point $\boldsymbol{X}^t = (X_1, X_2, X_3)$, and define $P_i^{l,t+\delta t}$ by its normal vector $(n_1, n_2, n_3)^T$ and its Hesse distance h to the origin (as proposed in appendix B), we can rewrite this constraint as follows (Rosenhahn et al., 2001a):

$$(n_1, n_2, n_3, -h) \cdot \left(I + \alpha\hat{\xi}\right) \cdot \boldsymbol{X}_t = 0 \tag{6.12}$$

$$(n_1, n_2, n_3, -h) \cdot \begin{pmatrix} 1 & -\alpha\omega_3 & \alpha\omega_2 & v_1 \\ \alpha\omega_3 & 1 & -\alpha\omega_1 & v_2 \\ -\alpha\omega_2 & \alpha\omega_1 & 1 & v_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \cdot (X_1, X_2, X_3, 1) = 0, \tag{6.13}$$

after reordering, we obtain the following equation:

$$\begin{pmatrix} n_1 \\ n_2 \\ n_3 \\ -n_3X_2 - n_2X_3 \\ -n_1X_3 - n_3X_1 \\ -n_2X_1 - n_1X_2 \end{pmatrix}^T \cdot \alpha \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = -h - n_1X_1 - n_2X_2 - n_3X_3. \tag{6.14}$$

In this formula, $\boldsymbol{v} = (v_1, v_2, v_3)$, $\boldsymbol{w} = (w_1, w_2, w_3)^T$ and α define the twist formulation of the RBM as in equation 6.9. A RBM contains 6 degrees of freedom (DOF); therefore, we need at least 6 constraints in order to estimate a RBM. From equation 6.14 we see that each 3D–point/2D–line correspondence yields one single constraint; hence, we need at least 6 of those correspondences in order to adequately define the motion estimation problem.

In this work we will draw such constraints from correspondences between a 3D–primitive $\boldsymbol{\Pi}_i \in \boldsymbol{\mathcal{S}}^t$ and a 2D–primitive $\boldsymbol{\pi}_j^f \in \boldsymbol{\mathcal{I}}^{f,t+\delta t},\ f \in l, r$, that are designed by the tuples $\left(\boldsymbol{\Pi}_i, \boldsymbol{\pi}_j^f\right)$.

6.1.3 Weighting of correspondences

One drawback of considering 3D Euclidian distances, is that 3D–primitives farther away from the camera lead to larger errors. This is due to the inaccuracy of the stereo–reconstruction of far objects. This means that constraints based on far (hence inaccurate) structures will tend to dominate the optimisation. In order to compensate bias, Wettegren et al. (2005) proposed to weigh the constraints according to their proximity. This allows to strengthen the influence of proximate (and therefore accurate) correspondences.

6.2 Finding correspondences

In the previous section we presented a framework for estimating the RBM from a set of correspondences $(\boldsymbol{\Pi}, \boldsymbol{\pi})$ (at least 6) between 3D–primitives $\boldsymbol{\Pi} \in \boldsymbol{\mathcal{S}}^t$ (reconstructed at time t) and 2D–primitives$\boldsymbol{\pi} \in \boldsymbol{\mathcal{I}}^{t+\delta t}$ (extracted from one of the stereo images captured at time $t + \delta t$). In this section we discuss how such correspondences are found.

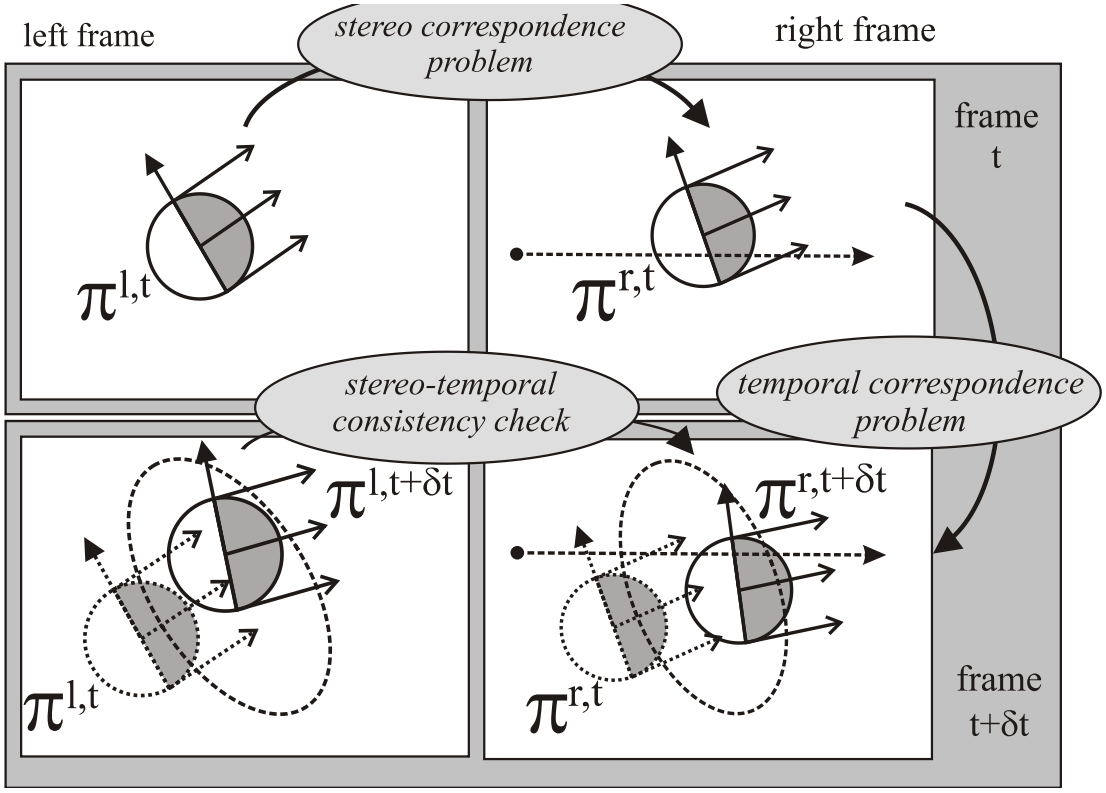


Figure 6.5: In order to estimate the RBM, one needs to address two correspondence problems. The first one is the stereo–correspondence problem, which has been addressed in the two previous chapters. The second is to find a correspondence between those primitives at another time frame. This is the temporal–correspondence problem, addressed in this chapter. One additional constraint, that the right match lie on the epipolar line of the left match, can be enforced. This ensures that the match defines a legal 3D–primitive at $t + \delta t$.

Effectively, in order to define a 3D–point/2D–line constraint, two separate matching operations need to be performed successfully (see Fig. 6.5). First, the classical stereo–matching is required to reconstruct a 3D–primitive — thus providing the 3D–point. This problem was discussed in chapters 4 and 5. Second, such a 3D–primitive $\mathbf{\Pi}^t \in \boldsymbol{\mathcal{S}}^t$ needs to be matched with a 2D–primitive $\boldsymbol{\pi}_j^{t+\delta t}$ providing the 2D–line from the image representation $\boldsymbol{\mathcal{I}}^{t+\delta t}$ extracted at a later instant $t+\delta t$ — thus providing the 2D–line. This section discusses the second problem.

We call $\boldsymbol{\pi}_i^t$ the i^{th} image primitive extracted at time t, and $\boldsymbol{x}_i^t$ its position in the image. A 2D–primitive's optical flow is an estimate of its image motion. Hence, we can infer an priori position $\hat{\boldsymbol{x}}_i^{t+\delta t}$ at an instant $t + \delta t$:

$$\hat{\boldsymbol{x}}_i^{t+\delta t} = \boldsymbol{x}_i^t + \delta t \boldsymbol{f}_i^t \tag{6.15}$$

Due to the noisiness of the optical flow information we do not expect this first estimate to be reliable. Moreover, it is unlikely that a 2D–primitive be extracted at this precise location at time $t + \delta t$, because of the 2D–primitives sparseness and the aperture problem. Therefore, we search nearby this position $\hat{\boldsymbol{x}}_i^{t+\delta t}$ for 2D–primitives that are similar to $\boldsymbol{\pi}_i^t$. [1] We search for a match within a radius of $r\delta t$ time the 2D–primitive's size λ. In our experiments setting $r = 10$ yields satisfying results.

To summarise, given a 2D–primitive $\boldsymbol{\pi}_i^t \in \boldsymbol{\mathcal{I}}^t$ at a position $\boldsymbol{x}_i^t$ and with an optical flow vector $\boldsymbol{f}_i^t$, a 2D–primitive $\boldsymbol{\pi}_j^{t+\delta t} \in \boldsymbol{\mathcal{I}}^{t+\delta t}$ at a position $\boldsymbol{x}_j^{t+\delta t}$, is considered a potential temporal–correspondence of $\boldsymbol{\pi}_i^t$ iff.

$$\begin{cases} d_E(\boldsymbol{x}_j^{t+\delta t}, \boldsymbol{x}_i^t + \boldsymbol{f}_i^t) & < & r\lambda\delta t \\ d_m(\boldsymbol{\pi}_i^t, \boldsymbol{\pi}_j^{t+\delta t}) & < & \tau_m \end{cases}, \tag{6.16}$$

where τ_m is a quantity small enough to only allow correspondences between fairly similar primitives, d_E is the Euclidian distance, and d_m the multimodal distance, from equation (4.4) in chapter 4.

If several $\boldsymbol{\pi}_j^{t+\delta t}$ are satisfying those constraints, the one minimising $d_m(\boldsymbol{\pi}_j^{t+\delta t}, \boldsymbol{\pi}_j^t)$ is chosen (see also Fig. 6.5). Moreover, we enforce the additional constraint that the correspondence in the right image must lie on the left image correspondence's epipolar line: *i.e.*, those 2D–primitives allow for the reconstruction of a valid 3D–primitive at time $t + \delta t$, hence form plausible matches.

[1] In the sense of the multi–modal metric in equation (4.4) from chapter 4

6.3 Evaluation of the RBM quality

The matching scheme presented in the previous section is expected to produce a fairly large pool of correspondences (even when using very selective confidence thresholds). This pool of correspondences is hereafter called $\mathcal{Q} = \left\{\left(\mathbf{\Pi}_i, \boldsymbol{\pi}_j^f\right)\right\},\ f \in \{l, r\}$. We only need 3 of those data points (3 correspondences in the left image plus 3 correspondences in the right) to estimate a RBM; hence the problem is largely overconstrained. On the other hand, we face the problem that a certain quantity of those correspondences is expected to be erroneous. Furthermore, a certain inaccuracy in the stereo–reconstruction of the 3D–primitives is to be expected. Due to these reasons, the problem of motion estimation becomes to select, out of this large pool $\mathcal{Q}$, a subset of constraints $\mathcal{R} \subset \mathcal{Q}$ that generates an accurate ego–motion. In order to compare the quality of the motion computed from different subsets $\mathcal{R}$, we need an online measurement of this quality. In the following sections, we will first propose an evaluation of the quality of the estimated RBM using known ground truth (section 6.3.1), then present an online measure of RBM quality (section 6.3.2) and show that it is a good estimation of the ground truth error.

6.3.1 Evaluation using ground truth

In the following and when describing the inaccuracy of an ego–motion estimate compared to the known ground truth, we will refer to the angle between the directions of the true and the estimated translation vectors as the *heading* error ϵ_h. The *magnitude* error ϵ_m is the difference between the norm of the translations. Likewise, we call the *axis* error ϵ_a the angular error between true and estimated rotation axes, and the *angle* error ϵ_α the difference between true and estimated rotation's angle. If we consider an estimated RBM composed of a translation $\boldsymbol{t}$ and a rotation of α around an axis $\boldsymbol{r}$, while the true motion is a translation $\boldsymbol{t}^*$ and a rotation α^* around the axis $\boldsymbol{r}^*$, then the errors are:

$$\epsilon_h = \widehat{\boldsymbol{t}, \boldsymbol{t}^*} \tag{6.17}$$

$$\epsilon_m = |\|\boldsymbol{t}\| - \|\boldsymbol{t}^*\|| \tag{6.18}$$

$$\epsilon_a = \widehat{\boldsymbol{r}, \boldsymbol{r}^*} \tag{6.19}$$

$$\epsilon_\alpha = \arctan\left(\frac{\sin(\alpha - \alpha^*)}{\cos(\alpha - \alpha^*)}\right) \tag{6.20}$$

where $\widehat{\boldsymbol{a}, \boldsymbol{b}}$ is the angle between the two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.

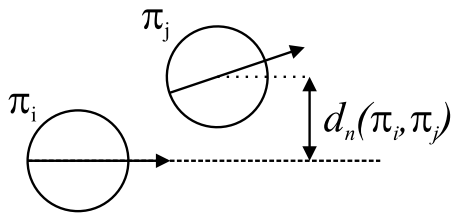


Figure 6.6: The normal distance $d_n(\boldsymbol{\pi}_i, \boldsymbol{\pi}_j)$ between $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$ is defined as the distance from the point $\boldsymbol{\pi}_i$ to the line defined by $\boldsymbol{\pi}_j$.

6.3.2 Online evaluation

In order to estimate the accuracy of an estimated motion $\boldsymbol{M}_{\mathcal{R}}$, computed from a subset of constraints $\mathcal{R}$, we move all 3D–primitives in the pool of correspondences $\mathcal{Q}$ according to $\boldsymbol{M}_{\mathcal{R}}$, and reproject them onto the image plane $I^{t+\delta t}$. If the motion $\boldsymbol{M}_{\mathcal{R}}$ is accurate, all correct 3D–primitives will reproject on the correspondences found in $\mathcal{I}^{t+\delta t}$ (assuming that these were correct matches). On the other hand, if the motion is incorrect, most of the 3D–primitives will be reprojected far from their correspondences. We propose to rate the quality of the estimated RBM as the mean difference between the reprojected and the matched positions.

Therefore, considering a constraint $\left(\boldsymbol{\Pi}_i, \boldsymbol{\pi}_j\right)$, we compare the position of the 2D–primitive $\hat{\boldsymbol{\pi}}_i$, (reprojected after moving $\boldsymbol{\Pi}_i$ according to the estimated motion) with the position of the actual correspondence $\boldsymbol{\pi}_j$. This comparison is made using the *normal distance*: as both primitives represent 2D lines, the normal distance $d_n(\hat{\boldsymbol{\pi}}_i, \boldsymbol{\pi}_j)$ between $\hat{\boldsymbol{\pi}}_i$ and $\boldsymbol{\pi}_j$ is defined as the distance from the point $\hat{\boldsymbol{\pi}}_i$ to the line defined by $\boldsymbol{\pi}_j$ (see Fig. 6.6). In this context, this distance is called the *deviation* of the correspondence $\left(\boldsymbol{\Pi}_i, \boldsymbol{\pi}_j\right)$, through the estimated motion. Hence the *mean deviation* $\langle \Delta_{\mathcal{R}} \rangle$ for a set of predictions $\mathcal{R}$ is:

$$\langle \Delta_{\mathcal{R}} \rangle = \sum_{i \in Q} \frac{d_n(\hat{\boldsymbol{\pi}}_i^l, \boldsymbol{\pi}_i^l) + d_n(\hat{\boldsymbol{\pi}}_i^r, \boldsymbol{\pi}_i^r)}{2\#(\mathcal{Q})} \tag{6.21}$$

where $\#(\mathcal{Q})$ stands for the cardinal function, namely: the number of elements in the set $\mathcal{Q}$.

Fig. 6.7 plots the estimated RBM's translation and rotation error, for sequences with ground truth (Fig. 6.1(a) and (b)); this figure shows a nearly linear relation between error and mean deviation, although this relation becomes less clear for sets with low errors. This shows that the mean deviation is a suitable online estimation of the real error.

In the following, we divided the pool of correspondences into two sets: the *generation set* $\mathcal{Q}_G$, from

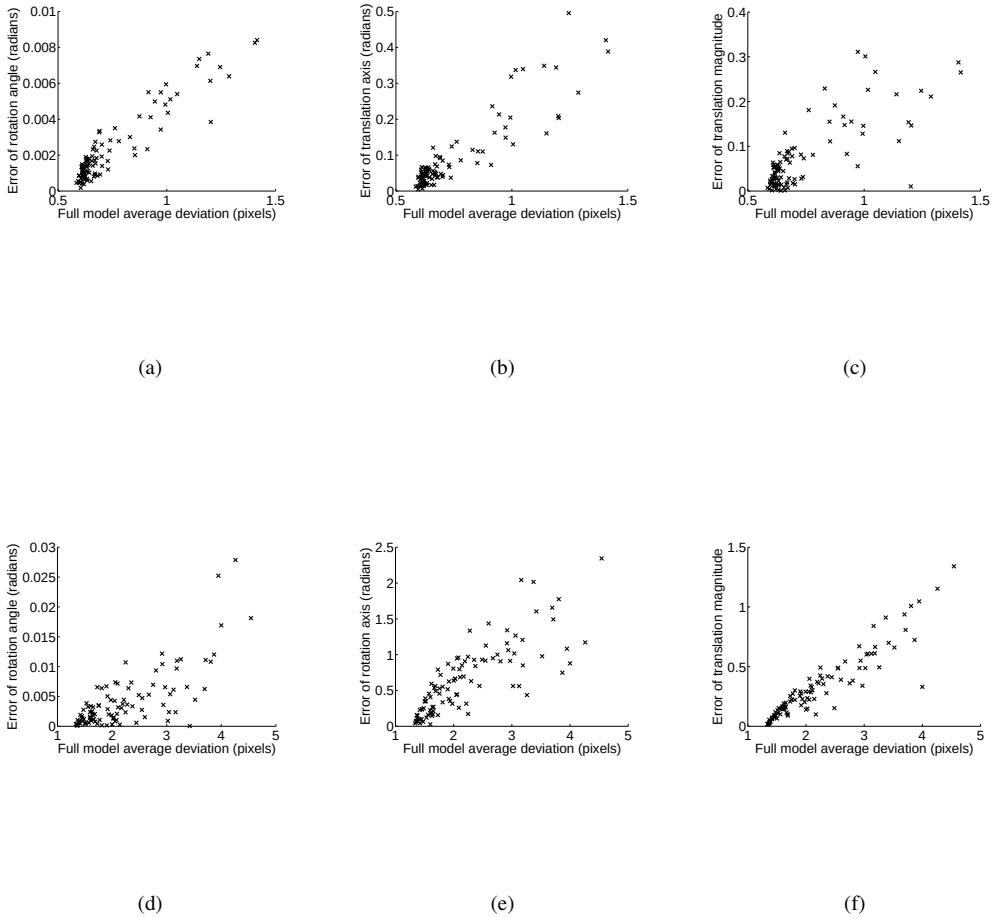


Figure 6.7: Accuracy of the RBM estimated plotted against the average deviation of the whole pool of correspondences. For the sequence 6.1(a) the true motion is a pure translation of vector $(0, 0, 2)^T$ — no rotation: 0 radians, and for the sequence 6.1(b) the true motion is a pure rotation of axis $(0, 1, 0)^T$ of 0.4 radians —no translation: $(0, 0, 0)^T$. The first row of graphs is the statistics of motion computed on sequence 6.1(a), while the second row shows motions computed on sequence 6.1(b). The horizontal axis shows the average deviation Δ and the vertical axis shows, for the sequence 6.1(a): 6.7(a) the rotation component error (the true rotation is 0 radians), 6.7(b) the angle between those two vectors the estimated translation and the true one, and 6.7(c) the difference in magnitude between those two translations, and for the sequence 6.1(b): 6.7(d) the rotation component error (the true rotation is 0 radians), 6.7(e) the norm of the difference between the estimated rotation axis and the true one, and 6.7(f) the the difference between the magnitude of the estimated translation vector and the true one.

which the estimation sets $\mathcal{R}_i \subset \mathcal{Q}_G$ are formed; and the *validation set* $\mathcal{Q}_V$ that is used to compute the mean deviation and consensus in the quantification in section 6.5. Those sets form a partition of $\mathcal{Q}$:

$$\mathcal{Q}_G \cup \mathcal{Q}_V = \mathcal{Q} \tag{6.22}$$

$$\mathcal{Q}_G \cap \mathcal{Q}_V = \varnothing \tag{6.23}$$

6.4 Selecting adequate sets of correspondences

In section 6.2, we defined how to constitute a large pool of correspondences. Although those correspondences are generally reliable, a certain subset of them is expected to be inaccurate, or even plainly wrong. Also, different sets of correspondences will estimate the motion with a different quality (see Fig. 6.7 for an illustration of this variation). Consequently, the selection of a suitable set is a crucial step for the motion estimation task. In this section we will compare three strategies for selecting such a set of correspondences: random sets, growing sets and RANSAC.

6.4.1 Random sets

Because the pool of correspondences is very reliable, the proportion of wrong correspondences is expected to be small. Therefore, we propose to form N random sets $\mathcal{R}_i, i \in [1, N]$, of M correspondences each.

If we consider a pool $\mathcal{Q}$ of $\#(\mathcal{Q})$ correspondences, and assume that a subset $\mathcal{F} \in \mathcal{Q}$ of them are erroneous, then the likelihood to randomly pick a correct correspondence in $\mathcal{Q}$ is:

$$q = \frac{\#(\mathcal{Q})}{\#(\mathcal{F})} \tag{6.24}$$

It follows that the chance for a subset $\mathcal{R} \in \mathcal{Q}$ to contain only correct correspondences is:

$$p = q^{\#(\mathcal{R})} \tag{6.25}$$

Thus, if we randomly build a population $\{\mathcal{R}_i\}_{i\in[1,N]}$ of N sets, where N is such that $N \gg \frac{1}{p}$, there is a high likelihood that at least one of the sets contains *only* correct correspondences. Furthermore, this

correct set is expected to produce a lower mean deviation $\langle \Delta_G \rangle$ than the other sets, containing erroneous correspondences. Hence, if we select the set $\mathcal{R}_i$ with the lowest deviation $\langle \Delta_G \rangle$, out of a large enough population of randomly selected sets, it is most likely to be formed of only correct correspondences.

This scheme has been evaluated on several sequences (see Fig. 6.1), using a population of $N = 100$ different sets, containing correspondences randomly picked from the generating pool. The quality of the estimated ego–motion was evaluated for set size ranging between 3 and 20 correspondences. The performance of the set minimising the mean deviation $\langle \Delta_G \rangle$ of the generation set $\mathcal{Q}_G$ is drawn in the Fig. 6.8, 6.9, 6.10, and 6.11. In these graphs, the deviation and consensus values are calculated from the *validation* set $\mathcal{Q}_V$.

6.4.2 Dynamic growing of a set of correspondences

The drawback of extending the size of a randomly chosen set of correspondences, is that the likelihood for a set to contains only correct correspondences decreases exponentially with the size of the set (see equation (6.25)). On the other hand, the motion estimated from larger sets is expected to be less sensitive to the inaccuracy that stems from the stereo–reconstruction. One simple way to address this problem is to generate randomly a small set of correspondences and to progressively increase it such that the mean deviation of the larger set is always smaller than the mean deviation of the larger one. The algorithm can be outlined as follows:

1. We generate randomly a population $\{\mathcal{R}_{n,i}\}_{i\in[1,N]}$ of $N = 100$ small sets (n correspondences, with n small, see previous section).

2. The set $\mathcal{R}_{n,*}$ minimising the deviation $\langle \Delta_G \rangle_i$ (over the generation set $\mathcal{Q}_G$) is chosen, as before.

3. We create a new population $\{\mathcal{R}_{n+1,i}\}_{i\in[1,N]}$ of sets of size $n + 1$, such as $\mathcal{R}_{n,*} \subset \mathcal{R}_{n+1,i}\ \forall i \in [1, N]$.

4. back to step 2 until $\mathcal{R}$ reaches a pre–defined size m.

Thus, the likelihood for each iteration of the algorithm to find a correct correspondence stays constant for any size of subset (as long as $\#(\mathcal{R}) \ll \#(\mathcal{Q}_G)$. The results of this algorithm for starting sets ranging from 3 to 20 correspondences is illustrated in Fig. 6.8, 6.9, 6.10, and 6.11.

6.4.3 Random Sample Consensus (RANSAC)

RANSAC is an algorithm proposed by Fischler and Bolles (1981) to select efficient sets of constraints from an unreliable pool. The problem of incorrect data points is that they generate large deviations, even when the motion is correct, and thus make the average deviation a very noisy measure (as the impact of erroneous data points is several orders of magnitude stronger than the impact of the correct points). RANSAC proposes to address this problem by only considering datapoints that have a deviation within a certain tolerance. Those datapoints are called the *consensus* for the solution. The algorithm relies on the assumption that wrong solutions will lead to small consensus. The algorithm is as follows:

1. First select a random set of n primitives $\mathcal{R}$ (n being a small number, close to the minimal set size).
2. Estimate the RBM from $\mathcal{R}$.
3. All correspondences from $\mathcal{Q}_G$ with a deviation lower than a certain threshold (in our case, set to λ pixels) are part of a new subset $\mathcal{R}^*$ called the consensus of $\mathcal{R}$.
4. If the *consensus ratio* is such that $\frac{\#(\mathcal{R}^*)}{\#(\mathcal{Q}_G)} \geq t_c$, with t_c the minimal consensus allowed, then the set $\mathcal{R}^*$ is chosen; otherwise we go back to step 1.
5. the RBM is recomputed from the whole consensus $\mathcal{R}^*$.

In our experiments we set $t_c = 0.3$. A larger value would make the algorithm more selective. This is not critical in our case, as we ran RANSAC 10 times, then choose the RBM with the largest consensus. This allows to compute the RBM from a very large set (in our case at least 30% of the pool) while excluding erroneous correspondences.

6.5 Results and discussion

The first strategy rely on having a pool of reliable correspondences, such that the likelihood to obtain a correct set is high enough (significantly higher than 1% for the process to be reliable). The larger the generative set, the more accurate and robust to noise is the process. Yet the likelihood to include an incorrect correspondence in the set grows quickly with the size of the set, especially when the pool of correspondences is unreliable. Consequently, the two other methods take advantage of the fact that it is

likely to obtain a small set of correct correspondences over a few trials. The growing method build up the size of a set, insuring to only include correct ones by only accepting correspondences decreasing $\langle \Delta_G \rangle$. RANSAC on the other hand tries to maximise a consensus, and then estimates the RBM for this whole consensus. Out of the 100 trials we take the largest consensus as the best estimation.

In order to evaluate the performance of this system, we applied the process to two video sequences for which the ground truth of the motion was known, as well as the 3D structure provided by a range scanner — see Fig. 6.1(a). Fig. 6.8, 6.9, 6.10, and 6.11 illustrate the performance of the different strategies for different sequences, ranging from very controlled semi–artificial scenes generated from range data (see Fig. 6.1(a) and (b)), to a real–life recorded sequence (Fig. 6.1(c)), and to uncontrolled complex outdoor driving scenario (Fig. 6.1(d)). Moreover, we also consider the mean deviation $\langle \Delta_V \rangle$ of the validation set Q_V. The *consensus* ratio is the ratio of correspondences in the validation set featuring a deviation lower than 3 pixels, and the consensus deviation is the mean of those deviations (hence the latter is always lower than 3). The ground truth of the motion is only known accurately for sequences (a) and (b), respectively a pure translation of 2 metres towards the positive z–axis and a pure rotation of 0.2 radians around the vertical y–axis (to the left). For sequence (c), the motion was measured during the recording, and is a translation of 56.5mm towards the positive z–axis. The heading of this ground truth is not perfectly accurate, though. For sequences (a) (b) and (c), the graphs show the mean deviation of the validation set and the size (as a ratio of the validation set size) and the deviation of the consensus. The ground truth for the translation magnitude and heading errors as well as the rotation estimation error are also shown. For sequence (d), as we do not have any ground truth of the motion we only show the mean deviation with the size and 6.11(b)) and the deviation of the consensus.

Because the motion is estimated using correspondences drawn from images, its accuracy is limited by the precision of the pixel sampling, and depends on the projection operated by the cameras. In short, the pixel sampling creates an inaccuracy in the 3D position estimations that is proportional to the distance of the 3D point to the camera — see (Faugeras, 1993) for a mathematical demonstration. Therefore, the actual accuracy varies depending on the image resolution, distance of the structures, *etc.* Consequently, we need an alternative evaluation of the quality of the estimated motion that is not related to the absolute 3D values, and as such subject to this reconstruction error. We use the overall deviation $\langle \Delta_Q \rangle$, mentionned earlier (equation 6.21) to estimate how accurately the estimated 3D motion allows us to predict the motion flow of 2D–primitives. This deviation is strongly affected by false correspondences, and therefore

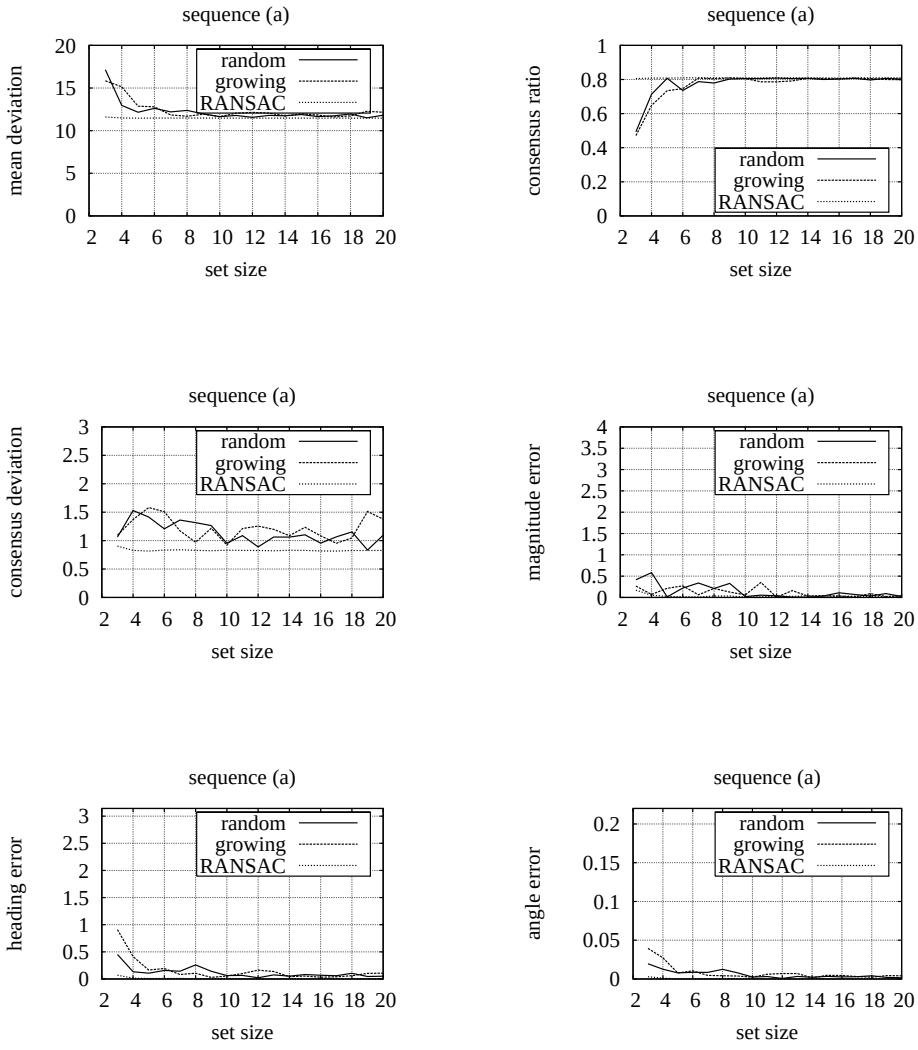


Figure 6.8: Comparative results for random sets, growing sets, and RANSAC for sequence (a). As this sequence has been generated from range scanner data, we have the exact ground truth for the motion, in this case a pure forward translation of 2 meters.

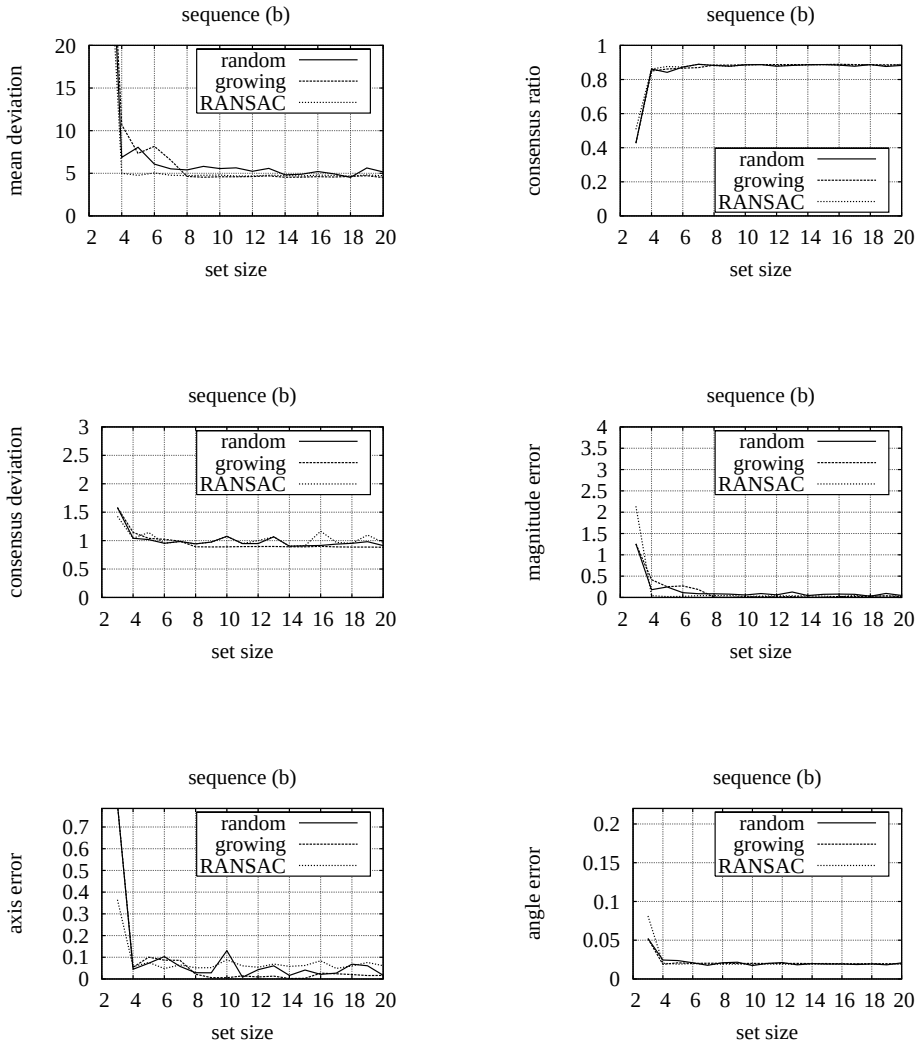


Figure 6.9: Comparative results for random sets, growing sets, and RANSAC for sequence (b). As this sequence has been generated from range scanner data, we have the exact ground truth for the motion, in this case a pure rotation around the y axis of 0.2 radians.

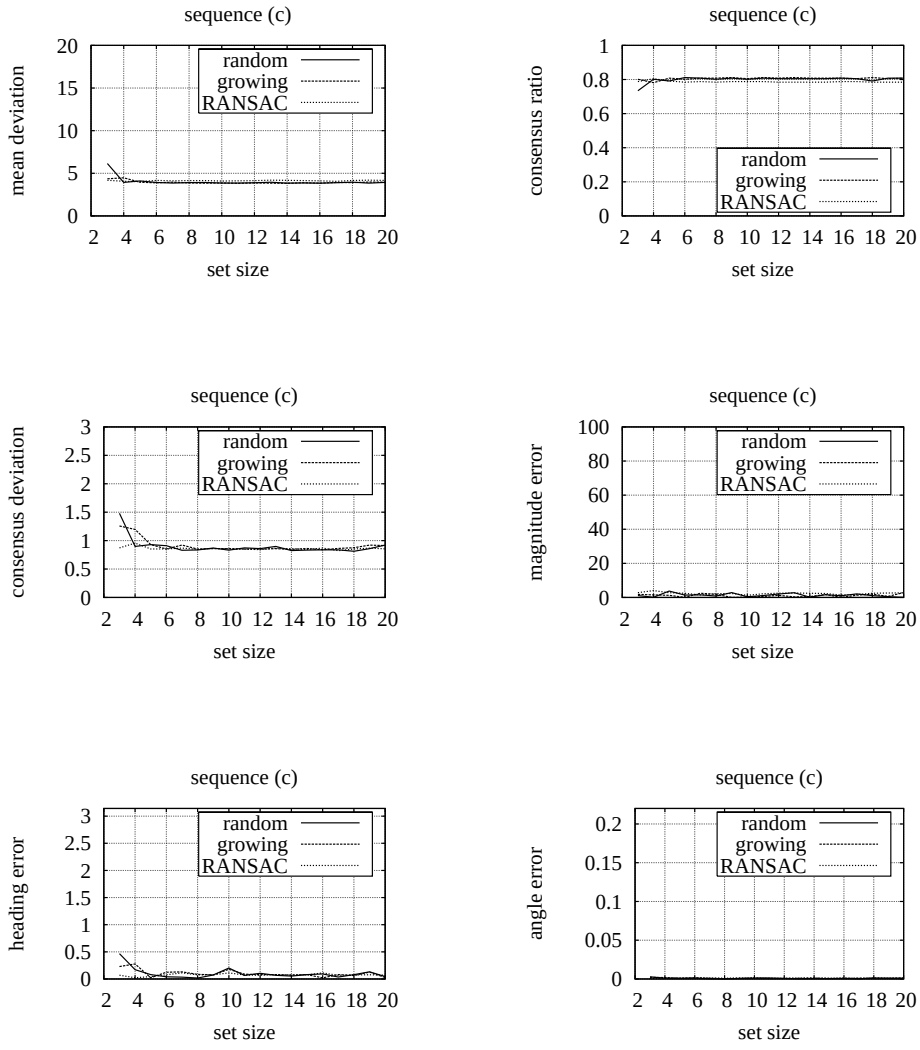


Figure 6.10: Comparative results for random sets, growing sets, and RANSAC for sequence (c). Note that for this sequence, the motion has been measured during the recording as a pure forward translation of a magnitude of 56.5mm.

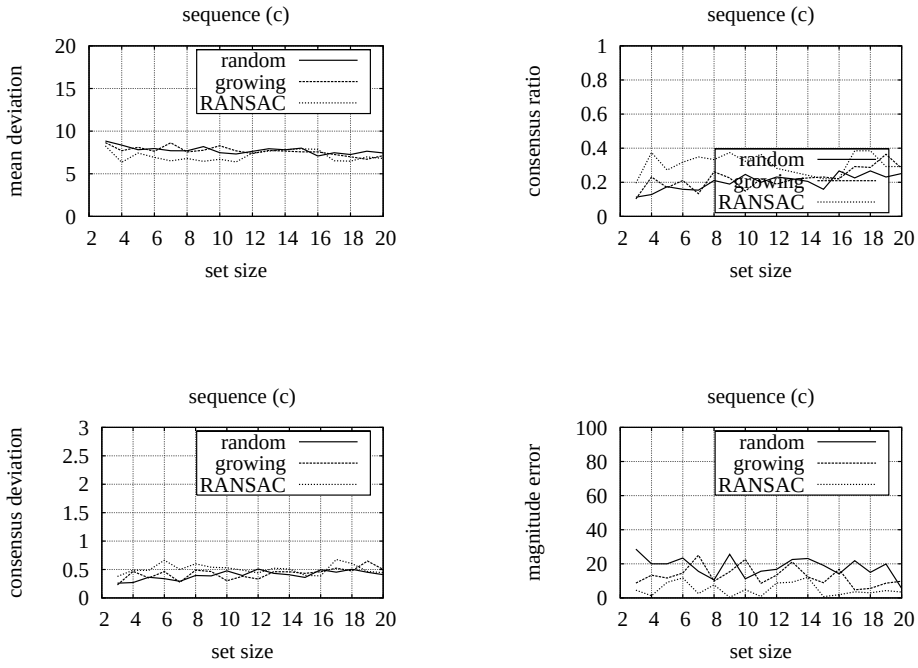


Figure 6.11: Comparative results for random sets, growing sets, and RANSAC for sequence (d). We do not have a ground truth for this sequence hence this figure only shows the deviation and consensus values.

yields a pessimistic evaluation of the RBM's quality. Moreover, the consensus for sets of at least 4 correspondences reaches 85%, 86%, and 50%, if we allow for a maximal deviation of 3 pixels. The deviation of this consensus $\langle \Delta_{\mathcal{R}^*} \rangle$, is generally lower than $\frac{t_c}{2} = 1.5$ — see graphs 6.8(c), 6.9(c), 6.10(c), and 6.11(c). Note also that we deal with high resolution images (1024x1024 for (a) and (b), and 1276x1016 for (c) and (d)), and that on these sequences, an RBM estimation algorithm using manually determined correspondences resulted into mean pixel deviations $\langle \Delta_Q \rangle$ around 2 pixel. We will show in section 7 that this deviation is small enough to allow for a robust tracking of the primitives over time.

Chapter 7

Accumulation of 3D information over time

Vision without action is a daydream. Action without vision is a nightmare.

- Japanese proverb

As was discussed in the previous chapter, the knowledge of ego–motion, and objects' motions, allows to predict the visual stimulus at a later instant.

Although there exists a wealth of literature on shape from stereo or motion, few studies combine stereopsis with temporal context. The geometrical aspect of the prediction of a third frame from two stereo views was discussed by Faugeras and Robert (1996). Mandelbaum et al. (1999) iteratively refined the ego–motion and structure estimates, in a pyramidal approach. Tao et al. (2001) represented the scene using piecewise planar patches, segmented by colour. Predictions are obtained by an incremental warping of these regions. Strecha and van Gool (2002) compared the depth estimates obtained from dense stereo and shape–from–motion algorithms. Zhang et al. (2003) proposed a spacetime dense stereo algorithm that is using a spatio–temporal window for matching. Their algorithm estimates scene structure without an explicit computation of the inter–frame motion, assuming that no motion discontinuity occur.

In this chapter, we propose a framework that makes use of the spatial representation $\boldsymbol{\mathcal{S}}^t$ extracted at instant t, and of the knowledge of the motion $\boldsymbol{M}_{t\rightarrow t+\delta t}$ (ego– or object–) between instants t and $t+\delta t$ in order to predict a spatial representation $\widehat{\boldsymbol{\mathcal{S}}}^{t+\delta t}$ at instant $t+\delta t$ — see section 7.1. This predicted

representation $\widehat{\mathcal{S}}^{t+\delta t}$ can then be compared to the reconstructed representation $\mathcal{S}^{t+\delta t}$; This is discussed in section 7.2. These predictions are then used to:

1. disambiguate the spatial representation generated from stereo–reconstruction (see chapter 4), and discard erroneous 3D–primitives in the representation (see section 7.4);
2. use the motion knowledge to merge $2\frac{1}{2}D$ representations of the scene, and of objects, obtained from different viewpoints (see section 7.3); and
3. to segment manipulated objects from the background using the known motion.

The last item is of particular importance in a robotic context: in such a context, the system can gain control over an object by grasping it, and therefore controls the independent motion of this object. This is a very strong cue for segmenting the scene representation obtained from stereo, and provides the first semantic mean for the birth of an object: an object is defined, in $\mathcal{S}$, as a rigid subset of 3D–primitives, such that when one is moved (*e.g.*, because the robot grasps it), all others move consistently. This definition is convenient because it applies very broadly to the robotic domain (deformable objects are rare, and difficult to interact with), and because it does not require any specific assumption. In order for this definition to be true, one need to discard the 3D–primitives that describe the robot's hand, as they will move with the same motion as the object. This is discussed in section 7.5. Some results of this application are shown in section 7.6, and (Pugeault et al., 2007a).

7.1 Making predictions from known motion

If we consider a 3D–primitive $\mathbf{\Pi}_i^t \in \mathcal{S}^t$ that is part of the scene representation extracted at an instant t (according to the previous chapters), and assuming that we know the objects' motion between the instants t and $t + \delta t$, then we can predict the 3D–primitive's position in the new coordinate system of the camera at $t + \delta t$.

We can therefore predict a scene representation $\mathcal{S}^{t+\delta t}$ by moving the anterior scene representation ($\mathcal{S}^t$) according to the motion estimated between instants t and $t+\delta t$ (ego–motion, object motion, or both). The mapping

$$\mathcal{M}_{t\rightarrow t+\delta t} : \mathcal{S}_t \rightarrow \mathcal{S}_{t+\delta t} \tag{7.1}$$

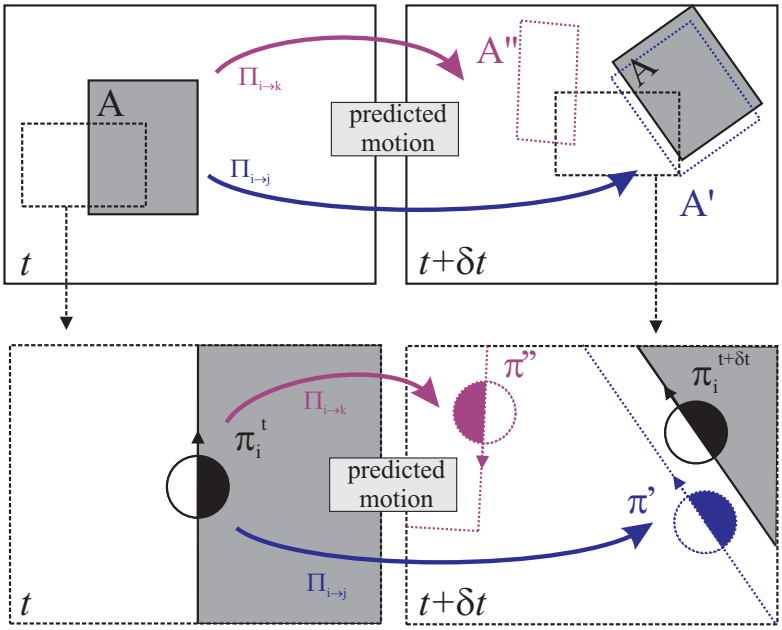


Figure 7.1: Example of the accumulation of a primitive (see text).

associates points in the camera's coordinate system at time t to points in the new coordinate system at time $t + \delta t$. We define this mapping explicitly for 3D–primitives:

$$\widehat{\mathbf{\Pi}}_i^{t+\delta t} = \mathcal{M}_{t \to t+\delta t}(\mathbf{\Pi}_i^t) \tag{7.2}$$

Assuming a scene representation $\mathcal{S}^t$ is correct, and that the motion between two instants t and $t + \delta t$ is known, the representation $\widehat{\mathcal{S}}^{t+\delta t}$, moved according to the motion $\mathcal{M}_{t \to t+\delta t}$, is a *predictor* for the scene representation $\mathcal{S}^{t+\delta t}$. Note that the predicted representation $\widehat{\mathcal{S}}^{t+\delta t}$ stems from 3D–primitives reconstructed from the images captured by the cameras at time t, whereas the actual scene representation $\mathcal{S}^{t+\delta t}$ is directly reconstructed from images provided by the cameras at time $t + \delta t$.

By extension, this relation also applies to the image representations re–projected onto each image planes $\mathcal{I}^f$, $f \in$ {left,right} (defined by a projection $\mathcal{P}^f$):

$$\widehat{\boldsymbol{\pi}}_i^{f,t+\delta t} = \mathcal{P}^f \left(\mathcal{M}_{t \to t+\delta t}(\mathbf{\Pi}_i^t) \right) \tag{7.3}$$

This prediction/verification process is illustrated in Fig. 7.1. The left column shows the image at time t,

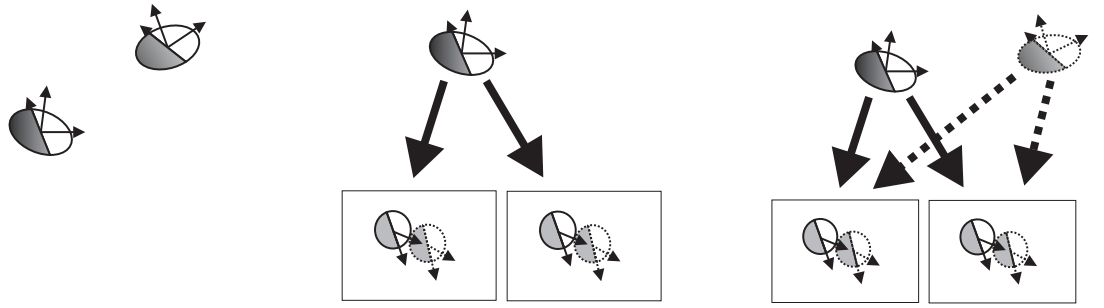
the right column at time $t+\delta t$. The top row shows the complete image of the object, the bottom row shows details of the object (of the area specified by the black rectangle). Consider the object **A** (solid rectangle in the top–left and top–right images) that moves according to a motion $M_{t\rightarrow t+\delta t}$, between the instants t and $t+\delta t$. Different hypotheses on the object's 3D shape lead to distinct predictions at time $t+\delta t$: **A'** (correct and close to the actual pose of the object, blue rectangle in the top–right image) and **A"** (erroneous, red rectangle). In the bottom row, we study the case of a specific 2D–primitive $\boldsymbol{\pi}_i^t$ lying on the contour of **A** at the instant t (bottom–left image). Consider two plausible, mutually exclusive, stereo–correspondences $\boldsymbol{\pi}_j^t$ and $\boldsymbol{\pi}_k^t$ for this 2D–primitive at time t. They lead to two mutually exclusive 3D reconstructions $\boldsymbol{\Pi}_{i\rightarrow j}^t$ and $\boldsymbol{\Pi}_{i\rightarrow k}^t$, each predicting a different pose at time $t+\delta t$: 1) the correct hypothesis $\boldsymbol{\Pi}_{i\rightarrow j}^t$ predicts a 2D–primitive $\boldsymbol{\pi}'$ that matches with $\boldsymbol{\pi}_i^{t+\delta t}$ (blue in the bottom–right image), one of the a 2D–primitive newly extracted at $t+\delta t$ from the contour of **A**, comforting the original hypothesis; 2) the incorrect hypothesis $\boldsymbol{\Pi}_{i\rightarrow k}^t$ predicts a 2D–primitive $\boldsymbol{\pi}''$ (red in the bottom–right image), that does not match any of the extracted 2D–primitives; therefore, the hypothesis is contradicted by this new information.

Differences in viewpoint and pixel sampling lead to large variations in the extracted 2D–primitives, the stereopsis, and the resulting 3D–primitives. In other words, scene contours will always be described in the image representation (apart from cases of occlusion), but by a different set of 2D–primitives, sampled at different points along these contours. Therefore, we need a tracking algorithm able to recognise similar structures described by different sets of descriptors.[1]

The robot we use (Staubli RX60) allows very precise motion, and therefore arm's motion is assumed to be known. 2D–primitives' position and orientation are commonly represented in the camera coordinate system (placed in the left camera) while the robot movements are relative to the robot's coordinate system (for the RX60 this is located at its first joint). To compute the mapping between the two coordinate systems we use a calibration procedure in which the robot's end effector is moved to the eight vertices of a virtual cube. At each location, the the end effector's position in both coordinate systems is recorded. This gives an over–determined system of linear equations that we solve to obtain the hand–eye calibration. Because the robot's arm is so precise, this calibration can be assumed to be constant.

In the following, we will make use of this known motion to integrate reconstructed visual information over time. The problem is three–fold: 1) comparing the two representations (section 7.2), 2) including

[1] The prediction described in this section is inaccurate for a specific class of occlusions that occurs when round surfaces become rotated (limb boundaries). In these cases the 2D–primitives reconstructed from the object's silhouette do not move according to an RBM.



a) 3D comparison b) 2D-perspective comparison c) 2D-stereo comparison

Figure 7.2: Illustration of the different matching domains — see text.

the extracted primitives that were not predicted (section 7.3), and 3) re–evaluating the confidence in each of the primitives according to their predictability (section 7.4).

7.2 Tracking 3D-primitives over time, and confidence re–assertion

In this section we address the problem of comparing two different scene representations, one extracted and one predicted, that both describe the same scene at the same instant from the same point of view. Because the 2D–primitives are sampled and condensed, the two scene representations are not expected to be identical but to describe the same scene contours. There are 3 ways to handle the problem, which we are going to compare before settling for the most appropriate.

7.2.1 3D comparison

The most intuitive way to compare the two representations is to compare all 3D–primitives from both representations in the 3D space. This is illustrated in Fig. 7.2(a). The spatial representation reconstructed $\boldsymbol{\mathcal{S}}^t$, and predicted $\widehat{\boldsymbol{\mathcal{S}}}^t$, are naturally very sparse. Hence, it is unlikely that a pair of 3D–primitives $\boldsymbol{\Pi}$ and $\widehat{\boldsymbol{\Pi}}$ match if they do not describe the same 3D–contour. This makes the matches found between the two representations very reliable. On the other hand, 3D–primitives' position accuracy degrades quickly with the depth, due to the limits of stereo reconstruction — see section 4.4 and (Pugeault et al., 2006b). This means that the notion of proximity between two 3D–primitives, required for matching, is ill defined. For this reason, we will compare 3D–primitives in the image plane, making use of the re–projection formulae

described in section 4.5.

7.2.2 2D comparison

In section 4.5 we explained how 3D–primitives can be re–projected onto the image planes, therefore allowing a duality between 3D–primitives and stereo–pairs of 2D–primitives.

We propose to use this re–projection relation to compare the predicted spatial representation $\widehat{\mathcal{S}}^t$, in the image plane, with the image representations $\mathcal{I}^{l,t}, \mathcal{I}^{r,t}$, extracted from the left and right images, respectively — see Fig. 7.2b). This comparison is fully described in section 7.2.4. There are some advantages to use such a 2D comparison: 1) because we compare the primitives in the image plane, we are not affected by 3D–reconstruction inaccuracy — see also (Wolff, 1989); 2) 2D–primitives that were not reconstructed at time t (for example because of their orientation) can nevertheless be matched. On the down side, there is no guarantee that the 2D–primitives matched in the left and right image representations form a valid stereo–pair (and therefore design a valid 3D–primitive). This can lead to a significant quantity of false matches, especially in images where 2D–primitivesextraction is dense. A second severe drawback is that, because the extracted 2D–primitives are not considered as 3D–entities, there is no reliable way to add 3D–primitives that were not extracted previously to the representation. We therefore propose to use the 2D–stereo comparison presented below.

7.2.3 Stereo comparison

This approach synthesizes qualities from both 2D and 3D comparisons presented above. We compare both 3D scene representations, predicted $\widehat{\mathcal{S}}^t$ and extracted $\mathcal{S}^t$, by re–projecting *both* of them onto both (left and right) image planes — see Fig. 7.2c). Then the stereo–pairs of 2D–primitivesre–projected from $\widehat{\mathcal{S}}^t$ and $\mathcal{S}^t$ are matched, in both image planes, as will be defined in section 7.2.4.

First, we avoid problems stemming from 3D reconstruction inaccuracy, because the matching is done in the plane domain. Second, matching stereo–pairs is effectively equivalent to matching the 3D–primitives directly: because 3D space is sparse, we have few false matches. Third, because we match 3D–primitives, we can classify those in three groups:

- predicted 3D–primitives that are matched in the new spatial representation, and hence which existence is *confirmed*.

- predicted 3D–primitives that are not matched in the new spatial representation, and hence which existence is *infirmed*.
- novel 3D–primitives that were not predicted from previous images.

In the following section we present a simple comparison algorithm to match predicted and extracted 2D–primitives in the image plane.

7.2.4 Matching 2D–primitives over time

The two representations are compared in the image plane. This can be done by reprojecting all the 3D–primitives in the predicted representation $\widehat{\mathcal{S}}^{t+\delta t}$ onto both image planes, creating two predicted image representations

$$\widehat{\mathcal{I}}^{f,t+\delta t} = \mathcal{P}^f\left(\widehat{\mathcal{S}}^{t+\delta t}\right),\ f \in \{l, r\} \tag{7.4}$$

Both predicted image representations $\widehat{\mathcal{I}}^{f,t+\delta t}$ can be compared with the extracted primitives $\mathcal{I}^{f,t+\delta t}$. For each predicted primitive $\widehat{\boldsymbol{\pi}}_i$, a small neighbourhood (the size of the primitive itself) is searched for an extracted primitive $\boldsymbol{\pi}_j$ whose position and orientation are very similar (with a distance less than a threshold t_θ).

Effectively a given prediction $\widehat{\boldsymbol{\Pi}}_i$ is labelled as matched $\mu\left(\widehat{\boldsymbol{\Pi}}_i\right)$ iff. for each image plane f defined by the projection $\mathcal{P}^f$ and having an associated image representation $\mathcal{I}^{f,t}$, we have the projection $\boldsymbol{\pi}_i^f = \mathcal{P}^f(\boldsymbol{\Pi}_i)$ that satisfy the following relation:

$$\exists \boldsymbol{\pi}_j \in \mathcal{I}^{f,t},\ \begin{cases} d_{2\mathrm{D}}(\widehat{\boldsymbol{\pi}}_i^f, \boldsymbol{\pi}_j) & < & r\lambda, \\ d_{\boldsymbol{\Theta}}(\widehat{\boldsymbol{\pi}}_i^f, \boldsymbol{\pi}_j) & < & t_{\boldsymbol{\Theta}} \end{cases} \tag{7.5}$$

with r being the radius of correspondence search in pixels (set to 1), $t_{\boldsymbol{\Theta}}$ the maximal orientation error allowed for matching, $d_{2\mathrm{D}}$ stands for the two–dimensional Euclidian distance, and $d_{\boldsymbol{\Theta}}$ is the orientation distance. This is also illustrated in Fig. 7.1.

7.3 Integration of different scene representations

Given two scene representations, one extracted $\mathcal{S}^t$ and one predicted $\widetilde{\mathcal{S}}^t$ we want to merge them into an accumulated representation $\mathcal{A}^t$. The application of the tracking procedure presented in section 7.2.2 provides a separation of the 3D–primitives in $\mathcal{S}^t$ into three groups: confirmed, unconfirmed and not predicted. The integration process consists into adding to the accumulated representation $\mathcal{A}^{t-1}$, all 3D–primitives extracted from the scene representation $\mathcal{S}^t$ that are not matched by any 3D–primitive in $\mathcal{A}^{t-1}$ (*i.e.*, the non–predicted ones).

$$\mathcal{A}^t = \mathcal{A}^{t-1} \cup \mathcal{S}^t \tag{7.6}$$

$$\mathcal{A}^0 = \varnothing \tag{7.7}$$

Therefore, the accumulated representation always strictly includes the newly extracted representation ($\mathcal{S}^t \subseteq \mathcal{A}^t$), while including new information in the representation.

7.4 Confidence re–evaluation from tracking

The second mechanism allows to re–evaluate the confidence in the 3D–hypotheses depending on their resilience. This is justified by the continuity assumption, which states that: 1) any given object or contour of the scene should not appear and disappear in and out of the field of view (FoV) but move gradually, according to the estimated ego–motion; and 2) the position and orientation of such a contour at any point in time is fully defined by the knowledge of its position at a previous point in time and its motion between these two instants.

Because the ego–motion is known and we disregard IMOs, all prerequisites are satisfied and for tracking a contour's position, from the time of its extraction t to any later stage $t + \delta t$, and for predicting the instant of its disappearance from the FoV.

We write the fact that a primitive $\mathbf{\Pi}_i$ that predicts a primitive $\widehat{\mathbf{\Pi}}_i^t$ at time t is matched (as described above) as the binary value $\mu_t(\widehat{\mathbf{\Pi}}_i) \in 0, 1$. By extension, this primitive's tracking history, from its first

appearance at time 0 until time t, is written as the binary vector:

$$\boldsymbol{\mu}(\boldsymbol{\Pi}_i) = \left(\mu_t(\widehat{\boldsymbol{\Pi}}_i), \mu_{t-1}(\widehat{\boldsymbol{\Pi}}_i), \cdots, \mu_1(\widehat{\boldsymbol{\Pi}}_i)\right)^T \tag{7.8}$$

Thus, applying Bayes formula:

$$p\left[\boldsymbol{\Pi}_i|\boldsymbol{\mu}(\widehat{\boldsymbol{\Pi}}_i)\right] = \frac{p\left[\boldsymbol{\mu}(\widehat{\boldsymbol{\Pi}}_i)|\boldsymbol{\Pi}_i\right] p\left[\boldsymbol{\Pi}_i\right]}{p\left[\boldsymbol{\mu}(\widehat{\boldsymbol{\Pi}}_i)|\boldsymbol{\Pi}_i\right] p\left[\boldsymbol{\Pi}_i\right] + p\left[\boldsymbol{\mu}(\widehat{\boldsymbol{\Pi}}_i)|\overline{\boldsymbol{\Pi}}_i\right] p\left[\overline{\boldsymbol{\Pi}}_i\right]} \tag{7.9}$$

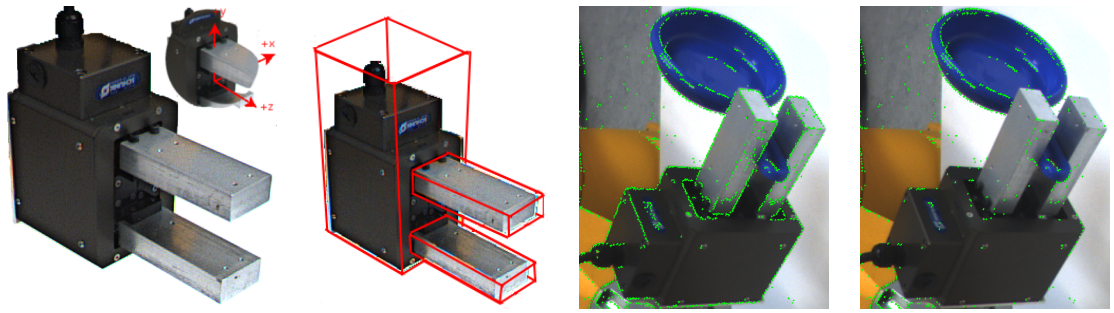
In this formula the prior likelihood for a reconstructed primitive to be correct is $p\left[\boldsymbol{\Pi}_i\right]$, to be erroneous is $p\left[\overline{\boldsymbol{\Pi}}_i\right]$. Furthermore, if we assume that the $\mu_t(\widehat{\boldsymbol{\Pi}}_i)$ are independent and that $\boldsymbol{\Pi}_i$ exists since n iterations, and has been successfully matched m times, we have:

$$\begin{aligned} p\left[\boldsymbol{\mu}(\widehat{\boldsymbol{\Pi}}_i)|\boldsymbol{\Pi}_i\right] &= \textstyle\prod_t p\left[\mu_t(\widehat{\boldsymbol{\Pi}}_i)|\boldsymbol{\Pi}_i\right] \\ &= p\left[\mu_t(\widehat{\boldsymbol{\Pi}}_i) = 1|\boldsymbol{\Pi}_i\right]^m p\left[\mu_t(\widehat{\boldsymbol{\Pi}}_i) = 0|\boldsymbol{\Pi}_i\right]^{n-m} \end{aligned} \tag{7.10}$$

In this case the probabilities for μ_t are equiprobable for all t, and therefore we define the quantities $\alpha = p\left[\boldsymbol{\Pi}_i\right], \beta = p\left[\mu_t(\widehat{\boldsymbol{\Pi}}_i) = 1|\boldsymbol{\Pi}_i\right]$ and $\gamma = p\left[\mu_t(\widehat{\boldsymbol{\Pi}}_i) = 1|\overline{\boldsymbol{\Pi}}_i\right]$ then we can rewrite (7.9) as follows:

$$p\left[\boldsymbol{\Pi}_i|\overline{\boldsymbol{\mu}}(\widehat{\boldsymbol{\Pi}}_i)\right] = \frac{\beta^m(1-\beta)^{n-m}\alpha}{\beta^m(1-\beta)^{n-m}\alpha + \gamma^m(1-\gamma)^{n-m}(1-\alpha)} \tag{7.11}$$

We measured these prior and conditional probabilities using a video sequence with known motion and depth ground truth obtained via range scanner — see section 4. We found values of $\alpha = 0.46$, $\beta = 0.83$, and $\gamma = 0.41$. This means that, in these examples, the prior likelihood for a stereo hypothesis to be correct is 46%, the likelihood for a correct hypothesis to be confirmed is 83% whereas for an erroneous hypothesis it is 41%. These probabilities show that Bayesian inference can be used to identify correct correspondences from erroneous ones. To stabilise the process, we will only consider the n first frames after the appearance of a new 3D–primitive. After n frames, the confidence is fixed for good. If the confidence is deemed too low at this stage, the primitive is forgotten. During our experiments $n = 5$ proved to be a suitable value.



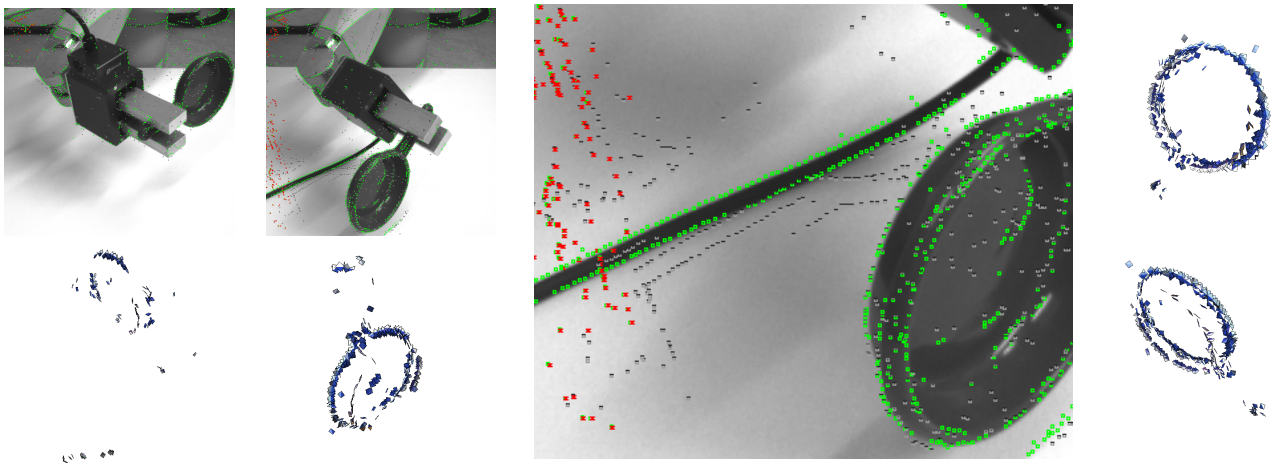
(a) (b) (c) (d)

Figure 7.3: Robot's hand elimination: (a) the robot's hand with its coordinate system; (b) bounding boxes envelopping the hand and its fingers; (c) predicted primitives before elimination (d) predicted primitives after elimination. In (c) and (d), green stands for matched predictions.

7.5 Eliminating the robot's hand

We want to apply this algorithm in a scenario where the robot has grasped an object (using, *e.g.*, the algorithm proposed by (Aarno et al., 2007)), and is inspecting it by moving it in front of the cameras. In this scenario, we want to use the visual information in conjunction with the precise knowledge of the robot's motions to learn a full–fledged 3D representation of the manipulated object. The algorithm presented in this chapter is adequate for such a task because: 1) it allows to refine the $2\frac{1}{2}D$ representation provided by stereo by discarding hypotheses that are not confirmed over time; 2) allows to segment the object from the scene by predicting the motion of the 3D–primitives with the known motion; and 3) provides a framework for integrating $2\frac{1}{2}D$ representations of the object (the so–called spatial representations $\boldsymbol{S}^t$) generated from different perspectives on the object.

However, the robot's hand and fingers follow the same motion as the object; therefore, motion would segment them with the manipulated object. To prevent this, we use our knowledge of the robot's hand geometry (Fig. 7.3(a)) to discard 3D–primitives that are within a bounding box envelopping the robot's hand and fingers — see Fig. 7.3(b) and (Pugeault et al., 2007a). Three bounding boxes are calculated in hand coordinate system (HCS) by using the dimensions of hand. Since the 3D primitives are in robot coordinate system (RCS), the transformation from RCS to HCS is applied to each primitive; if the resulting location is inside any of the bounding boxes, the primitive is eliminated. Fig. 7.3 shows the the locations of the 2D–primitives re–projected from predicted 3D–primitives, before (c), and after (d) elimination of the robot's hand.



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

Figure 7.4: Birth of an object (a)-(b) top: 2D projection of the accumulated 3D representation and newly introduced primitives, bottom: accumulated 3D representation. (c) detail of the newly introduced and accumulated primitives. Note that the primitives that are not updated are red and the ones that have low confidence are grey (d) final accumulated 3D representation from two different poses.

7.6 Results and discussion

We applied the accumulation scheme to a variety of scenes where the robot's arm manipulated several objects. The motion was a rotation of 5 degrees per frame. The accumulation process applied on one object is illustrated in Fig. 7.4. The top row shows the predictions at each frame. The bottom row, shows the 3D–primitives that were accumulated (frames 1, 12, 22, and 32). The object representation becomes fuller over time, whereas the primitives reconstructed from other parts of the scene are discarded. Figure 7.5 shows the accumulated representation for various objects. The hole in the model corresponds to the part of the object occluded by the robot's hand. Accumulating the representation over several distinct grasps of the objects would yield a complete representation.

A cognitive robot system should to be able to extract representations about its environment by exploration to enrich its internal representations and by this its cognitive abilities — see, *e.g.*, (Fitzpatrick and Metta, 2003; Fitzpatrick et al., 2003). The knowledge about the existence of objects and their shapes is of particular importance in this context. Having a 3D model of objects allows recognition and pose estimation (see, *e.g.*, (Lowe, 1987)) as well as grasp planning — *e.g.*, (Borst et al., 1999; Miller et al., 2003). However, the online extraction of such an object representation has proven to be very difficult. Hence, many robotic systems are still based on CAD models, or other user–provided information.

Figure 7.5: Objects and their related accumulated representation.

In this chapter we presented a scheme for extracting object model from manipulation. The knowledge of the robot's arm motion gives us two precious information: 1) it enables us to segment the object from the rest of the scene; and 2) it allows to track object features in a robust manner. In combination with the visually induced grasping reflex presented in (Aarno et al., 2007), this allows for an exploratory behaviour where the robot attempts to grasp parts of its environment, examines all successfully grasped shapes and learns their 3D model and by this becomes an important submodule of the cognitive system discussed in (Geib et al., 2006).

Note The work presented in this chapter was done with the cooperation of Emre Baseski (robot's hand elimination) and Dirk Kraft (robot's control and calibration), and was previously presented in (Pugeault et al., 2007a).

Chapter 8

Conclusions

The soul never thinks without a mental picture.

- Aristotle

In this thesis, we presented a novel framework for early vision, making use of feedback mechanisms between different levels of the processing hierarchy, in order to recurrently disambiguate the internal representation of visual information. The symbolic representation of visual information allows for the strong predictions that makes such feedback mechanisms possible and efficient.

In order to summarise the findings discussed herein, we will come back to the block diagram presented in the introduction of this thesis (Fig. 8.1). This thesis was divided into three parts corresponding to three different levels of representation: the image representation $\boldsymbol{\mathcal{I}}$ (part I, chapters 2 and 3); the $2\frac{1}{2}$D scene representation $\boldsymbol{\mathcal{S}}$ (part II, chapters 4 and 5); and the 3D accumulated representation $\boldsymbol{\mathcal{A}}$ (part III, chapters 6 and 7). Each part represents visual information in a progressively more abstracted, symbolic manner, from transient 2D–primitives to accumulated 3D–primitives that record their motion over time.

In chapter 2, we presented the image representation used throughout this thesis, and argued that such a local, multi–modal, symbolic representation is essential for drawing relations between visual events. These 2D–primitives were used for stereo–matching in chapter 4, and to reconstruct information about the scene structure, in terms of local 3D–primitives. The preservation of the dual representation 3D–primitives/stereo–pair of 2D–primitives by an adequate definition of the reconstruction and re–projection of 3D–primitives, allows to draw relations in the domain that is more adequate for each specific process.

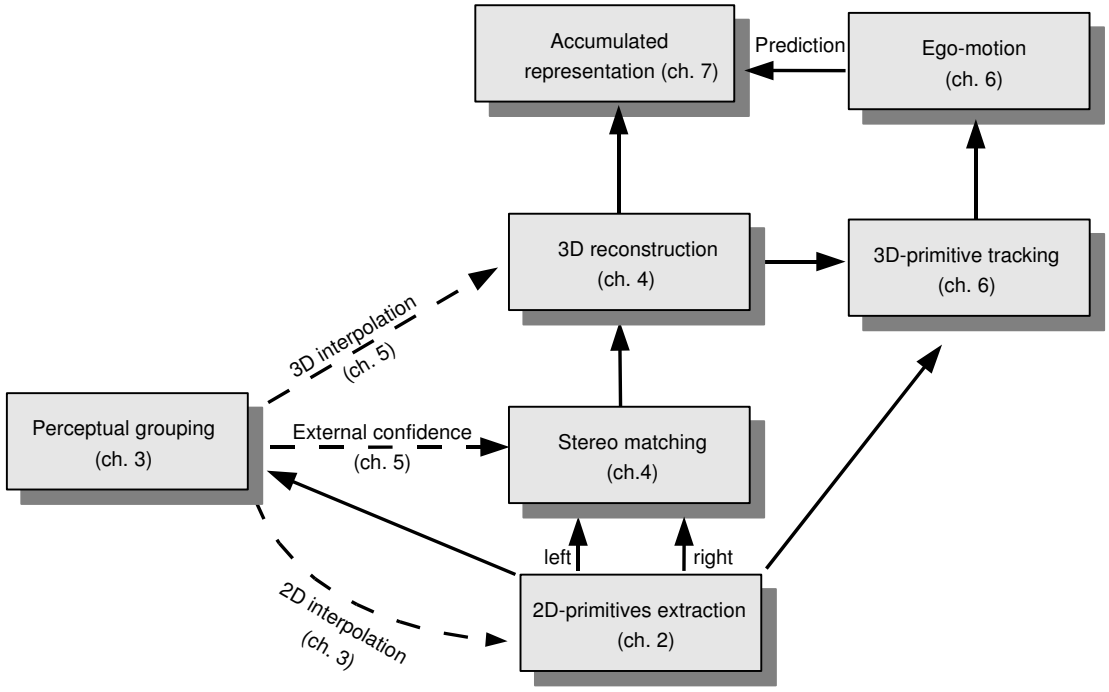


Figure 8.1: Presentation of the framework discussed in this thesis. The dashed lines represent feedback mechanisms, the full ones bottom–up processes.

For example, the grouping, presented in chapter 3, is defined in 2D then extended to the 3D domain. It was discussed that the inaccuracy of reconstructed 3D–primitives' localisation makes difficult to define the *Gestalt* law of *Good Continuation* in 3D space. On the other hand, defining the same law of Good Continuation in the image domain is fairly simple — we refer the interested reader to the numerous studies on 2D contour grouping, a selection of which is presented in section 3.1. Conversely, predicting a 2D–primitive's transformation under a rigid motion is difficult in the image domain, but is well defined in the 3D space; therefore, in chapter 7 we implemented our temporal prediction mechanism in the 3D domain.

In chapter 3, we made use of the rich semantic associated to the 2D–primitives to assess how likely it is for a pair of 2D–primitives to describe the same image contour. These links provide a good representation of an image's contours, in a more robust manner than individual 2D–primitives: a contour is always extracted as a string of 2D–primitives but how many, and at what locations along the contour the 2D–primitives are extracted varies depending on the sampling parameters. We used these links to correct the extracted 2D– and 3D–primitives, and to reconsider the belief in locally inferred potential stereo correspondences of 3D–primitives. The information provided by the links allowed to improve the representation reliability and accuracy — both in the image and in the spatial domains.

Chapter 6 used the image and spatial representations ($\mathcal{I}$ and $\mathcal{S}$) conjointly, to compute the ego–motion of the system. This chapter is a good example of the benefits of our approach: First, the local optic flow, obtained by dense computations, is used as a predictor for 2D–primitives' correspondences over time. Because we look for a corresponding 2D–primitive around this predicted location with similar multi–modal properties, the reliability of the matches is greatly improved from the original optic flow. Second, the dual (3D/2D+stereo) representation allows us to draw 3D–point/2D–line correspondences, and therefore to make use of the robust stereo correspondences obtained in chapters 4 and 5, while still matching 2D–primitives over time. Therefore, we efficiently used all available information (dense local optic flow, stereo correspondences, links) together in order to estimate the ego–motion.

Finally, chapter 7 presents a way to build an accumulated representation $\mathcal{A}$ from the transient spatial representations constructed in the preceding chapters. This process makes use of a known motion of the object studied, *e.g.*, from the ego–motion mechanism presented in chapter 6, or from the knowledge of the robot's arm motion. There we have shown that a full 3D model of the object/scene can be inferred — assuming that the system has viewed enough different perspectives of the object/scene. This mechanism

makes use of the motion's knowledge to predict the spatial representation from earlier representations. We assessed how reliably the predictions inferred from an hypothetical 3D–primitive are confirmed by the system's observation. This confirmation of the system's predictions were used to re–evaluate our confidence in 3D–primitives and thereby to discard outliers.

The whole framework is illustrated in Fig. 8.1. In this diagram the solid arrows show the normal flow of information, from earlier image based representations, towards higher, more abstracted, representations. The dashed arrows show feedback mechanisms that induces correction, and disambiguation processes (as described above). The progressive abstraction of the representation, comes together with vast improvements in reliability and accuracy.

8.1 Applications

The framework presented in this thesis has been developed during the last years, and applied in different contexts. It is currently used as visual–front end for the European projects PACO-PLUS (2006) and DrivSco (2006).

The PACO-PLUS (2006) project aims to address the symbol grounding problem in a robotic framework, associating an object with the actions it affords thereby defining Object Action Complexes (OACs) — see (Geib et al., 2006). Those OACs need to be learned by the system by 1) exploration, or 2) imitation. The former is of particular interest here: in this context, we need to provide the system with an *exploratory behaviour* that allows it to acquire knowledge about its environment, and the objects that populate it. In this context, the framework presented herein serves as a visual front–end. The representation of visual information described here provides rich semantics without requiring assumptions about the domain. Moreover, because conflicting hypotheses are preserved, early decisions can be re–evaluated when more contextual knowledge is available — in a manner similar to chapter 7. Feedback mechanisms can be initiated by higher level knowledge about the scene (*e.g.*, object knowledge).

In this context, the representation presented in this thesis was used by Aarno et al. (2007) to elicit grasping reflex. This grasping action is called a 'reflex' because it does not require knowledge of the object that is grasped, or even that it is an object: this action is merely elicited by a *structural configuration* of 3D–primitives in the scene — in this case coplanar pairs. This reflex is not expected to succeed everytime, but, provided it succeed once, allows the system to take control of the object and to manipulate

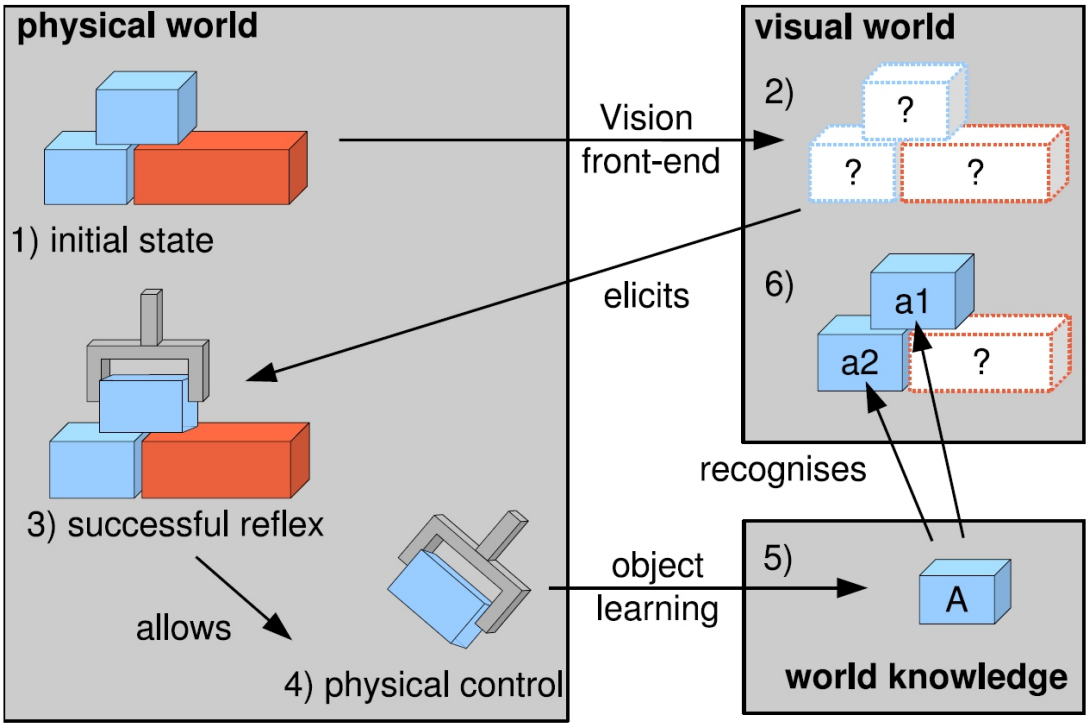


Figure 8.2: Illustration of the exploratory behaviour envisionned in the PACOplus project.

it. Given the assumption that everything that moves rigidly with the manipulated pair of 3D–primitives forms an object, this allows to segment the whole object from the scene. In 7, this is used to learn the full 3D model of the manipulated object.

Therefore, coupling the two above-mentioned mechanisms, we can define a primitive exploratory behaviour that allows the system to learn about the objects populating the scene, and what grasping actions are valid for them. Fig 8.2 illustrates such a behaviour: From the scenario shown in Fig. 8.2-1), the framework presented in parts I and II provides the unsegmented visual representation of the world — Fig. 8.2-2). The mechanism proposed by (Aarno et al., 2007) elicits grasping reflexes from this *a priori* visual representation — Fig. 8.2-3) — that, if successful, gives physical control on the object — Fig. 8.2-4). Then, the accumulation presented in 7, and (Pugeault et al., 2007a) is used to segment the object from the rest of the scene using motion knowledge, and to learn a full 3D representation of the object's shape — Fig. 8.2-5). From there on, the anticipated object recognition and pose estimation mechanisms can identify other instances of the same object in the scene. Furthermore, it can infer possible grasping actions from its experience with the first instance. This mechanism enables the system to learn about its environment by interacting with it, where the interaction is bootstrapped by reflexes elicited from the pre–attentive visual stimulus.

The European project DrivSco (2006) aims to learn driving behaviour through the study of human drivers, and its correlation with driving related events. More specifically, this system aims to identify Structured Visual Events (SVE) and to associate their appearance to the actions of the driver. SVEs are different from the context described above in the sense that most driving related visual events are very codified: traffic signs, road markings, other vehicles, *etc.*. Therefore, in this context, it is preferable for these SVE to be provided as prior knowledge by the designer. In this context, the representation presented herein is advantageous in the sense that it is explicit. 3D–primitives can easily be related to geometrical structures, and therefore prior model knowledge can be given in terms of such geometric structures (*e.g.*, red and white triangle) — see also (Pugeault et al., 2007b).

8.2 Future work

Although the present work covers a broad width of vision problems, the field is too wide, and the problems too complex, to be possibly covered during the course of one's PhD. Therefore, and although we

are aware of their importance, several aspects of the vision problem were purposefully left out:

Scale: Although in chapter 2 we discussed that 2D–primitives could be extracted from images at different scales, we did not integrate scale within our framework. This would require to define precisely the relations between 2D–primitives at different scales, and to study further the line/edge bifurcation problem as a function of scale — see (Krüger et al., 2007). Note that, because the 3D–primitives' size is computed dynamically there is no structural problem with extending the framework into scale–space.

Corners: Kalkan et al. (2007b) presented a novel junction extraction algorithm, based on the intrinsic dimension concept presented in chapter 2. Current work is carried out in order to provide a symbolic description of these junctions, as well as matching and grouping relations, in order to integrate them in the representation (especially in the ego–motion estimation algorithm).

Homogeneous surface patches & textures: Kalkan et al. (2007a) use the depth information provided at contours by the stereo described in chapter 4 to predict depth at homogeneous areas. Additional primitives need to be developed to encode texture knowledge.

Independently moving objects (IMO): In chapter 6 we computed the ego–motion of the system, disregarding independently moving objects. This is unsuitable for a system operating in a dynamic environment (*e.g.*, driving on a road with other vehicles). Therefore we intend to include an optic flow based segmentation mechanism and to estimate the RBM of all IMO — see (Pauwels and Van Hulle, 2004).

Active vision: Finally, the camera set–up used throughout this thesis is a static one. Using an active camera system would give a better account of how such mechanisms can occur in human vision.

Object description & recognition: The accumulated representation $\mathcal{A}$ presented here is not intended to be final, but rather to be the first step towards an abstract, symbolic and object–centred representation of the world. Such a representation requires to devise a symbolic description of the groups described herein, and a study of the relations between them. Moreover, object recognition and pose estimation mechanisms are critical to describe the scene not merely in terms of features, but rather in terms of the objects that populate it.

These subjects are actively researched within the two projects PACO-PLUS (2006); DrivSco (2006), and are the subject of several PhD and master theses, and we expect these projects to bring novel understanding of the connections between visual processes.

Appendices

Appendix A

Receiver Operating Characteristic (ROC) analysis

One efficient tool for making explicit the separability of two classes T and F from a given measurement is called *Receiver Operating Characteristic* (ROC) analysis. This analysis allows to evaluate how successfully one measurement allows separating two classes, across all possible thresholds. If we consider two classes T and F such that $T \cap F = \varnothing$, and a set of elements $x_i \in T \cup F$ for which we have a measurement $m(x_i)$ that is supposed to be characteristic of T. Then, using a threshold τ over $m(x_i)$ is a way to separate the two classes such that

$$m(x_i) > \tau \implies x_i \in T \tag{A.1}$$

This implication only stands if the two distributions of the measurement m for elements from T and F, respectively, are separable — see Fig. A.1.

If they are overlapping, then m can provide at best an imperfect classification. In this general case, and for each threshold, several quantities can be estimated: We define as True Positives (TP) the elements $x_i \in T$ for which the measurement $m(x_i) > \tau$. Similarly we define as False Positives (FP) the elements $x_i \in F$ that nonetheless satisfy $m(x_i) > \tau$. The True Negatives (TN) are then members of T such that $m(x_i) \leq \tau$ and and the False negatives (FN) are the members of F such that $m(x_i) \leq \tau$. Therefrom we define the two following ratios:

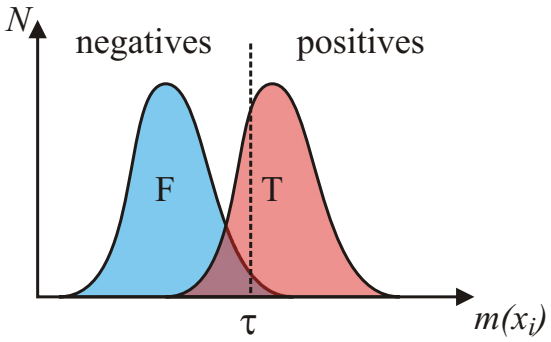


Figure A.1: Separation between classes T and F using a threshold τ over a measurement m. the value N stands for the number of x_i with an associated measurement m.

True Positive Rate (tpr(τ)) as the ratio of the number of true positives divided by the total number of true datapoints.

$$\text{tpr}(\tau) = \frac{TP}{TP + TN} \tag{A.2}$$

False Positive Rate (fpr(τ)) as the ratio of the number false positives divided by the total number false datapoints.

$$\text{fpr}(\tau) = \frac{FP}{FP + FN} \tag{A.3}$$

The ROC curve is the plot of fpr(τ) (on the y axis) against tpr(τ) (on the x axis), namely the proportion of true and false positive for any threshold τ. Note that for a threshold $\tau = 0$ all x_i satisfy $m(x_i) > 0$ and therefore are classified as belonging to A, thus the ROC curve reaches the point fpr(0) = 1 and tpr(0) = 1. Conversely, a threshold $\tau = 1$ none of the x_i will satisfy $m(x_i) > 1$ and therefore all x_i will be classified as belonging to B, thus the ROC curve also reach the point fpr(1) = 0 and tpr(1) = 0.

It follows that, if we consider a measurement m that is independent whether x belongs to T or F (such that $p[x \in T|m(x) > \tau] = p[x \in T]$ for all τ) then we have $\text{tpr}(\tau) \neq \text{fpr}(\tau), \forall\tau$, and the ROC curve is the straight line joining $(0, 0)$ and $(1, 1)$. On the other hand, if the measurement is a significantly better classifier than chance we have $\text{tpr}(\tau) > \text{fpr}(\tau), \forall\tau$, and the resulting ROC curve should be convex. Furthermore, the larger the area below the curve (*i.e.*, the stronger the convexity), the more adapted is the measurement for the classification task.

Such a method also give the additional advantage of illustrating in one figure the effect of the whole range of possible threshold. This is especially important in the field of computer vision, where experience

tends to show that the exact setting of the threshold can be very sensitive. Having a smooth ROC curve would be a guarantee that inaccuracies in the choice of the threshold degrades gracefully the performance of the classification.

Appendix B

Projective Geometry

The camera geometry is commonly formalised using a non Euclidian type of geometry, called *projective geometry*. We will quickly present the formulas used in this work in this section, the reader already familiar with this can skip to the next section. The material from this section comes mainly from (Maxwell, 1951) and (Faugeras, 1993) and the reader is directed to those works for demonstrations and justification of those formulae.

In this section we will assume that $\det(\boldsymbol{A})$ is the determinant of the matrix $\boldsymbol{A}$. Also, $\tilde{\boldsymbol{x}}$ is the homogeneous formulation of the vector $\boldsymbol{x}$, and $\tilde{\boldsymbol{A}}$ denotes an homogeneous matrix.

B.1 The projective plane $\mathcal{P}^2$

In the projective plane $\mathcal{P}^2$, a point $\tilde{\boldsymbol{m}}$ is represented by the triplet $(\tilde{u}, \tilde{v}, \tilde{w})^T$, with the constraint that $\tilde{u}$, $\tilde{v}$ and $\tilde{w}$ are not all null. Those coordinates are said *homogeneous* as their are only unique up to a scaling factor:

$$(\tilde{u}, \tilde{v}, \tilde{w})^T \equiv (\lambda\tilde{u}, \lambda\tilde{v}, \lambda\tilde{w})^T \tag{B.1}$$

Also, the classical Euclidian coordinates $(u, v)^T$ of the same point can be obtained as follows:

$$\begin{aligned} u &\equiv \tilde{u}/\tilde{w} \\ v &\equiv \tilde{u}/\tilde{w} \end{aligned} \tag{B.2}$$

Note that, in the following we will drop the tilde for scalar coordinates, homogeneous or not, and only write the tilde for homogeneous vectors and matrices. From equation (B.2) one can see that there is a class of points $(u, v, 0)^T$ of the projective plane which do not exist in the classical Euclidian plane. Those points are called *points at infinity* and are the projective equivalent of vectors. They linked to a very interesting property of the projective plane: two given lines are *always* concurrent: parallel lines intersect at infinity.

A line $\tilde{\boldsymbol{l}}$ in the projective space is defined by the triplet (a, b, c), such as a point $\tilde{\boldsymbol{m}} = (u, v, w)^T$ lie on this line if and only if

$$au + bv + cw = 0 \tag{B.3}$$

In the following we will prefer the more condensed notation:

$$\tilde{\boldsymbol{l}} \cdot \tilde{\boldsymbol{m}} = 0 \tag{B.4}$$

Three points $\tilde{\boldsymbol{m}}$, $\tilde{\boldsymbol{n}}$, and $\tilde{\boldsymbol{p}}$ are collinear iff. the determinant of the following 3x3 matrix vanishes:

$$\begin{vmatrix} \tilde{\boldsymbol{m}}^T \\ \tilde{\boldsymbol{n}}^T \\ \tilde{\boldsymbol{p}}^T \end{vmatrix} = 0 \tag{B.5}$$

An explanation of the origin of this matrix, and a proof of this theorem can be found in (Maxwell, 1951). Hence, the homogeneous coordinates $\tilde{\boldsymbol{l}}$ of the line l_n^m joining two points $\tilde{\boldsymbol{m}}$ and $\tilde{\boldsymbol{n}}$ is defined by all the points $\tilde{\boldsymbol{p}}$ for which the equation (B.5) holds. Therefore, we obtain that:

$$l_n^m \equiv \tilde{\boldsymbol{l}} = \left(\begin{vmatrix} m_v & m_w \\ n_v & n_w \end{vmatrix}, - \begin{vmatrix} m_u & m_w \\ n_u & n_w \end{vmatrix}, \begin{vmatrix} m_u & m_v \\ n_u & n_v \end{vmatrix} \right) \tag{B.6}$$

Also, because points and lines are dual in the projective plane, a similar formula can be derived to compute the intersection $\tilde{\boldsymbol{m}}$ between two lines $\tilde{\boldsymbol{l}}$ and $\tilde{\boldsymbol{l}}'$:

$$\tilde{\boldsymbol{m}} = \left(\begin{vmatrix} l_2 & l'_2 \\ l_3 & l'_3 \end{vmatrix}, - \begin{vmatrix} l_1 & l'_1 \\ l_3 & l' a_3 \end{vmatrix}, \begin{vmatrix} l_2 & l'_2 \\ l_3 & l'_3 \end{vmatrix} \right)^T \tag{B.7}$$

B.2 The projective space $\mathcal{P}^3$

This section extends previous formulae into the 3D projective space $\mathcal{P}^3$. In the projective space, points are defined as $\tilde{\boldsymbol{m}} = (\tilde{m}_x, \tilde{m}_y, \tilde{m}_z, \tilde{m}_w)^T$. The Euclidian coordinates $\boldsymbol{m} = (x, y, z)^T$ can be computed from homogeneous coordinates as before:

$$\begin{aligned} x &\equiv \frac{\tilde{m}_x}{\tilde{m}_w} \\ y &\equiv \frac{\tilde{m}_y}{\tilde{m}_w} \\ z &\equiv \frac{\tilde{m}_z}{\tilde{m}_w} \end{aligned} \tag{B.8}$$

Also, in the projective space points are dual to planes, and lines are self–duals.

B.2.1 Planes in space:

Four points $\tilde{\boldsymbol{m}}$, $\tilde{\boldsymbol{n}}$, $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{q}}$ are coplanar if the following condition is respected:

$$\begin{vmatrix} \tilde{\boldsymbol{m}} & \tilde{\boldsymbol{n}} & \tilde{\boldsymbol{p}} & \tilde{\boldsymbol{q}} \end{vmatrix} = 0 \tag{B.9}$$

This formula is demonstrated in (Maxwell, 1951). Hence, if we have three non–collinear points in space $\tilde{\boldsymbol{m}}$, $\tilde{\boldsymbol{n}}$ and $\tilde{\boldsymbol{p}}$, the equation of the plane they define is:

$$\tilde{\boldsymbol{P}} = \begin{pmatrix} \begin{vmatrix} m_y & n_y & p_y \\ m_z & n_z & p_z \\ m_w & n_w & p_w \end{vmatrix}, - \begin{vmatrix} m_x & n_x & p_x \\ m_z & n_z & p_z \\ m_w & n_w & p_w \end{vmatrix}, \begin{vmatrix} m_x & n_x & p_x \\ m_y & n_y & p_y \\ m_w & n_w & p_w \end{vmatrix}, - \begin{vmatrix} m_x & n_x & p_x \\ m_y & n_y & p_y \\ m_z & n_z & p_z \end{vmatrix} \end{pmatrix} \tag{B.10}$$

Also, the distance from a point $\tilde{\boldsymbol{m}}$ to a plane $\tilde{\boldsymbol{P}}$ is obtained using the following formula:

$$d(\tilde{\boldsymbol{m}}, \tilde{\boldsymbol{P}}) = \tilde{\boldsymbol{m}} \cdot \tilde{\boldsymbol{P}} \tag{B.11}$$

And the sign of this distance indicates on which side of the plane the point lies.

B.2.2 Lines in space:

The coordinates of a line defined by two points $\tilde{\boldsymbol{m}}$ and $\tilde{\boldsymbol{n}}$ are the determinants of the six 2x2 sub–matrices of the 4x2 matrix:

$$\begin{vmatrix} \tilde{\boldsymbol{m}} & \tilde{\boldsymbol{n}} \end{vmatrix} \tag{B.12}$$

The coordinates are defined as the

$$\Delta_{ij} = \begin{vmatrix} \tilde{m}_i & \tilde{n}_i \\ \tilde{m}_j & \tilde{n}_j \end{vmatrix} \tag{B.13}$$

Only 6 of those determinants are apparently independent, as $\Delta_{ij} = -\Delta_{ji}$, and they are defined up to a scale factor. We will consider the six determinants Δ_{41}, Δ_{42}, Δ_{43}, Δ_{23}, as in (Maxwell, 1951)[1]. The *ratio* between those six coordinates define uniquely a line in space. Those six numbers are called *Plücker (or Grassman) coordinates*. Note that the same line L is defined by the six numbers:

$$L \equiv \lambda\,(\Delta_{41}, \Delta_{42}, \Delta_{43}, \Delta_{23}, \Delta_{31}, \Delta_{12}) \tag{B.14}$$

up to a scaling factor $\lambda \neq 0$. Those coordinates are also additionally constrained such that:

$$\Delta_{41}\Delta_{23} + \Delta_{42}\Delta_{31} + \Delta_{43}\Delta_{12} = 0 \tag{B.15}$$

B.2.3 Line intersection

Two lines L and L' intersect if and only if their coordinates $\tilde{\boldsymbol{L}} = (\Delta_{41}, \Delta_{42}, \Delta_{43}, \Delta_{23}, \Delta_{31}, \Delta_{12})$ and $\tilde{\boldsymbol{L}}' = (\Delta'_{41}, \Delta'_{42}, \Delta'_{43}, \Delta'_{23}, \Delta'_{31}, \Delta'_{12})$ are such as:

$$(\Delta_{41}\Delta'_{23} + \Delta'_{41}\Delta_{23}) + (\Delta_{42}\Delta'_{31} + \Delta'_{42}\Delta_{31}) + (\Delta_{43}\Delta'_{12} + \Delta'_{43}\Delta_{12}) = 0 \tag{B.16}$$

B.2.4 Plane intersection

As in the conventional Euclidian space, the intersection of two non–parallel planes is a line. Yet in addition to that, the intersection of two parallel planes also exists in the projective space: it is a line at infinity. Due to the duality between points and planes in the projective space, the formulation of

[1] Note that the sign of the first three coordinates Δ_{41}, Δ_{42} and Δ_{43} may vary in some textbook, like in (Maxwell, 1951). This only lead to minor adjustments in the following formulae.

the intersection between two planes $\tilde{\boldsymbol{P}}$ and $\tilde{\boldsymbol{Q}}$ is similar to the junction between two points, *i.e.*, the determinants of 6 submatrices of:

$$\begin{vmatrix} \tilde{\boldsymbol{P}}^T & \tilde{\boldsymbol{Q}}^T \end{vmatrix} \tag{B.17}$$

The coordinates then are computed similarly to equation B.13

$$\Delta_{ij} = \begin{vmatrix} \tilde{p}_i & \tilde{q}_i \\ \tilde{p}_j & \tilde{q}_j \end{vmatrix} \tag{B.18}$$

To be consistent with the previous line formulation, we choose the following determinants:

$$L \equiv \lambda \tilde{\boldsymbol{L}} = \lambda \left(\Delta_{23}, \Delta_{31}, \Delta_{12}, \Delta_{41}, \Delta_{42}, \Delta_{43}\right) \tag{B.19}$$

cf. (Maxwell, 1951) for an explanation of this formula. Note the similarity of this formula with the equation B.14 an example of the duality between points and planes in the projective space.

B.3 Euclidian interpretation

With the above formulae we have defined coordinates for points, lines and planes in the projective space, and their intersections. Yet as those coordinates are homogeneous, they are all defined up to a factor. If we choose to constraint additionally those coordinates, we can obtain sets of coordinates and formulae carrying a nice Euclidian interpretation.

B.3.1 Points coordinates

We will in the following normalise the points coordinates such as:

$$\boldsymbol{m} \equiv \tilde{\boldsymbol{m}} = \begin{pmatrix} \boldsymbol{m} \\ 1 \end{pmatrix} \tag{B.20}$$

with $\boldsymbol{m}$ being the Euclidian coordinates of the point, and $\tilde{\boldsymbol{m}}$ the homogeneous coordinates of the same point. This allows us to simplify the other formulae as follows.

B.3.2 Plane coordinates

If one consider plane $\tilde{\boldsymbol{P}} = (a, b, c, d)$, and two points on this plane $\tilde{\boldsymbol{m}} = (m_x, m_y, m_z, 1)^T$ and $\tilde{\boldsymbol{n}} = (n_x, n_y, n_z, 1)^T$ on this plane, then we have the equations:

$$\begin{cases} am_x + bm_y + cm_z + d = 0 \\ an_x + bn_y + cn_z + d = 0 \end{cases} \tag{B.21}$$

$$a(n_x - m_x) + b(n_y - m_y) + c(n_z - m_z) = 0 \tag{B.22}$$

$$(a, b, c) \cdot (\boldsymbol{n} - \boldsymbol{m}) = 0 \tag{B.23}$$

From equation B.23 it is clear that the vector $\boldsymbol{\eta} = (a, b, c)^T$ is normal to the plane $\tilde{\boldsymbol{P}}$. As the coordinates of $\tilde{\boldsymbol{P}}$ are defined up to a scalar, we choose λ such a $\|\boldsymbol{\eta}\| = 1$. If one consider the projective formulation of the origin of the coordinate system $\tilde{\boldsymbol{o}} = (0, 0, 0, 1)$, it becomes obvious that $|d|$ is the distance of the plane $\tilde{\boldsymbol{P}}$ from the origin $\tilde{\boldsymbol{o}}$. Then the coordinates of the plane are defined up to a sign as :

$$P \equiv \tilde{\boldsymbol{P}} = \begin{pmatrix} \boldsymbol{\eta}^T & -h \end{pmatrix} \tag{B.24}$$

with $\boldsymbol{\eta}$ the normal vector to the plane, and h the Euclidian distance from the origin to the plane — also called Hesse distance of the plane.

B.3.3 Line coordinates

If one consider points $(m_x, m_y, m_z, 1)^T$ and $(n_x, n_y, n_z, 1)^T$, then we can give to the Plücker coordinates of the line L joining those two points the following Euclidian interpretation:

$$\begin{cases} \boldsymbol{\nu} = \lambda(\Delta_{41}, \Delta_{42}, \Delta_{43})^T = \frac{1}{\|\boldsymbol{n}-\boldsymbol{m}\|} (\boldsymbol{n} - \boldsymbol{m}) \\ \boldsymbol{\mu} = \lambda(\Delta_{23}, \Delta_{31}, \Delta_{12})^T = \frac{1}{\|\boldsymbol{n}-\boldsymbol{m}\|} (\boldsymbol{m} \times \boldsymbol{n}) \end{cases} \tag{B.25}$$

Note that, as the line coordinates are defined up to a scale factor, we normalise them by $\lambda = \frac{1}{\|\boldsymbol{n}-\boldsymbol{m}\|}$ such as $\|\boldsymbol{\nu}\| = 1$. The second part of those coordinates is called the the *moment* $\boldsymbol{\mu}$ of the line. For any point of

the line $\boldsymbol{M}$ there is:

$$\boldsymbol{m} \times \boldsymbol{\nu} = \boldsymbol{\mu} \tag{B.26}$$

Then the line coordinates can be described as the aggregation of the orientation vector of the line $\boldsymbol{\nu}$ and of the moment of the line $\boldsymbol{\mu}$.

$$L \equiv \tilde{\boldsymbol{L}} = \begin{pmatrix} \boldsymbol{\nu}^T & \boldsymbol{\mu}^T \end{pmatrix} \tag{B.27}$$

so that they are defined uniquely up to a sign.

B.3.4 Line to point distance

The distance from the line $\tilde{\boldsymbol{L}}$ to a point $\boldsymbol{m}$:

$$d(\boldsymbol{m}, \tilde{\boldsymbol{L}}) = \|\boldsymbol{m} \times \boldsymbol{\nu} - \boldsymbol{\mu}\| \tag{B.28}$$

Note that here the Euclidian coordinates $\boldsymbol{m}$ of the point is used. Quite naturally, it follows that the line $\tilde{\boldsymbol{L}}$ is defined by the kernel of $d(\boldsymbol{m}, \tilde{\boldsymbol{L}})$.

B.3.5 Plane intersection

The line at the intersection of two planes $\tilde{\boldsymbol{P}}_1 = (\boldsymbol{\eta}_1, -h_1)$ and $\tilde{\boldsymbol{P}}_2 = (\boldsymbol{\eta}_2, -h_2)$ is computed as follows:

$$\tilde{\boldsymbol{L}} = \begin{pmatrix} \boldsymbol{\eta}_1 \times \boldsymbol{\eta}_2 & h_2\boldsymbol{\eta}_1 - h_1\boldsymbol{\eta}_2 \end{pmatrix} \tag{B.29}$$

B.3.6 Line-Plane intersection

Intersection between a line and a plane $\tilde{\boldsymbol{P}} = (\boldsymbol{\eta}, -h)$ and a line $\tilde{\boldsymbol{L}} = (\boldsymbol{\nu}, \boldsymbol{\mu})$ is obtained as follows:

$$\tilde{\boldsymbol{m}} = \begin{pmatrix} \boldsymbol{\eta} \times \boldsymbol{\mu} + h\boldsymbol{\nu} \\ \boldsymbol{\eta} \cdot \boldsymbol{\nu} \end{pmatrix} \tag{B.30}$$

Note that this formula yield a homogeneous formulation of the intersection point, which need to be normalised to get the non–homogeneous coordinates $\boldsymbol{m}$.

Appendix C

Camera Geometrical Model

The perception of the world of a camera can be seen as a projection of the 3D space onto a image plane I^i. The geometrical properties of this projection depends on the optics and electronics of the camera being used — we refer to (Geißler, 1999) for an accurate description of imaging optics and related problems. In this work we will assume a perfect spherical optic, which projective properties are solely dependent on its focal length. In general, we estimated the projection matrices using the calibration Matlab package from (Bouguet, 2007). The resulting projection matrices show very small deviation from the ideal spherical model, so we will neglect the other parameters for explanation purposes, although they were included in computations of the full system.

We will assign a coordinate system $\{\boldsymbol{O}, \boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e_3}\}$ to the space, where $\boldsymbol{O}$ is the origin, and $\boldsymbol{e_2}$ is the vertical coordinate, oriented downwards. In general each sequence will start with $\boldsymbol{e_1}$ being horizontal, oriented towards the right of the camera, and $\boldsymbol{e_3}$ is horizontal facing forward. Of course, as the camera itself is moving through a sequence, the relative orientation of those axis depends on the timeframe and camera motion. Similarly, we assign to the image plane a system of coordinates $\{\boldsymbol{o}, \boldsymbol{i}, \boldsymbol{j}\}$, where the origin $\boldsymbol{o}$ is at the top left corner of the image and the coordinates $\boldsymbol{i}$ and $\boldsymbol{j}$ describe the column and row, respectively, of a pixel in the image. As before we will note $\boldsymbol{X} = (X_1, X_2, X_3)^T$ a point in space, where $\boldsymbol{X} = \boldsymbol{O} + X_1\boldsymbol{e_1} + X_2\boldsymbol{e_2} + X_3\boldsymbol{e_3}$ and $\boldsymbol{x} = (u, v)^T$ a point in the image plane such as $\boldsymbol{x} = \boldsymbol{o} + u\boldsymbol{i} + v\boldsymbol{j}$

Then the projection performed by the camera can be formalised as a matrix operation, using the homogeneous formulation presented in the previous section. This operation transforms a 3D homogeneous point $\tilde{\boldsymbol{X}} = (X_1, X_2, X_3, 1)^T$ into a 2D homogeneous point $\tilde{\boldsymbol{x}}^i = (u, v, w)^T$ using the following homogeneous

matrix:

$$\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{P}}^{i}\tilde{\boldsymbol{x}} \tag{C.1}$$

The 3x4 matrix $\tilde{\boldsymbol{P}}$ is called the *projection matrix* associated to the camera. The properties of this matrix are exposed in details in (Faugeras, 1993). In the following we will present only the basics which are sufficient in the general case.

C.1 The Projection Matrix

$\tilde{\boldsymbol{A}}$ is the matrix containing the actual projection operated by the camera lens. As stated before, and for simplification reasons, we can assume a perfectly spherical lens, insofar that the non–spherical components of the matrix only have a minor impact on the computations and would only complicate the present model. Also, all of the parameters were included in the actual computations. Then the camera lens is fully determined by its focal length f, and can be formalised as the following 3x4 homogeneous matrix:

$$\tilde{\boldsymbol{A}} = \begin{pmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{C.2}$$

The matrix $\tilde{\boldsymbol{H}}$ holds the coordinate system of the image plane. This effectively carries the geometry of the photoreceptors of the camera. It is a 3x3 matrix as follows:

$$\tilde{\boldsymbol{H}} = \begin{pmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} \tag{C.3}$$

with k_u and k_v being the scaling in, respectively, the u and v coordinates of the image plane, and (u_0, v_0) the coordinates of the intersection between the optical axis and the optical plane.

Consequently the projection matrix for camera located at the origin O and which optical axis is along $\boldsymbol{e_3}$ would be :

$$\tilde{\boldsymbol{P}} = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{A}} \tag{C.4}$$

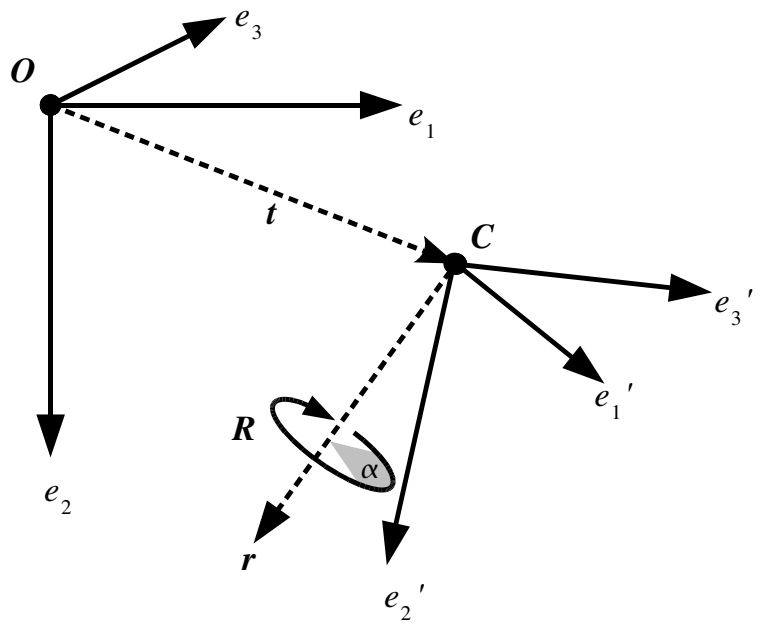


Figure C.1: In order to compute the projection operation of a given camera, we compute the coordinate system transformation between the default point of view $\{\boldsymbol{O}, \boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e_2}\}$ and the real one $\{\boldsymbol{C}, \boldsymbol{e_1}', \boldsymbol{e_2}', \boldsymbol{e_3}'\}$. In this case the rotation $\boldsymbol{R}$ is around the axis $\boldsymbol{r}$.

Now this formulation only apply for a camera located at the origin of the coordinate system, and which optical axis is $\boldsymbol{e_2}$. In the following we will generalise it so that it applies to any camera position and orientation in space.

C.2 Pose Matrix and Rigid Body Motion

In the following we will consider a special family of transformations in the space, which has the quality that, for any two points $\boldsymbol{X}$ and $\boldsymbol{X}'$ it transforms, the distance between those two points in conserved. Those transformations are called the *Rigid Body Motions* (RBM) in space. Now if we consider two different points of view on the same scene. The 3D distance between any two points of the scene is obviously a constant between those two points of view. Consequently, the coordinate transformation from one point of view to another can be formalised as a Rigid Body Motion of the world.

If one consider the coordinate system $\{\boldsymbol{C}, \boldsymbol{e'_1}, \boldsymbol{e'_2}, \boldsymbol{e'_3}\}$ of the camera, and $\{\boldsymbol{O}, \boldsymbol{e_1}, \boldsymbol{e_2}, \boldsymbol{e_3}\}$ of the world, then

we define a rotation $\boldsymbol{R}$ such as:

$$\begin{cases} \boldsymbol{e'_1} = \boldsymbol{Re_1} \\ \boldsymbol{e'_2} = \boldsymbol{Re_2} \\ \boldsymbol{e'_3} = \boldsymbol{Re_3} \end{cases} \tag{C.5}$$

and a translation $\boldsymbol{t}$ such as:

$$\boldsymbol{C} = \boldsymbol{O} + \boldsymbol{t} \tag{C.6}$$

As translations and rotations conserve the distance between all the points they are applied to, then the combination of those two is a Rigid Body Motion, and is fully contained in the following homogeneous 4x4 matrix:

$$\tilde{\boldsymbol{K}} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{t} \\ \boldsymbol{0}_3^T & 1 \end{pmatrix} \tag{C.7}$$

The matrix $\tilde{\boldsymbol{K}}$ Hold the position and the orientation of the camera relatively to the world coordinate systems. Those are called the *extrinsic parameters* of the camera. Note that the same RBM formulation is used in chapter 6, to formalise object and camera motion over time.

Consequently, the full projection matrix is the combination of 3 different geometrical operations: first, a rigid body motion in space; second, the actual projection onto the image plane; third, a scaling of the coordinates to the image plane coordinate system.

$$\tilde{\boldsymbol{P}} = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{A}}\tilde{\boldsymbol{K}} \tag{C.8}$$

C.3 Inferring the origin in space of image features

Now that we have defined mathematically the relation between a point in space and a pixel in the image plane, we propose to draw the inverse relationship: to infer the possible origins of one pixel in the image. Because one dimension has been lost (depth) through the projection on the image plane, the possible origins in space for a pixel is a one–dimensional manifold. Note that this is inaccurate, insofar that a pixel is not an ideal point in the image plane, but rather a circle of size dependent on the photoreceptors. Consequently, the constraint should not be a line in space but an infinite cone, that originates at the optical centre and which intersects with the image plane exactly at this pixel. For simplification, only the pixel's

centre is actually considered, allowing to constrain the 3D position to a line in space. Then the section of the cone at any point along this line yield an estimation of the uncertainty of our constraint at this depth, due to pixel sampling.

C.3.1 Back–projecting points

We call *optical ray* the 3D line containing all possible origins for a given pixel, and we will show in the following how the coordinates of this ray can be computed from the projection matrix of the camera and the pixel position. A line is fully defined by two points; in this case we will use two points that are easily obtained from the projection matrix: the optical centre and the intersection of the optical ray with the plane at infinity.

Computing the optical centre C: The coordinates of the optical centre C can be extracted from the projection matrix as follows: if one consider the projection matrix as $\tilde{\boldsymbol{P}} = [\boldsymbol{P}\ \tilde{\boldsymbol{p}}]$ then the Euclidian coordinates of the optical centre are:

$$C \equiv \boldsymbol{C} = -\boldsymbol{P}^{-1}\tilde{\boldsymbol{p}} \tag{C.9}$$

Computing the intersection with the plane at infinity X_∞: Secondly, if one consider a point $x \equiv \tilde{\boldsymbol{x}}$ in the optical plane I, then the optical ray generated by this point intersect with the plane at infinity at an homogeneous point having the coordinates $X_\infty \equiv (\boldsymbol{D}^T 0)^T$, with:

$$\boldsymbol{D} = \boldsymbol{P}^{-1}\tilde{\boldsymbol{x}} \tag{C.10}$$

Therefore, we can compute the coordinates of the optical ray $[C\ X_\infty)$ using equation B.25:

$$L_m \equiv \frac{1}{||\boldsymbol{D}||}(\boldsymbol{D}\ ,\ \boldsymbol{C} \times \boldsymbol{D}) \tag{C.11}$$

This gives us an equation of the possible origins in the 3D space of the image pixel.

C.3.2 Back–projecting lines

If we consider the line $l_x \in I$, we want to infer what are the possible origins of this line. If we consider a point x that lie on this line, we have seen in the previous section that its possible origins X in space forms

a 3D line L_x. We then consider a second point that lie on l_x, such that $y\,\boldsymbol{x} + \lambda \boldsymbol{u}$, where u is the direction vector of l_x. The possible origins Y of this second point also form a line in space L_y.

Consequently, those two points x and y give us 3 points in space: the optical centre $C \equiv (c_1, c_2, c_3, 1)^T$, and the intersections of the optical rays generated by the two points with the plane at infinity, respectively: $X_\infty \equiv (x_1, x_2, x_3, 0)^T$ and $Y_\infty \equiv (y_1, y_2, y_3, 0)^T$. From those three points we can then reconstruct the plane P_l using the formula B.10.

$$P_l \equiv \begin{pmatrix} \begin{vmatrix} x_2 & y_2 \\ x_3 & y_3 \end{vmatrix}, -\begin{vmatrix} x_1 & y_1 \\ x_3 & y_3 \end{vmatrix}, \begin{vmatrix} x_1 & y_1 \\ x_3 & y_3 \end{vmatrix}, -\begin{vmatrix} c_1 & x_1 & y_1 \\ c_2 & x_2 & y_2 \\ c_3 & x_3 & y_3 \end{vmatrix} \end{pmatrix} \tag{C.12}$$

C.4 The stereo case

Due to the loss of depth information, we need to have some additional information to identify which point in the optical ray is the actual origin of the pixel. The most obvious way to do that is to have a second image of the scene from another viewpoint, and to draw a second constraint on the 3D position from it. This is feasible, assuming we know two points, one in each image which *correspond*, meaning they are the manifestation in each image plane of the same 3D point. In general the two viewpoints are chosen with parallel optical axes, and the distance between their optical centres is called the *baseline*.

C.4.1 Point reconstruction

If we consider two cameras viewing the scene, which projection properties are defined by the projection matrices $\tilde{\boldsymbol{P}}^l$ and $\tilde{\boldsymbol{P}}^r$. The image planes of those cameras are called respectively I^l and I^r. Then if we assume that two points $x^l \in I^l$ and $x^r \in I^r$ correspond, then we know that the position of their origin in 3D X is at the intersection between the two optical rays :

$$X \equiv L^l_{\boldsymbol{x}} \cap L^r_{\boldsymbol{x}} \tag{C.13}$$

This hold in theory, assuming the two points are ideal points, and correspond perfectly. Empirically the points are pixels sampled in the image and this condition is never exactly met. This implies that the optical rays are actually more similar to cones in space, and therefore their intersection do not form a

precise point, but an area of the space. This area is proportional to the sampling factor and to the distance of the object, and is an estimation of the reconstruction inaccuracy produced by the pixel sampling.

C.4.2 Line reconstruction

If we consider two corresponding lines, l^l, and l^r, where $\boldsymbol{u}$ and $\boldsymbol{u'}$ are a points lying on each line, $\boldsymbol{v}$ and $\boldsymbol{v'}$ are the orientations of the lines. From section C.3.2, the possible origins of the pair are contained in the line at the intersection of the two planes:

$$L^{l,r} \equiv P^l \cap P^r \tag{C.14}$$

where P^l and P^r are the back–projections of the 2D–lines l^l and l^r.

C.5 Epipolar Constraint

The camera geometries can be used to constrain additionally the position of the correspondence in the second image. As we know that a given point present in an image has to lie in the half line between the optical centre and the infinity, then the possible positions for its correspondence on the second image lies on the projection of this half–line onto the second optical plane. This line is called the *epipolar line* — see Fig. C.2.

We can estimate this line by projecting the two points C^l and X^l_∞ onto I^r:

$$\begin{aligned} \tilde{\boldsymbol{e}}^r_x &= \tilde{\boldsymbol{P}}^r \tilde{\boldsymbol{C}}^l \\ \tilde{\boldsymbol{x}}^r_{\text{inf}} &= \tilde{\boldsymbol{P}}^r \tilde{\boldsymbol{X}}^l_{\text{inf}} \end{aligned} \tag{C.15}$$

The point $\tilde{\boldsymbol{e}}^r_x$ is called the *epipole*, and in the common case of a fronto–parallel set–up, this point lies on the line at infinity of the optical plane — in Fig. C.2, and for the sake of clarity, we chose a stereo set–up with a strong vergence, and the epipole lie within the image. The point $\tilde{\boldsymbol{x}}^r_\infty$ is the correspondence of $\tilde{\boldsymbol{x}}^l$ if the original 3D point lies on the plane at infinity. Points beyond this point on the epipolar line correspond to 3D origins behind the camera. The line

$$\xi^r_x \equiv (\tilde{\boldsymbol{x}}^r_\infty, \tilde{\boldsymbol{e}}^r_x) \tag{C.16}$$

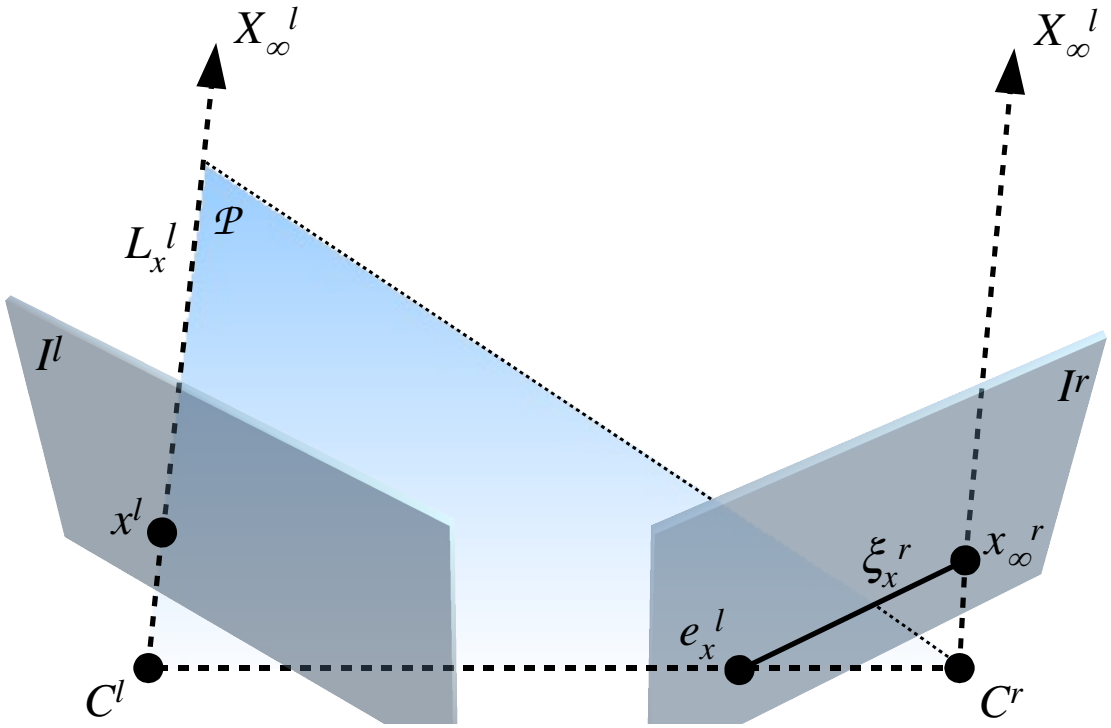


Figure C.2: Illustration of the epipolar line constraint — see text.

is called the *epipolar line*. For specific set–ups, when both points lie actually on the retinal plane (*e.g.*, see Fig. C.2), the search can be reduced to the segment between those two points $[\tilde{\boldsymbol{x}}^r_{\text{inf}}, \tilde{\boldsymbol{e}}^r_x]$. Note that the epipolar line is commonly computed using the so–called *fundamental matrix* $\boldsymbol{F}$, such that two points x^r lie on the epipolar line of x^r iff.

$$\boldsymbol{x}^{rT}\boldsymbol{F}\boldsymbol{x}^l = 0 \tag{C.17}$$

The fundamental matrix can be computed from the projection matrices as follows:

$$\boldsymbol{F} = [\tilde{\boldsymbol{e}}^r]_\times \tilde{\boldsymbol{P}}^{\boldsymbol{r}} \tilde{\boldsymbol{P}}^{\boldsymbol{l}^+} \tag{C.18}$$

where $[\tilde{\boldsymbol{e}}^r]_\times$ is the cross–product matrix for the vector $\tilde{\boldsymbol{e}}^r$, and $\tilde{\boldsymbol{P}}^{l^+}$ is the pseudo–inverse of $\tilde{\boldsymbol{P}}^l$. We refer the reader to (Hartley and Zisserman, 2000, chapter 8) for a discussion of the fundamental matrix.

Bibliography

Aarno, D., Sommerfeld, J., Kragic, D., Pugeault, N., Kalkan, S., Wörgötter, F., Kraft, D., and Krüger, N. (2007). Model-independent grasping initializing object-model learning in a cognitive architecture. *IEEE International Conference on Robotics and Automation, ICRA07, Workshop: From features to actions - Unifying perspectives in computational and robot vision*. 16, 144, 146, 150, 152

Aloimonos, Y. and Shulman, D. (1989). *Integration of Visual Modules — An Extension of the Marr Paradigm*. Academic Press, London. 14

Amir, A. and Lindenbaum, M. (1998). A generic grouping algorithm and its quantitative analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):168–185. 49, 50, 51

Ayache, N. and Faverjon, B. (1987). Efficient registration of stereo images by matching graph descriptions of edge segments. *International Journal of Computer Vision*, pages 107–131. 69, 71

Baker, H. and Binford, T. (1981). Depth from edge and intensity based stereo. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, volume 2, pages 631–636. 71

Barlow, H., Blakemore, C., and Pettigrew, J. (1967). The neural mechanisms of binocular depth discrimination. *Journal of Physiology (London)*, 193:327–342. 16, 47

Barron, J., Fleet, D., and Beauchemin, S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77. 14, 44

Baumberg, A. (2000). Reliable feature matching across widely separated views. In *Proceedings of the International Conference on Pattern Recognition*, pages 774–781. 24, 26

Borst, C., Fischer, M., and Hirzinger, G. (1999). A fast and robust grasp planner for arbitrary 3D objects. In *Ieee international conference on robotics and automation*, pages 1890–1896, Detroit, Michigan. 145

Bouguet, J.-Y. (2007). Camera Calibration Toolbox for Matlab. `http://www.vision.caltech.edu/bouguetj/calib_doc/`. 166

Boyer, K. and Kak, A. (1988). Structural stereopsis for 3–D vision. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 10(2):144–166. 97

Bregler, C. and Malik, J. (1998). Tracking people with twists and exponential maps. *IEEE computer Society conference on Computer Vision and Pattern Recognition*, pages pp.8–15. 116

Brunswik, E. and Kamiya, J. (1953). Ecological Cue–Validity of 'Proximity' and of Gther Gestalt Factors. *American Journal of Psychology*, 66(1):20–32. 49

Burns, J., Weiss, R., and Riseman, E. (1992). The Non–Existence of General–Case View–Invariants. In J.L. Mundy and A. Zisserman, editor, *Geometric Invariance in Computer Vision*, pages 120–131. The MIT Press. 25, 69, 72

C. Schmid and R. Mohr and C. Baukhage (2000). Evaluation of Interest Point Detectors. *International Journal of Computer Vision*, 37(2):151–172. 22

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6). 24

Cham, T.-J. and Cipolla, R. (1996). Geometric Saliency of Curve Correspondances and Grouping of Symmetric Contours. In *Proceedings of the ECCV*, volume 1, pages 385–398. 50, 51

Chung, R. and Nevatia, R. (1991). Use of monocular groupings and occlusion analysis in a hierarchical stereo system. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 50–56. 98

Chung, R. and Nevatia, R. (1995). Use of monucular groupings and occlusion analysis in a hierarchical stereo system. *Computer Vision and Image Understanding*, 62(3):245–268. 98

Coxeter, H. (1969). *Introduction to Geometry (2nd ed.)*. Wiley & Sons. 31

Crevier, D. (1999). A probabilistic method for extracting chains of collinear segments. *Computer Vision And Image Understanding*, 76(1):36–53. 50

Denett, D. (1984). Cognitive wheels: The frame problem of AI. In Hookway, C., editor, *Minds, Machine, and Evolution*, pages 129–151. Cambridge University Press. 15

DrivSco (2006). DrivSco: learning to emulate perception–action cycles in a driving school scenario. FP6-IST-FET, contract 016276-2. 16, 150, 152, 154

ECOVISION (2003). Artificial visual systems based on early-cognitive cortical processing (EU–Project). *http://www.pspc.dibe.unige.it/ecovision/project.html*. 16

Elder, J. and Goldberg, R. (1998). Inferential reliability of contour grouping cues in natural images. *Perception Supplement*, 27. 49

Elder, J. and Goldberg, R. (2002). Ecological statistics of Gestalt laws for the perceptual organization of contours. *Journal of Vision*, 2(4):324–353. 17, 50, 51

Elder, J., Krupnik, A., and Johnstone, L. (2003). Contour grouping with prior models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):661–674. 51

Elder, J. H. (1999). Are edges incomplete ? *International Journal of Computer Vision*, 34:97–122. 24, 45

Elder, J. H. and Zucker, S. W. (1998). Local scale control for edge detection and blur estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):699–716. 23, 37

Faugeras, O. (1993). *Three–Dimensional Computer Vision*. MIT Press. 11, 67, 69, 72, 114, 129, 159, 167

Faugeras, O. and Hébert, M. (1986). The Representation, Recognition, and Locating of 3–D Objects. *International Journal of Robotics Research*, 5(3):27–52. 114

Faugeras, O. and Robert, L. (1996). What can two images tell us about the third one? *International Journal of Computer Vision*, 18(1). 135

Felsberg, M. (2002). *Low-Level Image Processing with the Structure Multivector*. PhD thesis, Institute of Computer Science and Applied Mathematics, Christian-Albrechts-University of Kiel. 30

Felsberg, M., Duits, R., and Florack, L. (2005). The monogenic scale space on a rectangular domain and its features. *International Journal of Computer Vision*, 64(2/3):187–201. 22

Felsberg, M., Kalkan, S., and Krüger, N. (2006). Continuous characterization of image structures of different dimensionality. *IEEE Transactions on Image Processing (submitted)*. submitted. 30, 32

Felsberg, M. and Krüger, N. (2003). A probablistic definition of intrinsic dimensionality for images. *Pattern Recognition, 24th DAGM Symposium*. 30, 31

Felsberg, M. and Sommer, G. (2001). The monogenic signal. *IEEE Transactions on Signal Processing*, 49(12):3136–3144. 27, 34

Field, D. J., Hayes, A., and Hess, R. F. (1993). Contour integration by the human visual system: Evidence for a local "association field". *Vision Research*, 33(2):173–193. 48, 51

Fischler, R. and Bolles, M. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):619–638. 114, 128

Fitzpatrick, P. and Metta, G. (2003). Grounding vision through experimental manipulation. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 361:2165 – 2185. 145

Fitzpatrick, P., Metta, G., Natale, L., Rao, S., and Sandini, G. (2003). Learning About Objects Through Action - Initial Steps Towards Artificial Cognition. In *IEEE Int. Conf on Robotics and Automation*, pages 3140–3145. 145

Geib, C., Mourao, K., Petrick, R., Pugeault, N., Steedman, M., Krüger, N., and Wörgötter, F. (2006). Object action complexes as an interface for planning and robot control. *Workshop 'Toward Cognitive Humanoid Robots' at IEEE-RAS International Conference on Humanoid Robots (Humanoids 2006)*. 146, 150

Geisler, W., Perry, J., Super, B., and Gallogly, D. (2001). Edge Co–occurrence in Natural Images Predicts Contour Grouping Performance. *Vision Research*, 41:711–724. 17, 49

Geißler, P. (1999). Imaging optics. In Jähne, B., Haußecker, H., and Geißler, P., editors, *Handbook of computer vision and applications*, volume 1, pages 63–101. Academic Press. 166

Granlund, G. H. and Knutsson, H. (1995). *Signal Processing for Computer Vision*. Kluwer Academic Publishers, Dordrecht. 27

Grimson, W. (1985). Computational experiments with a feature based stereo algorithm. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 7:17–24. 69

Grimson, W. (1993). Why stereo vision is not always about 3D reconstruction. Technical Report AIM-1435, Massachusetts Institute of Technology. 24, 67

Guy, G. and Medioni, G. (1996). Inferring global perceptual contours from local features. *International Journal of Computer Vision*, 20(1–2):113–133. 51

Harris, C. G. and Stephens, M. (1988). A combined corner and edge detector. In *4th Alvey Vision Conference*, pages 147–151. 23

Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision.* Cambridge University Press. 11, 67, 69, 114, 174

Herman, M. and Kanade, T. (1986). Incremental reconstruction of 3D scenes from multiple complex images. *Artificial Intelligence*, 30:289–341. 97

Hoff, W. and Ahuja, N. (1989). Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 11(2):121–136. 98

Horaud, R. and Skordas, T. (1989). Stereo correspondences through feature grouping and maximal cliques. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 11(11). 71, 97

Horn, B., editor (1986). *Robot Vision*. MIT Press. 16, 36

Hubel, D. and Wiesel, T. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *J. Physiology*, 160:106–154. 16, 22, 47

Hubel, D. and Wiesel, T. (1968). Receptive fields and functional architecture of monkey striate cortex. *J. Physiology*, 195(1):215–243. 22

Hubel, D. and Wiesel, T. (1969). Anatomical demonstration of columns in the monkey striate cortex. *Nature*, 221:747–750. 16, 21, 24, 47

Irani, M. and Anandan, P. (2000). About direct methods. In *ICCV'99: proceedings of the International Workshop on Vision Algorithms*, pages 267–277, London, UK. Springer-Verlag. 22

Jähne, B. (1997). *Digital Image Processing – Concepts, Algorithms, and Scientific Applications*. Springer. 27, 30

Jones, J. and Palmer, L. (1987). An evaluation of the two dimensional Gabor filter model of simple receptive fields in striate cortex. *Journal of Neurophysiology*, 58(6):1223–1258. 47

Kalkan, S., Calow, D., Wörgötter, F., Lappe, M., and Krüger, N. (2005). Local image structures and optic flow estimation. *Network: Computation in Neural Systems*, 16(4):341–356. 36

Kalkan, S., Wörgötter, F., and Krüger, N. (2007a). Depth prediction at homogeneous image structures. Technical Report 2, Robotics Group, Maersk Institute, University of Southern Denmark. 95, 153

Kalkan, S., Yan, S., Pilz, F., and Krüger, N. (2007b). Improving junction detection by semantic interpretation. In *International Conference on Computer Vision Theory and Applications (VISAPP)*. 153

Kim, N. H. and Bovik, A. C. (1988). A contour–based stereo matching algorithm using disparity continuity. *Pattern Recognition*, 21(5):505–514. 69, 71, 108

Koenderink, J. J. and van Doorn, A. J. (1987). Representation of Local Geometry in the Visual System. *Biological Cybernetics*, 55(6):367–375. 26

Koffka, K. (1935). *Principles of Gestalt Psychology*. Lund Humphries, London. 48

Köhler, K. (1947). *Gestalt Psychology: An introduction to new concepts in psychology*. New York: Liveright. 48

König, P. and Krüger, N. (2006). Perspectives: Symbols as self-emergent entities in an optimization process of feature extraction and predictions. *Biological Cybernetics*, 94(4):325–334. 17

Kovesi, P. (1999). Image features from phase congruency. *Videre: Journal of Computer Vision Research*, 1(3):1–26. 24, 26, 27, 69

Krüger, N. (1998a). Collinearity and parallelism are statistically significant second order relations of complex cell responses. *Neural Processing Letters*, 8(2):117–129. 17

Krüger, N. (1998b). Collinearity and parallelism are statistically significant second order relations of complex cell responses. *Proceedings of I&ANN 98*. 49

Krüger, N. and Felsberg, M. (2003). A continuous formulation of intrinsic dimension. *Proceedings of the British Machine Vision Conference*, pages 261–270. 30

Krüger, N. and Felsberg, M. (2004). An explicit and compact coding of geometric and structural information applied to stereo matching. *Pattern Recognition Letters*, 25(8):849–863. 29

Krüger, N., Hulle, M. V., and Wörgötter, F. (accepted). Ecovision: Challenges in early-cognitive vision. *International Journal of Computer Vision*. 16

Krüger, N., Lappe, M., and Wörgötter, F. (2004). Biologically motivated multi-modal processing of visual primitives. *Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, 1(5):417–428. 17, 26, 47

Krüger, N., Pugeault, N., and Wörgötter, F. (2007). Multi–modal primitives: Local, condensed, and semantically rich visual descriptors and the formalisation of contextual information. Submitted. 17, 21, 26, 37, 40, 45, 153

Krüger, N. and Wörgötter, F. (2002). Multi modal estimation of collinearity and parallelism in natural image sequences. *Network: Computation in Neural Systems*, 13:553–576. 50

Krüger, N. and Wörgötter, F. (2004). Statistical and deterministic regularities: Utilisation of motion and grouping in biological and artificial visual systems. *Advances in Imaging and Electron Physics*, 131:82–147. 15, 16, 115, 116

Lades, M., Vorbrüggen, J., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R., and Konen, W. (1993). Distortion invariant object recognition in the dynamik link architecture. *IEEE Transactions on Computers*, 42(3):300–311. 16

Lee, S.-H. and Leou, J.-J. (1994). A dynamic programming approach to line segment matching in stereo vision. *Pattern Recognition*, 27(8):961–986. 69, 71

Lennie, P. (2000). *Principles of Neural Science (4th edition)*, chapter Color Vision. McGraw Hill. 23

Lindeberg, T. (1998a). Edge detection and ridge detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):117–156. 22, 23, 29, 37

Lindeberg, T. (1998b). Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116. 22

Liu, B., Yu, M., Maier, D., and Männer, R. (2005). An efficient and accurate method for 3D–point reconstruction from multiple views. *International Journal of Computer Vision*, 65(3):175–188. 72, 87

Lorusso, A., Eggert, D., and Fisher, R. (1995). A comparison of four algorithms for estimating 3-d rigid transformations. In *Proceedings of the british machine vision conference*. 114

Lowe, D. (1987). Three–dimensional object recognition from single two images. *Artificial Intelligence*, 31(3):355–395. 51, 145

Lowe, D. G. (2004). Distinctive Image Features from Scale–Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110. 24, 25

Mandelbaum, R., Salgian, G., and Sawhney, H. (1999). Correlation–based estimation of ego–motion and structure from motion and stereo. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 544–550. 135

Marr, D. (1982). *Vision*. Freeman. 12, 16, 24, 48, 67

Marr, D. and Poggio, T. (1976). Cooperative computation of stereo disparity. *Science*, 194(4262):283–287. 71

Maxwell, E. (1951). *General homogeneous coordinates in space of three dimensions*. Cambridge University Press. 159, 160, 161, 162, 163

Mayhew, J. E. and Frisby, J. P. (1981). Psychophysical and computational studies towards a theory of human stereopsis. *Artificial Intelligence*, 17:349–385. 68, 69, 71, 97, 108

Mikolajczyk, K. and Schmid, C. (2005). A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630. 14, 22, 24, 25

Miller, A., Knoop, S., Christensen, H., and Allen, P. (2003). Automatic grasp planning using shape primitives. In *Ieee international conference on robotics and automation*, volume 2, pages 1824–1829. 145

Mohan, R., Medioni, G., and Nevatia, R. (1989). Stereo Error Detection, Correction, and Evaluation. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 11(2):113–120. 97, 108

Mundy, J. L., Liu, A., Pillow, N., Zisserman, A., Abdallah, S., Utcke, S., Nayar, S., and Rothwell, C. (1996). An experimental comparison of appearance and geometric model based recognition. In *Object Representation in Computer Vision*, pages 247–269. 83

Murray, R., Li, Z., and Sastry, S. (1994). *A mathematical introduction to Robotic Manipulation*. CRC Press. 117

Mykolajczyk, K. and Schmid, C. (2004). Scale & affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86. 14, 22, 24

Nagel, H.-H. and Enkelmann, W. (1986). An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:565–593. 36

Ogale, A. and Aloimonos, Y. (2006). A roadmap to the integration of early visual modules. *International Journal of Computer Vision*. 24, 69

Ohta, Y. and Kanade, T. (1985). Stereo by intra– and inter–scanline search using dynamic programming. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 7(2). 71, 97

Oram, M. and Perrett, D. (1994). Modeling visual recognition from neurobiological constraints. *Neural Networks*, 7:945–972. 16

PACO-PLUS (2006). PACO–PLUS: perception, action and cognition through learning of object-action complexes. Integrated Project (IST-FP6-IP-027657). 16, 150, 154

Palus, H. (1998). Colour spaces. In Sangwine, S. and Horne, R., editors, *The Colour Image Processing Handbook*, pages 67–90. Chapman & Hall. 43

Parent, P. and Zucker, S. (1989). Trace inference, curvature consistency, and curve detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(8):823–839. 50, 51

Parker, A. and Cumming, B. (2001). Cortical mechanisms of binocular stereoscopic vision. *Prog Brain Res*, 134:205–16. 47

Pauwels, K. and Van Hulle, M. M. (2004). Segmenting Independently Moving Objects from Egomotion Flow Fields. In *Proceedings of the Early Cognitive Vision Workshop*. 153

Perona, P. and Freeman, W. (1998). A Factorization Approach to Grouping. In *Proceedings of the ECCV*, volume 1406. 50, 51

Phong, T., Horaud, R., Yassine, A., and Tao, P. (1995). Object pose from 2-D to 3-D point and line correspondences. *International Journal of Computer Vision*, 15:225–243. 114

Pritchett, P. and Zisserman, A. (1998). Wide baseline stereo matching. In *Proceedings of the International Conference on Computer Vision*, pages 754–760. 22

Pugeault, N., Baseski, E., Kraft, D., Wörgötter, F., and Krüger, N. (2007a). Extraction of multi–modal object representations in a robot vision system. In *Robot Vision Workshop at the Int. Conf. on Computer Vision Theory and Applications (VISAPP)*, pages 126–135. 16, 136, 144, 146, 152

Pugeault, N., Wörgötter, F., , and Krüger, N. (2006a). Rigid body motion in an early cognitive vision framework. In *Proceedings of the IEEE Systems, Man and Cybernetics Society Conference on Advances in Cybernetic Systems*. 16

Pugeault, N., Wörgötter, F., , and Krüger, N. (2007b). Extraction of structural visual events. In *Technical report of the Robotics Group, Maersk Institute, University of Southern Denmark*, number 1. 152

Pugeault, N., Wörgötter, F., and Krüger, N. (2006b). Multi-modal scene reconstruction using perceptual grouping constraints. In *Proceedings of the IEEE Workshop on Perceptual Organization in Computer Vision (in conjunction with CVPR'06)*. 21, 139

Rosenhahn, B. (2003). *Pose Estimation Revisited (PhD Thesis)*. Institut für Informatik und praktische Mathematik, Chrsitian–Albrechts–Universität Kiel. 116

Rosenhahn, B., Granert, O., and Sommer, G. (2001a). Monocular pose estimation of kinematic chains. In Dorst, L., Doran, C., and Lasenby, J., editors, *Applied Geometric Algebras for Computer Science and Engineering*, pages 373–383. Birkhäuser Verlag. 116, 119

Rosenhahn, B., Krüger, N., Rabsch, T., and Sommer, G. (2001b). Automatic tracking with a novel pose estimation algorithm. *Robot Vision 2001*. 116

Sabatini, S., Gastaldi, G., Solari, F., Pauwels, K., van Hulle, M., Diaz, J., Ross, E., Pugeault, N., and Krüger, N. (2007). Compact (and accurate) early vision processing in the harmonic space. In *International Conference on Computer Vision Theory and Applications (VISAPP)*. 27

Sangwine, S. and Horne, R., editors (1998). *The Colour Image Processing Handbook*. Chapman & Hall. 36

Sarkar, S. and Soundararajan, P. (2000). Supervised learning of large perceptual organization: Graph spectral partitioning and learning automata. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(5):504–525. 50, 51

Schaffalitzky, F. and Zisserman, A. (2002). Multi–view matching for unordered image sets, or "how do I organize my holiday snaps?". *Lecture Notes in Computer Science*, 2350:414–431. in Proceedings of the BMVC02. 26

Scharstein, D. and Szeliski, R. (1998). Stereo Matching with Nonlinear Diffusion. *International Journal of Computer Vision*, 28(2):155–174. 71

Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two–frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1/2/3):7–42. 14, 77, 78

Schiele, B. and Crowley, J. (1996). Probabilistic object recognition using multidimensional receptive field histograms. *Advances in Neural Information Processing Systems*, 8:865–871. 16

Schmid, C. and Zisserman, A. (1997). Automatic line matching across views. *Proc. IEEE Conference on Computer Vision and Pattern Recognition,*, pages 666–671. 69

Schmid, C. and Zisserman, A. (2000). The geometry and matching of lines and curves. *International Journal of Computer Vision*, 40(3):199–233. 69

Sha'ashua, A. and Ullman, S. (1990). Grouping contours by iterated pairing network. In *Neural Information Processing Systems (NIPS)*, volume 3. 50

Shevelev, I., Lazareva, N., Tikhomirov, A., and Sharev, G. (1995). Sensitivity to cross–like figures in the cat striate neurons. *Neuroscience*, 61:965–973. 16, 47

Shevlin, F. (1998). Analysis of orientation problems using Plücker lines. *International Conference on Pattern Recognition, Bisbane*, 1:65–689. 114

Shi, J. and Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905. 50

Strecha, C. and van Gool, L. (2002). Motion - stereo integration for depth estimation. In *Proceedings of the European Conference on Computer Vision*, volume 2351 of *Lecture Notes in Computer Science*, pages 170–185. 135

Sun, J., Shum, H.-Y., and Zheng, N.-N. (2002). Stereo matching using belief propagation. In *Proceedings of the European Conference on Computer Vision*, volume LNCS 2351, pages 510–524. 71

Tao, H., Sawhney, H., and Kumar, R. (2001). A Global Matching Framework for Stereo Computation. In *Proceedings of the Eightth IEEE International Conference on Computer Vision*, volume 1, pages 532–539. 135

Tarantola, A. (2005). *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society of Industrial and Applied Mathematics (SIAM), Philadelphia. 12

Tony Lindeberg (1994). Scale space theory: A basic tool for analysing structures at different scales. *Journal of Applied Statistics*, 21(2):225–270. 22

Torr, P. H. S. and Zisserman, A. (2000). Feature based methods for structure and motion estimation. In *Iccv '99: proceedings of the international workshop on vision algorithms*, pages 278–294, London, UK. Springer-Verlag. 24

Ullman, S. (1976). The interpretation of structure from motion. In *MIT AI Memo*. 51

van Gool, L., Moons, T., and Ungureanu, D. (1996). Affine / Photometric Invariants for Planar Intensity Patterns. *Lecture Notes In Computer Science*, 1064:642–651. in Proceedings of the 4th European Conference on Computer Vision — Volume 1. 26

Venkateswar, V. and Chellappa, R. (1995). Hierarchical stereo and motion correspondence using feature grouping. *International Journal of Computer Vision*, 15:245–269. 98

Wertheimer, M., editor (1935). *Laws of Organsation in Perceptual Forms*. Harcourt & Brace & Javanowitch, London. 48

Wettegren, B., Christensen, L., Rosenhahn, B., Granert, O., and Krüger, N. (2005). Image uncertainty and pose estimation in 3d euclidian space. *Proceedings DSAGM 13th Danish Conference on Pattern Recognition and Image Analysis, DSAGM 2005*, pages 76–84. 120

Wikipedia (2007). Cubic Hermite Spline. 61

Wolff, L. (1989). Accurate measurements of orientation from stereo using line correspondence. In *Proceedings of the IEEE Computer Vision and Pattern Recognition conference*. 72, 84, 103, 140

Wörgötter, F., Krüger, N., Pugeault, N., Calow, D., Lappe, M., Pauwels, K., Hulle, M. V., Tan, S., and Johnston, A. (2004). Early cognitive vision: Using gestalt-laws for task-dependent, active image-processing. *Natural Computing*, 3(3):293–321. 15

Wurtz, R. and Kandel, E. (2000a). *Principles of neural science (4th edition)*, chapter Central visual pathways, pages 523–547. McGraw Hill. 47

Wurtz, R. and Kandel, E. (2000b). *Principles of Neural Science (4th edition)*, chapter Perception of motion, depth and form, pages 548–571. McGraw Hill. 47

Zetzsche, C. and Barth, E. (1990). Fundamental limits of linear filters in the visual processing of two dimensional signals. *Vision Research*, 30. 30

Zhang, L., Curless, B., and Seitz, S. (2003). Spacetime Stereo: Shape Recovery for Dynamic Scenes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 367–374. 135

Zhang, Z., Deriche, R., Faugeras, O., and Luong, Q.-T. (1995). A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial Intelligence*, 78:87–119. 24

Ziou, D. and Tabbone, S. (1998). Edge detection techniques — an overview. *International Journal of Pattern Recognition and Image Analysis*, 8:537–559. 14

Index

Curriculum Vitae

Personal information

Name	Nicolas Pugeault
Birth date	3^{rd} of May 1978
Birth place	Strasbourg (France)
Nationality	French

Academic history

1996–2001	Dipl. Engineer from the ESIEA, Paris.
2001–2002	M. Sc. in Computational Intelligence, University of Plymouth, UK.
2002–2005	Research Assistant at the Departement of Psychology, University of Stirling, UK.
2005–2007	Research Associate at the School of Informatics, University of Edinburgh, UK.