Multi-modal Statistics of Local Image Structures and its Applications for Depth Prediction

Dissertation

zur Erlangung des mathematisch-naturwissenschaftlichen Doktorgrades

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

vorgelegt von

Sinan Kalkan

aus Ankara

Göttingen 2007

Referentin/Referent: Prof. Florentin Wörgötter

Koreferentin/Koreferent: Prof. Norbert Krüger

Tag der mündlichen Prüfung: 15 Januar 2008

Abstract

Processing in most artificial vision systems and in the human vision system starts with early vision which involves the extraction of local visual modalities (like optical flow, disparity and contrast transition etc.) and local image structures (edge-like, junction-like and texture-like structures). Since information in early vision is processed only locally, it is inherently ambiguous. For example, estimation of optical flow faces the aperture problem, and thus, only the flow along the intensity gradient is computable for edge-like structures. Moreover, the extracted flow information at weakly-textured image areas are unreliable. Analogously, stereopsis needs to deal with the correspondence problem: as correspondences at weakly textured image areas cannot be found, the disparity information at such places is not accurate. One way to deal with the missing and ambiguous information is to make use of the redundancy of visual information by exploiting the statistical regularities of natural scenes. Such regularities are carried in the visual system using feedback mechanisms between different layers, or by lateral connections within a layer.

This thesis is interested in the ambiguities and the biased and missing information in the processing of optic flow, stereo and junctions using statistical means. It uses statistical properties of images to analyze the extent of the ambiguous processing in optical flow estimation and whether the missing information in stereo can be recovered using interpolation of depth information at edge-like structures. Moreover, it proposes a feedback mechanism for dealing with the bias in junction detection, and another model for recovering the missing depth information in stereo computation using only the depth information at the edges.

Acknowledgements

First of all, I would like to thank my *unofficial* supervisor Prof. Norbert Krüger from Denmark for his supervision and his important contributions to my scientific thinking. Moreover, it is his understanding and friendly support about my personal requirements that kept this study going.

Prof. Florentin Wörgötter, my official supervisor, is an important ingredient of this thesis. He, being the official supervisor, allowed me to perform my research independent of a physical location. I also learned a lot from him regarding management of a big research group whose members are from totally different fields, working in different subjects.

This study is a product of working in three different locations: Stirling from Scotland; Odense, Copenhagen and Esbjerg from Denmark; and, Göttingen. I would like to thank all my friends and colleagues from these countries, some of which are (ordered with surname): Emre Başeski, Babette Dellen, Tao Geng, Matthias Hennig, Christoph Kolodziejski, Dirk Kraft, Irene Markelić, Ailsa Millen, Florian Pilz, Yan Shi, Steffen Wischmann and Alexander Wolf. I also should mention that kicker (*i.e.*, table football) was an amusing part of my daily life in Göttingen: thank you the Kicker Gang!

My friends and colleagues Nicolas Pugeault, Marina Wimmer and Ailsa Millen from Stirling were really important for this study. It is them who managed to make Scotland feel like a second home for me. Tomas Kulvičius is also to be thanked for leaving Scotland with me for Göttingen, and going through a tough beginning in Germany.

I'd like to thank my friends from Turkey whose remote and cyber friendship reserves a big credit, as going abroad never meant leaving them behind: İrem Aktuğ, Barış Sertkaya, Ergül Pekesen, Gülhan Bilen, Levent Karagöl, Gökhan Kars, Behiye Erkenci, Burçin Sapaz, Sevgi Yaşar. Special credits go to

Esin Saka whose support cannot be thanked with words.

I would not know how to thank Gökçe Yıldırım, either. Without her kind understanding and patience, I would not dare to stay abroad longer than one year.

At last but not the least, I would like to thank my family whose mere existence is more than anything.

Contents

Chapter 1

Introduction

Vision is the process of understanding scenes from their 2D projections, which are in the form of a set of images. The intensity values in an image are formed by one or more of the following factors: (1) the geometry, and (2) the illumination of the environment, (3) the reflectances of the surfaces, and (4) the viewpoint. By definition, this makes vision an *ill-posed*[1] *inverse* problem [Bertero et al., 1987].

Processing in most artificial vision systems and in the human vision system starts with the extraction of local visual modalities (like optical flow, disparity and contrast transition etc.) and local image structures (edge-like, junction-like and texture-like structures). This stage is called *early vision* in, *e.g.*, [Papathomas et al., 1995]. Since information in early vision is processed only localy, it is inherently ambigious. For example, estimation of optical flow faces the aperture problem, and thus, only the flow along the intensity gradient is computable for edge-like structures. Moreover, the extracted flow information at weakly-textured image areas are unreliable. Analogously, stereopsis needs to deal with the correspondence problem: as correspondences at weakly textured image areas cannot be found, the disparity information at such places is not accurate. Nonetheless, the human visual system can extract meaningful 3D interpretations from early vision in spite of the ambiguities and the missing information. Accordingly, an artificial vision system is expected to operate and create 3D world models from such information, too.

The ambiguous and the biased information from early vision is processed and integrated by global mechanisms at a stage of *early cognitive vision* (as defined in [Wörgötter et al., 2004]) in order to create

[1] According to [Hadamard, 1923], a problem is well-posed if (1) a solution exists, (2) the solution is unique, and (3) it depends continuously on the data. A problem is ill-posed if it is not well-posed.

more accurate and meaningful visual entities. At this stage, the visual information is disambiguated by recurrent loops and attention, and by feedback from higher visual processing layers. Among others, such feedback mechanisms are realized for edge-like structures in [Pugeault et al., 2006]. To realize such feedback mechanisms, it is argued in [Krüger et al., 2004b] that a transformation of the local signal to a more *condensed* representation in terms of a semantic descriptor of a reduced dimensionality is required; *i.e.*, feedback mechanisms should make use of sparse symbolic descriptors rather than the signal-level information.

Homogeneous image areas are signals of uniform intensity. Such image areas are neglected in early vision since retinal ganglion cells are excitable only by contrast differences. Early cognitive vision is believed to *infer* visual information (including first estimates of depth information) at homogeneous image areas from the available visual information in early vision using interpolation mechanisms[2]. There are already psychophysical experiments [Anderson et al., 2002, Collett, 1985, Julesz, 1971, Treue et al., 1995] and computational theories [Barrow and Tenenbaum, 1981, Grimson, 1982, Terzopoulos, 1988] which suggest that the human visual system performs interpolation in depth and completes the missing depth information at weakly-structured image areas.

Feedback mechanisms in a vision system make use of the regularities in the input images. In fact, it is believed that the human visual system is adapted to the statistics of the retinal projections of the environment, in order to make use of the regularities or the redundancy of information in the environment [Brunswik and Kamiya, 1953]. With the availability of computational and technological means, it has been possible to prove such claims [Krueger, 1998, Geisler et al., 2001], and the results of such investigations have proven to be useful in several computational vision problems [Elder et al., 2003, Pugeault et al., 2004, Zhu, 1999] (see [Simoncelli, 2003] for a review).

As a summary, biological vision systems can cope with the ambiguities and the missing information mentioned above by (1) exploiting the redundancy of information in the natural images, (2) using feedback information from higher visual levels and (3) using lateral feedback information between different visual modalities (such as optical flow, colour, contrast etc.), for example, in the form of an interpolation process.

This thesis is concerned with the analysis of ambiguities in the visual modalities such as optical

[2]The term *interpolation* is not meant in mathematical terms (*i.e.*, regression) in this thesis, and filling-in missing information is usually called interpolation in the literature (see, *e.g.*, [Grimson, 1982])

flow and disparity, and with the computational modeling of feedback mechanisms for two problems: (1) reliable and complete extraction of junctions in images (chapter 4), and (2) estimation of depth at weakly-structured image areas (chapter 7) where correspondence-based depth cues provide unreliable information or no information at all (see [Bayerl and Neumann, 2007] for a computational model of feedback mechanisms in optical flow estimation). Statistical investigations from natural images and chromatic range data are provided that support the models developed in this work and the previous works from other researchers and quantify some widely made assumptions by the vision community (chapters 3, 5 and 6).

This thesis contributes to an existing early cognitive vision framework in two aspects: (1) Junctions with condensed symbolic descriptors. (2) Homogeneous image patches with predicted depth information. This early cognitive vision framework is mainly developed in the European ECOVISION project [ECOVISION, 2003], which so far makes use of only edge-like structures [Pugeault et al., 2006]. By having depth information available in this framework, homogeneous image patches can be combined to create object surfaces which then can be used for several tasks such as grasping objects using a robot arm (European PACO-PLUS project [PACO-PLUS, 2007]), or driving a car on the road (European DRIVSCO project [Drivsco, 2007]).

In the following sections, the thesis is put into several contexts, describing the contributions of the thesis in every context.

1.1 Marr's Theory of Vision

Vision research has been influenced most by David Marr's paradigm [Marr, 1982]. This is because the paradigm (1) layed down the computational vision as an information processing task, (2) addressed the main problems that had to be solved in order to achieve this processing task, and (3) proposed a computational framework as a solution to it.

One of the first contributions of Marr was to combine the findings and the theories of his time from neurophysiology, psychology and artificial intelligence into a coherent and complete vision theory. He clearly defined vision as an information-processing task, and in combination with the existing psychophysical experiments, he could arrive at a distinction between (1) the *computational theory*, (2) the *representation and algorithmic implementation* of a theory, and (3) the *hardware* implementation (for

example, in a computer or in a biological neural mechanism). In the context of vision, he proposed what these three levels are, binding together the evidences and the computer vision theories at that time.

Marr proposed the following *representational* framework for deriving the 3D shape information:

1. *Image.* The input to the system, which is a 2D projection of the scene.

2. *Primal sketch.* The sketch of the image is extracted in terms of edges, corners and other local structures as well as perceptual groups.

3. $2\frac{1}{2}$*-D sketch.* This level is viewer-based and concerned with extracting the relative or absolute 3D distances and orientations of objects.

4. *3-D mode representation.* This is the goal of a *complete* visual system. It includes models of objects, in an object-centered coordinate system, as well as how these objects are organized in space.

Marr reduced the vision process into a set of *subproblems* that are called *visual modules*. The visual modules include stereo, shape-from-X methods, extraction of several visual modalities like optical flow, contrast transition etc. However, in the last decade, (1) the evidences from neurophysiology that biological visual systems are equiped with feedback mechanisms which constitute an important proportion of the visual cortex, and (2) the ambiguities and missing information in early vision led scientists to realize that visual modules cannot be solved *unambiguously* without feedback from other visual modules or from higher levels of visual processing, and several attempts have been initiated for combining the different visual modules (see, *e.g.*, [Aloimonos and Shulman, 1989]).

In this context, this thesis contributes two feedback mechanisms for two different tasks. First, a simple feedback mechanism is proposed for the detection and extraction of junctions using their semantic interpretation (chapter 4). The semantic interpretation of junctions is used to detect and remove outliers and produce very reliable detections in spite of high sensitivity to contrast. Second, 3D features, which are extracted from a feature-based stereo algorithm, are used in a depth prediction model to laterally feedback and interpolate depth in homogeneous image areas where correspondence-based methods usually fail to compute depth (chapter 7).

1.2 Early Vision and Early Cognitive Vision

According to Marr's paradigm (see section 1.1 and [Marr, 1982]), vision involves extraction of meaningful representations from input images, starting at the pixel level and building up its interpretation more or less in the following order: local filters, extraction of relevant features, the $2\frac{1}{2}$-D sketch and the 3-D sketch. One possible distinction of image structures are as described below:

- Homogeneous structures: Homogeneous patches are signals of uniform intensities. It is assumed that they correspond to continuous surfaces (which is quantified in chapter 5), and they are not much made use of in early vision because retinal ganglion cells are not excitable by homogeneous intensities [Bruce et al., 2003].

- Edge–like structures: Edges are low-level structures which constitute the boundaries between homogeneous or texture-like image areas (see, *e.g.*, [Koenderink and Dorn, 1982, Marr, 1982] for their importance in vision). Detection of edge-like structures in the human visual system starts with orientation sensitive cells in V1 [Hubel and Wiesel, 1969], and biological and machine vision systems depend on their reliable extraction and utilization [Marr, 1982, Koenderink and Dorn, 1982].

- Junction–like structures: Junctions are image patches where two or more edge-like structures with significantly different orientations intersect (see, *e.g.*, [Guzman, 1968, Rubin, 2001, Shevelev et al., 2003] for their importance in vision). It has been suggested that the human visual system makes use of them for different tasks like recovery of surface occlusion [Guzman, 1968, Rubin, 2001] and shape interpretation [Malik, 1987, Shevelev et al., 2003]. It is known that junctions are detected in the primary visual cortex (see, *e.g.*, [Shevelev et al., 1998]).

- Texture–like structures: Although there is not a generally-agreed definition, textures are often defined as image patches which consist of repetitive, random or directional structures (for their analysis, extraction and importance in vision, see *e.g.*, [Tuceryan and Jain, 1998]). Our world consists of textures on many surfaces, and the fact that we can reliably reconstruct the 3D structure from any textured environment indicates that the human visual system makes use of and is very good at the analysis and the utilization of textures.

Note that semantic description of these image structures requires different descriptors. For example, for a homogeneous image patch, image orientation is not defined, and a colour value (and possibly size) is sufficient to represent it. However, for describing an edge-like structure, image orientation, contrast transition and three colour values are required [Krüger et al., 2007].

Early vision involves acquisition of a set of visual modalities as well as extraction of the local image structures (*except* for homogeneous image structures). These visual modalities include disparity, optical flow, texture information, occlusions etc. and, together with the local image structures, carry the information necessary to interpret a scene.

Owing to only local processing, early vision usually carries ambigious, biased or false information. For example, the visual modalities face the correspondence problem; *i.e.*, looking for the corresponding image features between the different views of a scene. Due to the correspondence problem, only the optic flow along the intensity gradient of an edge can be found; or, in the case of stereopsis, no disparity can be computed at weakly-structured image areas (see, [Baker et al., 2001]).

The ambiguous and biased information from early vision is processed and integrated by global mechanisms at the stage of *early cognitive vision* (as defined in [Wörgötter et al., 2004]) in order to create more accurate, meaningful and complete visual entities. At this stage, the visual information is disambiguated by recurrent loops, attention and feedback from higher visual processing layers. Moreover, it is our belief that homogeneous image patches that are neglected in early vision are added back to visual processing at this stage.

Physiological evidences [Angelucci et al., 2002, Galuske et al., 2002] as well as computational models [Bayerl and Neumann, 2007, Bullier, 2001] already exist that study and support the usage of feedback mechanisms in the processing of different kinds of visual information.

The contributions of this thesis in the context of early vision and early cognitive vision are:

1. As mentioned already in section 1.1, using a simple feedback mechanism to improve junction detection through semantic interpretation (chapter 4). The extracted interpretation of detected junctions is used to remove outliers and select reliable detections.

2. Analysis of the extent of the ambiguity of visual information in the context of optical flow using natural image statistics (chapter 3).

3. Analysis of the relation between local image structures and local 3D structures. Such an analysis

is important for understanding possible mechanisms underlying interpolation processes (chapters 5 and 6).

4. As mentioned already in section 1.1, proposal of a depth prediction model that uses lateral feedback between 3D features, extracted from a feature-based stereo, to interpolate depth at homogeneous image areas (chapter 7). With this contribution, this thesis is a part of an early cognitive vision framework that so far includes edge features only [Krüger et al., 2003, Pugeault et al., 2006].

1.3 3D Reconstruction, the Correspondence Problem and Depth Interpolation

Depth cues can be classified as pictorial, or monocular (such as shading, utilization of texture gradients or linear perspective) and multi-view (like stereo and structure from motion) [Faugeras, 1993]. Depth cues which make use of multiple views require correspondences between the different 2D views of a scene. In contrast, pictorial cues use statistical and geometrical relations in one image to make statements about the underlying 3D structure.

Finding the correspondences between the different views of a scene means matching image points in one view to image points in other views that might have originated from the same 3D point. Junctions are the most distinctive local image features, which makes them suitable for finding correspondences. So are edge-like structures, unless they are parallel with the epipolar line, in which case correspondences cannot be found. As for homogeneous image areas, the correspondence problem is not solvable or very difficult to solve by direct methods as there is no structure (see, *e.g.*, [Baker et al., 2001] for a systematic evaluation). However, many surfaces have only weak texture or no texture at all. Nevertheless, humans are able to reconstruct 3D information for these surfaces, too. Existing psychophysical experiments (see, *e.g.*, [Anderson et al., 2002, Collett, 1985, Julesz, 1971, Treue et al., 1995]) and computational theories (see, *e.g.*, [Barrow and Tenenbaum, 1981, Grimson, 1982, Terzopoulos, 1988]) suggest that in the human visual system, an *interpolation process* is realized that, starting with the local analysis of edges, corners and textures, computes depth also in areas where correspondences cannot easily be found.

Processing of depth in the human visual system starts with the processing of local image structures (such as edge-like structures, corner-like structures and textures) in V1 [Gallant et al., 1994, Hubel and Wiesel, 1969,

Lee et al., 1998, Shevelev et al., 1998]. These features are utilized in stereo vision, depth from motion, depth from texture gradients and other depth cues, which are localized in different parts of the brain, starting from V1 and involving V2, V3, V4 and MT (see, *e.g.*, [Sereno et al., 2002]).

There exists supporting evidence that depth cues which are not directly based on correspondences evolve rather late in the development of the human visual system. For example, pictorial depth cues are made use of only after approximately 6 months [Kellman and Arterberry, 1998]. This indicates that experience may play an important role in the development of these cues, *i.e.*, that we have to understand depth perception as a statistical learning problem [Knill and Richards, 1996, Purves and Lotto, 2002, Rao et al., 2002]. A step towards such an understanding is the investigation and use of the statistical relations between the local image structures and the underlying 3D structure for each of these depth cues [Knill and Richards, 1996, Purves and Lotto, 2002, Rao et al., 2002].

This thesis distinguishes *depth prediction* from *surface interpolation* because surface interpolation assumes that there is already a dense depth map of the scene available in order to be able to estimate the 3D surface-orientation at points which is then used to complete the missing depth information (see, *e.g.*, [Grimson, 1982, Grimson, 1984, Guy and Medioni, 1994, Lee and Medioni, 1998, Lee et al., 2002, Terzopoulos, 1982, Terzopoulos, 1988]) whereas the understanding of depth prediction in this thesis makes use of only 3D line-orientations at edge-segments which are computed using a feature-based stereo algorithm proposed in [Pugeault and Krüger, 2003].

This thesis, in the context of 3D reconstruction, makes the following contributions:

1. Analysis of the relation between local image structures and local 3D structure which is important for understanding the possible underlying mechanisms of depth interpolation processes (chapters 5 and 6).

2. As already mentioned in sections 1.1 and 1.2, proposal of a depth prediction model that uses lateral feedback between 3D features (extracted from a feature-based stereo) to interpolate depth at homogeneous image areas (chapter 7).

1.4 Vision and Natural Image Statistics

The amount of images that can be observed in nature is a very small subset of the images that can be constructed using arbitrary combinations of intensity values [Field, 1994]. This suggests that the natural images bear intrinsic regularities which are believed to be exploited by our visual system for perceiving the environment (see, *e.g.*, [Krüger and Wörgötter, 2004]), especially for the purpose of resolving ambiguities inherent in local processing of various visual modalities such as optic flow and disparity.

For example, it is widely acknowledged that Gestalt principles for perceptual organization are the results of our visual system's adaptation to the statistical regularities in natural scenes. This hypothesis was first pointed out in [Brunswik and Kamiya, 1953], but could not be tested or justified until 90s due to insufficient computational means. In [Field et al., 1993], computer-generated randomly-oriented data was used to develop a theory of contour grouping in the human visual system, called the *association field*. In 1998, [Krueger, 1998] used natural images instead of computer generated data to prove the relation between grouping mechanisms and the natural image statistics. Such investigations were extended in [Elder and Goldberg, 2002, Geisler et al., 2001, Krüger and Wörgötter, 2002], and the results were utilized in several computer vision tasks, including contour grouping, object recognition and stereo (see, *e.g.*, [Elder et al., 2003, Pugeault et al., 2004, Zhu, 1999]).

Statistical regularities of natural images also helped researches to understand the principles of sensory coding in the early stages of visual processing. It was shown that Independent Component Analysis and Principle Component Analysis of image patches from natural images produce Gabor-wavelet like patterns which are believed to be what the simple cells in V1 of the human visual system are doing (see, *e.g.*, [Jones and Palmer, 1987]).

Availability of relatively cheaper range scanners made it possible to analyze the statistical properties of 3D world together with its 2D image projections. Such analyses are important (1) for quantifying and understanding the assumptions that the vision researchers have been making and (2) for understanding the intrinsic properties of the 3D world. In [Yang and Purves, 2003, Huang et al., 2000, Potetz and Lee, 2003], the correlation between the properties of 3D surfaces (like roughness, 3D orientation, distance, size, curvature etc.) and the intensity of the images are analyzed. Such studies

mainly justify assumptions made by shape from shading studies and confirm that natural scene geometry is quite regular and less complex than luminance images. In [Kalkan et al., 2006], a higher-order representation of the 2D local image patches and the 3D local patches were considered, and the probability of observing a certain kind of 3D structure given its 2D projection is provided (see chapter 5 for details). Moreover, range image statistics allow explanation of several visual illusions [Howe and Purves, 2002, Howe and Purves, 2004].

[Krüger and Wörgötter, 2004] provides a summary of the evidences from developmental psychology which suggest that depth extraction based on statistical regularities used in perceptual organization develops at a later stage than depth extraction based on stereopsis and motion. In particular, it is discussed that perceptual organization based on edge structures are in place after approximately 6 months of visual experience but not before [Kellman and Arterberry, 1998, Spelke, 1993] as also mentioned in the previous section. This suggests that the detection of statistical regularities in visual scenes plays an important role in the establishment of such abilities.

This thesis provides natural image statistics (some of which have already been mentioned in sections 1.2 and 1.3) regarding several visual processing phenomena. Chapter 3 investigates the extent of the aperture problem based on local image structures, and the quality of several optical flow algorithms, using ground truth optical flow. In chapter 5, the relation between local image structures and the underlying local 3D structure is analyzed. Chapter 6 tries to answer whether the depth at homogeneous image areas can be predicted from the depth of edge-like structures. The results provided in chapter 6 are important for understanding the possible mechanisms underlying depth interpolation processes and motivate the depth prediction model provided in chapter 7.

1.5 Outline and Contributions

In this section, the contributions of the thesis are summarized, and the relevant publications of the author are listed.

- **Chapter 2** provides background information about the continuous definition of intrinsic dimensionality that is used throughout the whole thesis for distinguishing between different local image structures. Moreover, this chapter introduces the visual features, called *primitives*, that represent different local image structures.

Relevant publication from the author: [Felsberg et al., 2007b].

- **Chapter 3** analyzes the quality of different optical flow algorithms based on different image structures. This analysis provides insight into the extent of the aperture problem for different image structures. This chapter proposes intrinsic dimensionality as a new tool for better analysis of the inherent properties of optic flow algorithms depending on the local image structures.

  Relevant publications from the author: [Kalkan et al., 2004a, Kalkan et al., 2004b, Kalkan et al., 2005].

- **Chapter 4** discusses the problems of junction detection methods, in relation to their sensitivity to contrast, and proposes a local feedback mechanism for improving the quality of any junction detection method. The feedback comes from the condensed description, *i.e.*, semantic interpretation of the junctions, which is used to differentiate true positives from false positives. The chapter presents results on real examples showing the usefulness of such a feedback mechanism for different junction detection methods.

  Relevant publication from the author: [Kalkan et al., 2007f, Pilz et al., 2007]

- **Chapter 5** uses chromatic range data to investigate the likelihood of observing a certain local 3D structure, given its 2D projection. The results justify a widely used assumption called *'no news is good news'*. This assumption basically states that two image points which do not have any contrast difference in-between can be assumed to be on the same surface. This chapter challenges this assumption by showing that most contrast differences also form continuous surfaces.

  Relevant publications from the author: [Kalkan et al., 2006, Kalkan et al., 2007c]

- **Chapter 6** investigates whether depth at homogeneous image areas can be predicted from the depth of edge-like structures. It shows that an edge segment in the neighborhood of a homogeneous image patch can predict the depth at the homogeneous image patch. The strength of this prediction is shown to decrease with distance and to increase with the existence of a second coplanar edge-segment. This investigation is important for understanding possible mechanisms that might underlie depth interpolation mechanisms.

  Relevant publications from the author: [Kalkan et al., 2007d, Kalkan et al., 2007c]

- **Chapter 7**, motivated from the statistics provided in chapter 6, develops a voting model that predicts depth at homogeneous image areas from the depth of edge-like structures. The model is able

to make prediction in spite of strong outliers in the disparity map. The results are shown to be comparable to several dense stereo algorithms. Moreover, the effect of texture on the performance of the depth prediction model and the dense stereo algorithms is investigated.

Relevant publications from the author: [Kalkan et al., 2007b, Kalkan et al., 2007a, Kjargaard et al., 2007, Kraft et al., 2007, Başeski et al., 2007, Kalkan et al., 2008]

The list of *accepted* publications:

Citation	Year	Journal/Conference Title	Publication Type
[Kalkan et al., 2004a]	2004	Brain Inspired Cognitive Systems	Conference
[Kalkan et al., 2004b]	2004	Dynamic Perception Workshop	Workshop
[Kalkan et al., 2005]	2005	Network: Computation in Neural Systems	Journal
[Kalkan et al., 2006]	2006	IEEE Computer Vision and Pattern Recognition	Conference
[Kalkan et al., 2007d]	2007	Computer Vision Theory and Applications	Conference
[Kalkan et al., 2007f]	2007	Computer Vision Theory and Applications	Conference
[Kalkan et al., 2007b]	2007	Maersk Institute, Uni. of Southern Denmark	Technical Report
[Kalkan et al., 2007a]	2007	Maersk Institute, Uni. of Southern Denmark	Technical Report
[Kjargaard et al., 2007]	2007	Maersk Institute, Uni. of Southern Denmark	Technical Report
[Kalkan et al., 2007c]	2007	Network: Computation in Neural Systems	Journal
[Başeski et al., 2007]	2007	3D Representation for Recognition	Workshop
[Pilz et al., 2007]	2007	Int. Symposium on Visual Computing	Conference
[Kraft et al., 2007]	2007	International Journal of Humanoid Robotics	Journal
[Kalkan et al., 2008]	2008	Computer Vision Theory and Applications	Conference

The list of *submitted/being-written* publications:

Citation	Year	Journal/Conference Title	Publication Type
[Felsberg et al., 2007b]	2007	Image and Vision Computing	Journal

Chapter 2

Background

This chapter presents two crucial tools that are used throughout the thesis. Section 2.1 describes the concept of intrinsic dimensionality, which is used in this thesis for distinguishing between different kinds of local image structures, and section 2.2 briefly introduces the local homogeneous and edge-like features.

2.1 A Continuous Definition of Intrinsic Dimensionality

In image processing, intrinsic dimensionality (*iD*) was introduced by [Zetzsche and Barth, 1990] and was used to formalize a *discrete distinction* between homogeneous, edge-like and junction-like structures. This corresponds to a classical interpretation of local image structures in computer vision.

Homogeneous, edge-like and junction-like structures are respectively classified by *iD* as *intrinsically zero dimensional (i0D)*, *intrinsically one dimensional (i1D)* and *intrinsically two dimensional (i2D)*.

The spectral representation of a local image patch (see figure 2.1(a,b)) reveals that the energy of an i0D signal is concentrated in the origin (figure 2.1(b)-top), the energy of an i1D signal is concentrated along a line (figure 2.1(b)-middle) while the energy of an i2D signal varies in more than one dimension (figure 2.1(b)-bottom).

It has been shown [Felsberg and Krüger, 2003, Krüger and Felsberg, 2003, Felsberg et al., 2007b] that the structure of the *iD* can be understood as a triangle that is spanned by two measures: origin variance and line variance. Origin variance describes the deviation of the energy from a concentration at the origin while line variance describes the deviation from a line structure (see figure 2.1(b) and 2.1(c));

in other words, origin variance measures non-homogeneity of the signal whereas the line variance measures the junctionness. The corners of the triangle then correspond to the 'ideal' cases of *iD*. The surface of the triangle corresponds to signals that carry aspects of the three 'ideal' cases, and the distance from the corners of the triangle indicates the similarity (or dissimilarity) to *ideal* i0D, i1D and i2D signals.

The triangular structure of the intrinsic dimension is counter-intuitive, in the first place, since it realizes a two-dimensional topology in contrast to a linear one-dimensional structure that is expressed in the discrete counting 0, 1 and 2. As shown in [Krüger and Felsberg, 2003, Felsberg and Krüger, 2003, Felsberg et al., 2007b], this triangular interpretation allows for a *continuous formulation* of *iD* in terms of 3 confidences assigned to each discrete case. This is achieved by first computing two measurements of origin and line variance which define a point in the triangle (see figure 2.1(c)). The bary-centric coordinates (see, e.g., [Coxeter, 1969]) of this point in the triangle directly lead to a definition of three confidences that add up to one:

$$\begin{aligned} c_{i0D} &= 1 - x, \\ c_{i1D} &= x - y, \\ c_{i2D} &= y. \end{aligned} \tag{2.1}$$

These three confidences reflect the volume of the areas of the three sub-triangles which are defined by the point in the triangle and the corners of the triangle (see figure 2.1(c)). For example, for an arbitrary point P in the triangle, the area of the sub-triangle i0D-P-i1D denotes the confidence for i2D as shown in figure 2.1(c). That leads to the decision areas for i0D, i1D and i2D as seen in figure 2.1(d). See appendix A and [Felsberg et al., 2007a] for more details.

For the example image in figure 2.1, computed *iD* is given in figure 2.2.

Figure 2.3 shows how a set of example local image structures map on to the *iD*triangle. The figure shows that different visual structures map to different areas in the triangle. A detailed analysis of how 2D structures are distributed over the intrinsic dimensionality triangle and how some visual information depends on this distribution can be found in chapters 3 and 5 and references [Kalkan et al., 2005, Kalkan et al., 2006].

This thesis proposes intrinsic dimensionality as a new tool for analyzing the inherent properties of different image structures using the intrinsic dimensionality triangle. In chapter 3, this is performed for

the analysis of the distribution of local image structures and the quality of different optic flow algorithms. Chapter 5 uses the *iD* triangle for the analysis of the relation between local 2D and 3D structures.

2.2 Multi-modal Visual Features – Primitives

This thesis extensively utilizes *primitives* which are local, multi-modal visual feature descriptors that were introduced in [Krüger et al., 2004b]. They are semantically and geometrically meaningful descriptions of local image patches, motivated by the hyper-columnar structures in V1 ([Hubel and Wiesel, 1969]).

Primitives can be edge-like and homogeneous and either 2D or 3D. For edge-like primitives, the corresponding 3D primitive is extracted using stereo. As for homogeneous primitives, the 3D primitive is estimated from the 3D edge-like primitives, which is the topic of chapter 7.

An edge-like 2D primitive is defined as:

$$\pi^e = (\boldsymbol{x}, \theta, \omega, (\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r), f), \tag{2.2}$$

where $\boldsymbol{x}$ is the image position of the primitive; θ is the 2D orientation; ω represents the contrast transition; $(\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r)$ is the representation of the color, corresponding to the left ($\mathbf{c}_l$), the middle ($\mathbf{c}_m$) and the right side ($\mathbf{c}_r$) of the primitive; and, f is the optical flow extracted using Nagel-Enkelmann optic flow algorithm [Nagel and Enkelmann, 1986].

As the underlying structure of a homogeneous image patch is different from that of an edge-like patch, a different representation is needed for homogeneous 2D primitives (called *mono*s in this thesis):

$$\pi^m = (\boldsymbol{x}, \mathbf{c}), \tag{2.3}$$

where $\boldsymbol{x}$ is the image position, and $\mathbf{c}$ is the color of the mono[1].

See [Krüger et al., 2007] for more information about these modalities and their extraction. Figure 2.4 shows extracted primitives for an example scene.

π^e is a 2D feature which can be used to find correspondences in a stereo framework to create 3D primitives (as introduced in [Krüger and Felsberg, 2004, Pugeault et al., 2006]) which have the following

[1] For analyzing shape from shading, representation of local intensity variance can be included in a further study.

formulation:

$$\Pi^e = (\boldsymbol{X}, \Theta, \Omega, (\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r)), \tag{2.4}$$

where $\boldsymbol{X}$ is the 3D position; Θ is the 3D orientation; Ω is the phase (i.e., contrast transition); and, $(\mathbf{c}_l, \mathbf{c}_m, \mathbf{c}_r)$ is the representation of the color, corresponding to the left ($\mathbf{c}_l$), the middle ($\mathbf{c}_m$) and the right side ($\mathbf{c}_r$) of the 3D primitive.

In chapter 7, we estimate the 3D representation Π^m of monos which stereo fails to compute due to the correspondence problem:

$$\Pi^m = (\boldsymbol{X}, \mathbf{n}, \mathbf{c}), \tag{2.5}$$

where $\boldsymbol{X}$ and $\mathbf{c}$ are as in equation 2.3, and $\mathbf{n}$ is the orientation (*i.e.*, normal) of the plane that locally represents the mono.

2.3 Acknowledgements

Section 2.1 is a product of collaboration with Michael Felsberg and is published in a co-authored journal [Felsberg et al., 2007b].

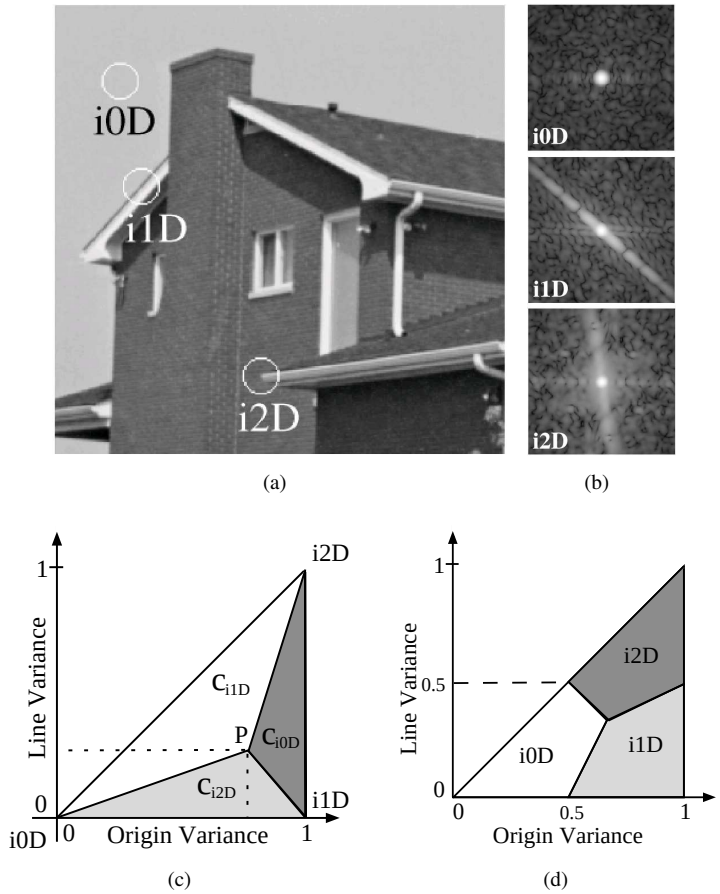


Figure 2.1: Illustration of *iD* (Sub-figures (a,b) are taken from [Felsberg and Krüger, 2003]). (a) Three image patches for three different intrinsic dimensions. (b) The 2D spatial frequency spectra of the local patches in (a), from top to bottom: i0D, i1D, i2D. (c) The topology of *iD*. Origin variance is variance from a point, i.e., the origin. Line variance is variance from a line, measuring the junctionness of the signal. c_{iND} for $N = 0, 1, 2$ stands for confidence for being i0D, i1D and i2D, respectively. Confidences for an arbitrary point P is shown in the figure which reflect the areas of the sub-triangles defined by P and the corners of the triangle. (d) The decision areas for local image structures.

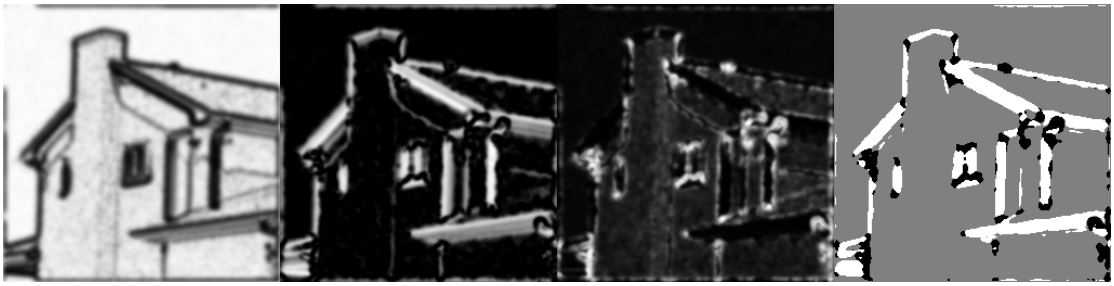


Figure 2.2: Computed *iD* for the image in figure 2.1, black means zero and white means one. From left to right: $c_{\text{i0D}}, c_{\text{i1D}}, c_{\text{i2D}}$ and highest confidence marked in gray, white and black for i0D, i1D and i2D, respectively.

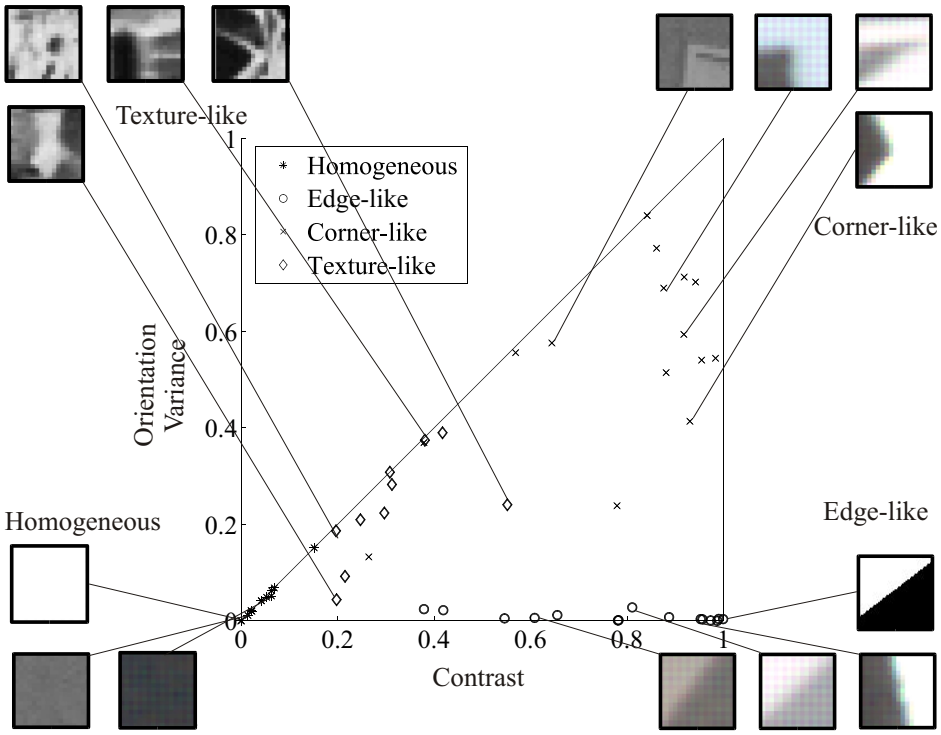


Figure 2.3: How a set of 54 patches map to the different areas of the intrinsic dimensionality triangle. Some examples from these patches are also shown. The horizontal and vertical axes of the triangle denote the contrast and the orientation variances of the image patches, respectively.

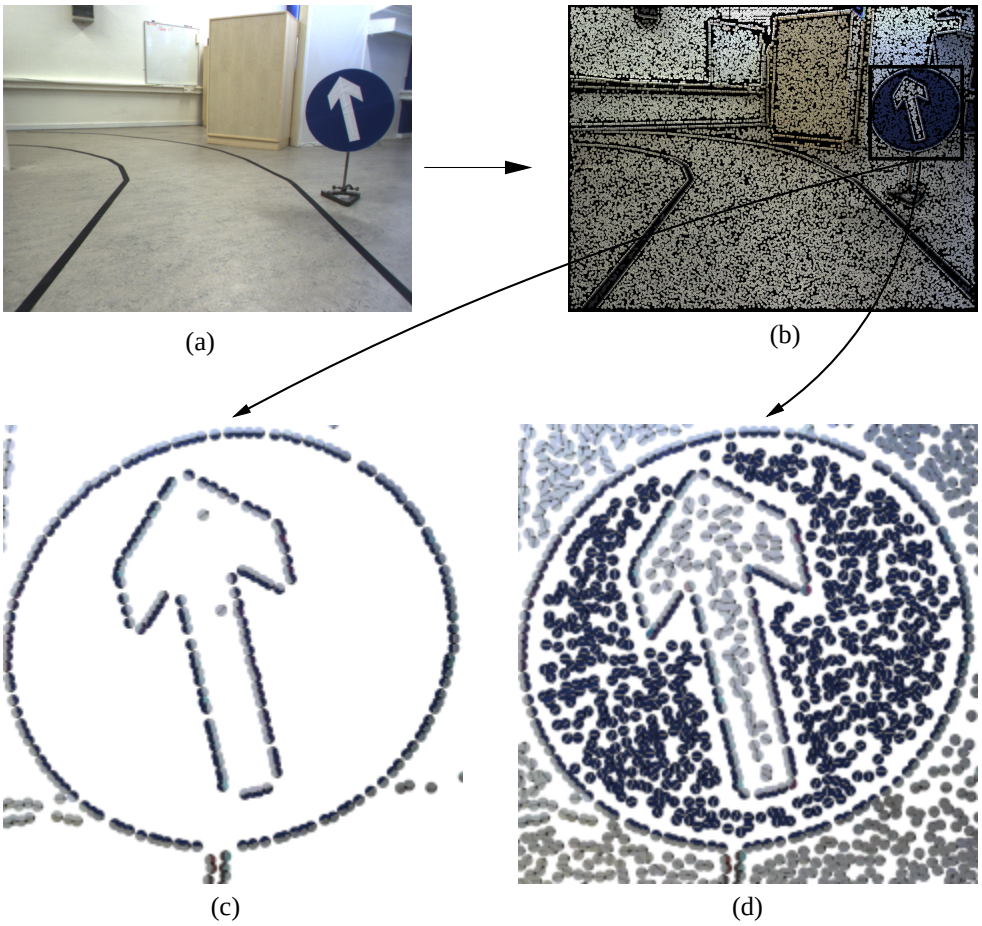


Figure 2.4: Extracted primitives (b) for the example image in (a). Magnified edge primitives and edge primitives together with monos are shown in (c) and (d) respectively.

Chapter 3

Local Image Structures and Optic Flow Estimation

As mentioned in 1.2, optic flow information in early vision is ambiguous. This ambiguity in optic flow can be disambiguated by using the flow information available at the junction-like structures in early cognitive vision. Such a disambiguation has been modeled as a feedback mechanism in [Bayerl and Neumann, 2007]. This chapter investigates the extent of the ambiguity in optic flow estimation and analyzes it for different local image structures. Namely, the continuous definition of intrinsic dimensionality introduced in section 2.1 is used to investigate (1) the quality of different optic flow estimation algorithms depending on the underlying local image structure and (2) the distribution of signals in natural images according to their intrinsic dimensionality. Namely, it suggests that the quality of optic flow estimation and the underlying local image structure are strongly linked.

Regarding the distribution of signals, the chapter shows that:

D0. i0D signals split into two clusters; one peak corresponding to over-illuminated (white) or under-illuminated (black) patches and a Gaussian-shaped cluster corresponding to image noise at homogeneous but not under- or over-illuminated image patches (see figure 3.1(a)).

D1. For i1D signals, there exists a concentration of signals in a stripe-shaped cluster corresponding to high origin variance (high amplitude) *and* low line variance (see figure 3.1(a)). This also reflects the

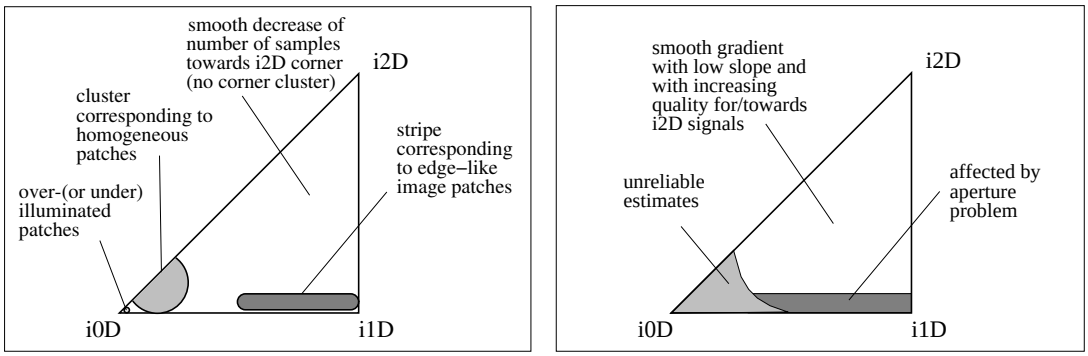


Figure 3.1: **a)** Schematic representation of the distribution of local image patches in natural images according to their intrinsic dimension. **b)** Schematic representation of the quality of optic flow estimation according to the intrinsic dimension of the underlying signal.

importance of an orientation criterion that is based on local amplitude and orientation information (see, e.g., [Princen et al., 1990]).

D2. In contrast to the i0D and i1D cases, there exists no cluster for i2D signals but there is a smoothly decreasing surface towards the i2D corner. This continuity in the distribution for the i2D case indicates that it is rather difficult to formulate a purely local criterion to detect corners in natural images.

Optic flow in early vision is ambiguous because local estimation of optic flow faces the well-known *aperture problem*: Through an aperture, the true flow is observable only for two dimensional structures, *i.e.*, corner-like structures, end of edges and some kinds of textures. As for edge-like structures, only the flow which is along the intensity gradient can be computed.

The property of optic flow estimation at homogeneous image patches, edges and corners has been discussed extensively (see, e.g., [Barron et al., 1994, Zetzsche et al., 1991, Mota and Barth, 2000]). It has been argued that many different motion detectors specialised to particular image structures exist in human vision (for a discussion, see [Cavanagh and Mather, 1989, Johnston and Clifford, 1995]). In general, it is acknowledged that;

A0. Optic flow estimates at homogeneous image patches tend to be unreliable as the lack of structure makes it impossible to find correspondences in consecutive frames.

A1. Optic flow at edge-like structures faces the aperture problem; i.e., using local information, only the normal flow for these structures can be computed.

A2. Only for i2D structures, optic flow estimation can lead to true optic flow estimates by local methods. However, many i2D structures result from depth discontinuities where optic flow estimation algorithms fail to estimate the true motion. In order to get the true motion field, flow algorithms need to deal with at least two different motions in the local area [Bayerl and Neumann, 2007].

This chapter investigates these claims more closely for several optic flow algorithms (namely, Nagel-Enkelmann, [Nagel and Enkelmann, 1986], Lucas-Kanade [Lucas and Kanade, 1981] and a phase-based approach from [Gautama and Hulle, 2002]). It will be shown that the continuous formulation of intrinsic dimensionality allows for a better quantitative investigation and characterization of the quality of optic flow estimation (and hence, the optic flow properties as stated in A0-A2) depending on the local signal structure. Namely:

- The algorithms that have been tested in this chapter all had problems with local image structures that were very close to the i0D corner of the iD-triangle (see figure 3.1(b)).
- The performance for image structures in the stripe shaped cluster corresponding to edge-like structures was effected by the aperture problem (see figure 3.1(b)). However, the results depend both quantitatively and qualitatively on the different algorithms and even on different parameters when the same algorithm was used.
- The improvement of performance for signals in the i2D area of the iD triangle was visible but small. Average performance increases smoothly and slightly towards the i2D corner (see figure 3.1(b)).

These results support the above-mentioned statements (A0)-(A2) about optic flow estimation. However, by making use of a continuous understanding of intrinsic dimensionality, these statements have been made quantitatively more specific in terms of (1) characterization of sub-areas for which they hold and (2) their strength. The analysis in this chapter suggests a relationship between the distribution of the signals in the continuous intrinsic dimensionality space and properties of optic flow estimation. In this way, a new tool for better analysis of optic flow algorithms is introduced.

There has been other works analyzing errors in optic flow estimates [Fermueller et al., 2001, Simoncelli et al., 1991, Nagel and Haag, 1998]. In [Simoncelli et al., 1991], using a probabilistic framework for estimating optical flow, it is proven that uncertainty is involved in this estimation process due to several causes such as image noise and inherent limits of motion estimation. In [Nagel and Haag, 1998], it is shown that gradient-based motion estimation methods underestimate the true flow. In [Fermueller et al., 2001], too, it is analytically shown that certain kinds of bias in different classes of optic flow algorithms caused by noise in the image data usually lead not only to underestimate of the magnitude of optical flow and but also to consistent bias in the estimation of the direction. In contrast to the investigations in [Fermueller et al., 2001, Simoncelli et al., 1991, Nagel and Haag, 1998], this chapter is interested in the quality of flow estimates depending on the local image structure. This is achieved not by analytic means but by statistical comparisons using ground truth data.

3.1 Distribution of Local Image Structures

The distribution of local image structures is analyzed using a set of 7 natural sequences with 10 images each (see figure 3.5). The images have a resolution of 1276×1016. For the analysis, the origin and the line variance are computed for each pixel (for details see section 2.1). This corresponds to one point in the iD triangle (figure 2.1(c)). The distribution of the frequency of these points in the triangular structure is shown in figure 3.2(a). Since there exist large differences in the histogram, only the logarithm is shown.

The distribution shows two main clusters. The peak close to the origin corresponds to low origin variance. It is visible that most of the signals that have low origin variance have high line variance. These correspond to nearly homogeneous image patches. Since the orientation is almost random for such homogeneous image patches, it causes high line variance. There is also a small peak at position $(0, 0)$ that corresponds to saturated/black image patches. The other cluster is for high origin variance signals with low line variance, corresponding to edge-like structures. The form of this cluster is a small horizontal stripe rather than a peak. Finally, there is a smooth decrease while approaching to the i2D area of the triangle. That means that there does not exist a cluster for corner-like structures like the ones for homogeneous image patches or edges. Along the origin variance axis, a small continuous gap is observed. This gap suggests that there are no signals with zero line variance. This is due to the fact that at positions with positive origin–variance (i.e., positive magnitude), there is always noise included which

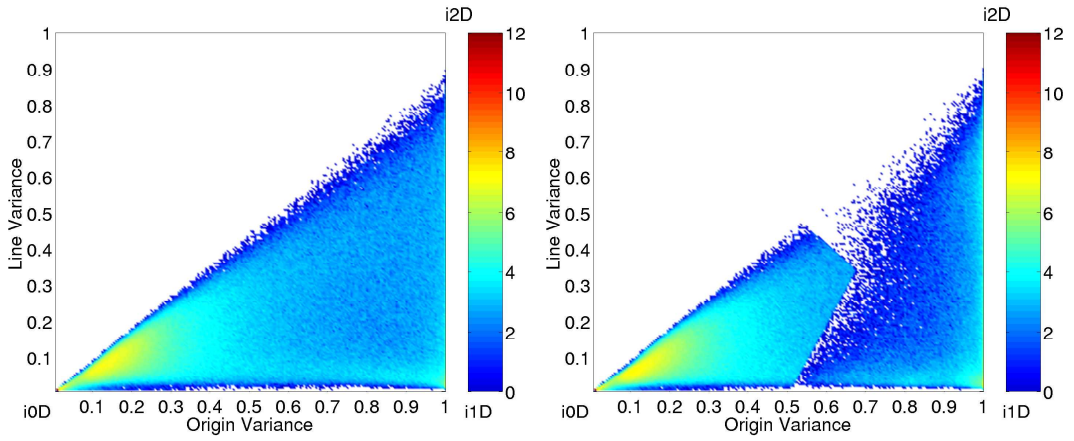


Figure 3.2: Logarithmic plot of the distribution of intrinsic dimensionality. **(a)** The distribution for regularly sampled points. **(b)** The distribution when the positions are modified according to iD (See the text for details of this modification).

causes some line variance.

Also seen from the figure is that there are far more i0D signals than i1D or i2D signals. Besides, it is clear that there are more i1D structures than i2D structures in natural images. The percentages of i0D, i1D and i2D structures turned out to be 86%, 11% and 3%, respectively, in the natural images that have been used.

Figure 3.3: Illustration of positioning for an edge. (a) Without positioning. (b) With positioning as explained in the text.

Edges or corners are structures that are bound to a specific position. For example, the position of an edge is supposed to be placed directly on the image discontinuity; or, for a corner, in which a certain number of lines intersect, the corner should be placed directly on the intersection. This positioning can be achieved by making use of the local amplitude information in the image depending on the intrinsic dimensionality which is described in detail in [Krüger et al., 2004a] (see figure 3.3). Note also that

features such as orientation and optic flow depend on this positioning. When the positions of edges and corner-like structures accordingly are determined accordingly, the distribution of local image structures becomes as shown in figure 3.2(b). It is qualitatively similar to the distribution achieved with regular sampling. However, since the position is determined depending on the local amplitude (and in this way by maximizing origin variance; see [Krüger and Felsberg, 2003]), there is a shift towards positions with higher amplitude that constitute the gaps at the border between i0D, i1D and i2D signals and the stripe along the i1D-i2D border of the triangle. In the later stages of the analysis in this chapter, this positioning is adopted.

Zetzsche and his colleagues also investigated the distribution of local image structures in [Wegmann and Zetzsche, 1990]. They analyzed the multi-dimensional hyperspace which was constructed from all possible combinations of orientation filter outputs. The hyperspace consisted of m axes corresponding to m different orientations such that the origin denoted the homogeneous signals; the axes and the planes between the neighboring axes denoted the i1D structures; and, the planes between the non-neighboring axes denoted i2D signals. Zetzsche and his colleagues could drive proportions of the different local structures (which basically reflect the percentages provided above) and visualize clusters of the structures for a few orientation pairs. Due to the complexity of the hyperspace, however, the visualization becomes more complex than the triangular representation of iD.

3.2 Distribution of Orientation of Local Image Structures

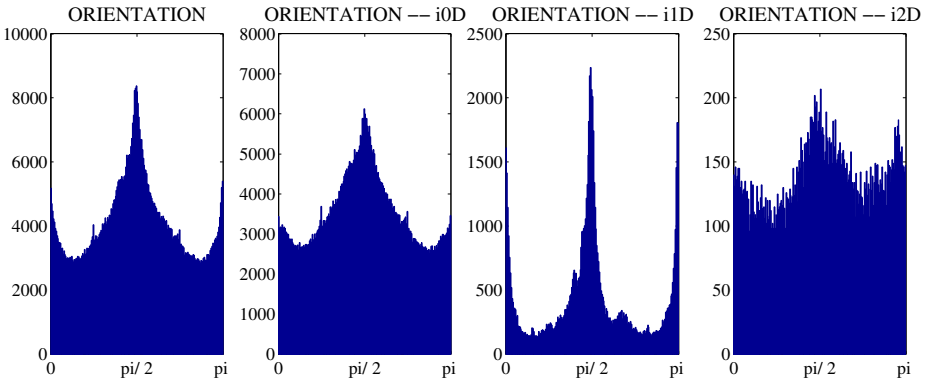


Figure 3.4: Orientation distribution depending on iD. The first image shows the total distribution. The sequences that have been used for this analysis are introduced in section 3.1.

It is known (see, e.g., [Krueger, 1998, Coppola et al., 1998]) that the distribution of orientations of i1D signals shows strong peaks at horizontal and vertical structures (i.e., for the values 0, $\pi/2$ and π). However, neither for a completely homogeneous image patch nor for a corner the concept of orientation (although computable) makes sense and the computed orientation is random. Therefore, it is expected that the distribution of orientations of i0D and i2D signals should be homogeneous.

The distribution of the orientation of signals and the quantitative differences depending on the intrinsic dimensionality of the patches is displayed in figure 3.4. The figure shows that there are significant peaks for the i0D and i2D signals although they are smaller than the peaks in the distribution of i1D signals. This suggests that orientation is a meaningful concept for some non-i1D signals, too. This also stresses the advantages of a continuous understanding of intrinsic dimensionality.

3.3 Optic Flow Estimation Algorithms

This section briefly describes the optic flow algorithms that have been used in this chapter.

3.3.1 The Lucas-Kanade Algorithm

The Lucas-Kanade algorithm works by minimizing the following functional [Lucas and Kanade, 1981] over a spatial neighborhood Ω:

$$\iint_{\Omega} W^2(x,y) \left[\nabla I(x,y,t) \cdot \mathbf{v} + I_t(x,y,t)\right]^2 \mathrm{d}x\, \mathrm{d}y, \tag{3.1}$$

where $W(x,y)$ is the window function over Ω that gives more influence to constraints at the center of the neighborhood; $\nabla I(x,y,t)$ denotes the intensity gradient at time t at spatial location (x,y); $\mathbf{v}$ is the velocity field to be found; and, I_t denotes the derivative of I with respect to t. Basically, the Lucas-Kanade algorithm makes use of the well-known gradient constraint equation $\nabla I^T \cdot \mathbf{v} + I_t = 0$ where weighting is performed over a local neighbourhood.

The Lucas-Kanade is an optic flow algorithm which uses first order derivatives. Due to its smaller complexity when compared with others, it is known to be a fast algorithm.

3.3.2 The Nagel–Enkelmann Algorithm

The Nagel–Enkelmann algorithm [Nagel and Enkelmann, 1986] also makes use of the gradient constraint equation but applies a second order derivative constraint in addition. The following functional is minimized:

$$\iint (\nabla I^T \mathbf{v} + I_t)^2 + \frac{\alpha^2}{\| \nabla I \|_2^2 + 2\delta} \left[(u_x I_y - u_y I_x)^2 + (v_x I_y - v_y I_x)^2 + \delta(u_x^2 + u_y^2 + v_x^2 + v_y^2) \right] \mathrm{d}x \, \mathrm{d}y, \quad (3.2)$$

where α and δ are constants; u and v are respectively the horizontal and the vertical components of the velocity vector $\mathbf{v}$; and, for a function F, F_z denotes the partial derivative of F with respect to variable z. Main terms of the formula are $(u_x I_y - u_y I_x)^2 + (v_x I_y - v_y I_x)^2$ and $(u_x^2 + u_y^2 + v_x^2 + v_y^2)$. The first term smoothes velocity an-isotropically, i.e., orthogonal to the intensity gradient. The second isotropic term states that velocity should be constant over position[1].

Since the Nagel–Enkelmann algorithm can be interpreted as a diffusion process (see [Alvarez et al., 2000]) with fixed number of iterations, an increase in the number of iterations means an increase in the region of influence used in the computation, and hence, using more global information. The Nagel-Enkelmann algorithm encourages slow variations in the gradient of the vector field by the smoothing term in equation 3.2. This leads with increasing number of iterations (i.e., increasing diffusion) naturally to a more regular distribution of directions (as visible in the first two rows of figure 3.6). In this chapter, the effect of using more global information on the accuracy of the flow estimation is also provided.s

3.3.3 The Phase-Based Approach

Phase-based optic flow algorithms make use of the phase gradient for finding the flow. It has been shown that temporal evolution of contours of constant phase provides a better approximation to local flow (see e.g., [Fleet and Jepson, 1990]). The basic assumption is that phase contours should be constant over time [Fleet and Jepson, 1990, Gautama and Hulle, 2002]. This assumption can be formulated as $\phi(x, y, t) = c$, where $\phi(x, y, t)$ denotes the phase component at spatial location (x, y) at time t. Taking differentiation

[1] In our simulations, the standard values 0.5 and 1.0 for α and δ, respectively, are used as suggested and usually practiced in the literature (see, e.g. [Barron et al., 1994]).

with respect to time, the following constraint is found:

$$\nabla\phi(x, y, t) \cdot (\nabla(x, y), 1) = \nabla\phi(x, y, t) \cdot (\mathbf{v}, 1) = 0. \tag{3.3}$$

Among phase-based approaches, this chapter uses a recent implementation [Gautama and Hulle, 2002] in which the constraint (3.3) is solved for a number of Gabor filters and the flow orthogonal to the orientation of each filter is found. Combining the solutions reached by the filters yields the true displacement. This chapter will show that in this way, even for a large number of i0D signals good optic flow can be estimated.

Figure 3.5: Some of the image sequences used in our analysis. The first 3 images are from one of the sequences (the starting image, the middle image and the last image). Remaining figures are the images from other sequences.

3.4 Optic Flow Estimation

This section analyzes the distribution of optic flow direction (subsection 3.4.1) and the error of optic flow estimation and its relation to the iD triangle (subsection 3.4.2).

3.4.1 Optic Flow Direction

The distribution of the flow direction of the optic flow vectors (using the Nagel–Enkelmann algorithm with 10 and 100 iterations, and the phase-based approach) is shown in figure 3.6.

The distribution of the direction varies significantly with the intrinsic dimensionality. The statistics of the true flow can be expected to show some homogeneity since a translational forward motion is dominant in the sequences that leads to a regular flow field (see, e.g., [Lappe et al., 1999]). A detailed discussion of first order statistics of optic flow in natural scenes can be found in [Calow et al., 2004]. They showed that the main factor for irregularity is that the large amount of structure near in the lower visual field as compared to the lack of structure in the upper visual field causes larger flow in the lower visual field. This, however, does not effect the magnitude but only the orientation. However, for the Nagel–Enkelmann algorithm with 10 iterations (figure 3.6, top row), the distribution of the direction of optic flow vectors of i1D signals directly reflects the distribution of orientation of i1D signals. Since only the normal flow can be computed for ideal i1D signals (using local information only), the dominance of vertical and horizontal orientations (see section 3.2) leads to peaks at horizontal and vertical flows. The fact that basically there exits a direct quantitative equivalence of the distribution of i1D orientations and the distribution of optic flow directions reflects the seriousness of the aperture problem. In contrast, the distribution of direction of optic flow vectors of i0D and i2D signals is much more homogeneous. When the number of iterations is increased (and hence, more global information is used in the computation of the flow as explained in section 3.3), the peaks that correspond to horizontal and vertical lines become smaller (figure 3.6, middle row). For the phase-based approach and Lucas-Kanade, a different picture occurs (figure 3.6, last two rows): the peaks are less apparent.

As a summary, figure 3.6 suggests that there is a relation between the direction of estimated optic flows and the orientation distribution of signals. However, the strength of this relation depends on the particular algorithm and its parameters. For example, when the used information is very local, the Nagel–Enkelmann algorithm computes basically the normal flow which results in a strong relation between the distribution of optic flow direction and distribution of orientations in the images. However, when the number of iterations is increased, this relation becomes weaker because of the decrease of the aperture effect due to using more global information.

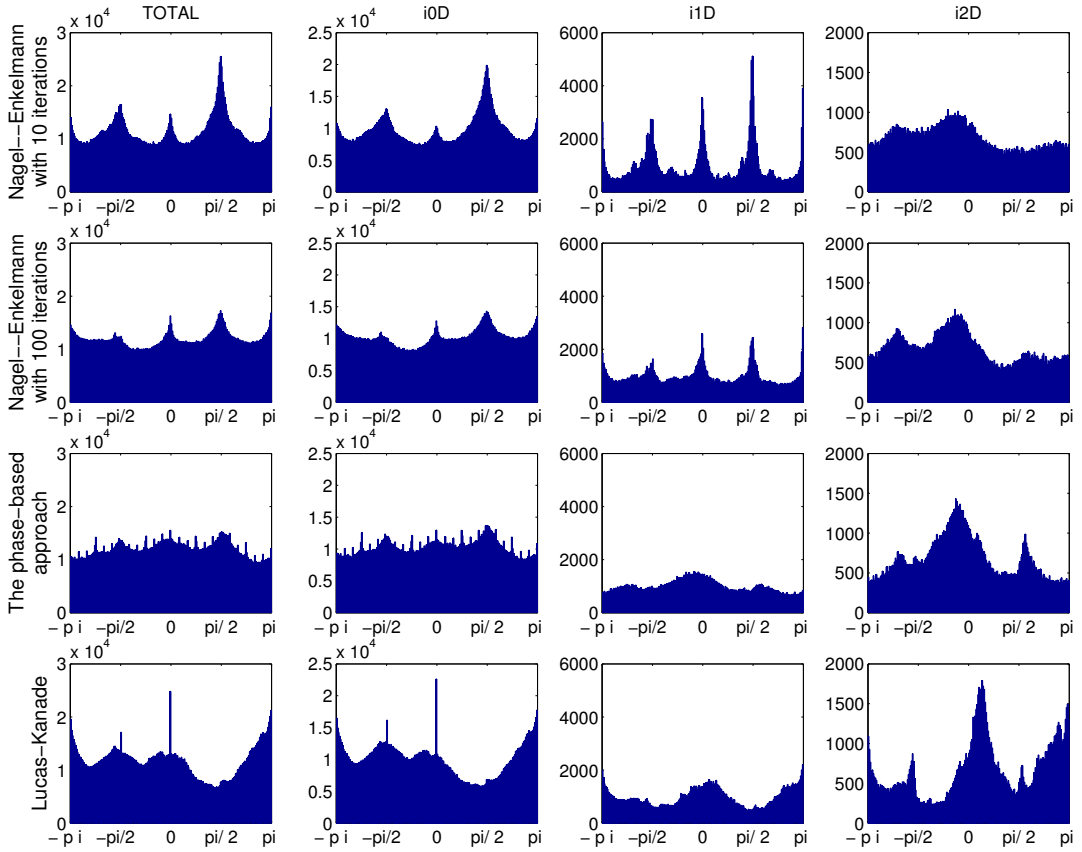


Figure 3.6: Distribution of direction of optic flow vectors depending on the intrinsic dimension. The histograms show the summed up distributions over the sequences which are introduced in section 3.1. From top to bottom: The Nagel–Enkelmann algorithm with 10 iterations; The Nagel–Enkelmann algorithm with 100 iterations; The phase-based approach; The Lucas-Kanade algorithm. From left to right: The total distribution; The distribution for i0D signals; The distribution for i1D signals; The distribution for i2D signals.

3.4.2 Analysis of Quality of Optic Flow Estimation

This subsection analyzes the qualities of the optic flow estimation depending on the intrinsic dimension. For this, the computed flow needs to be compared with a ground truth. For this, the Brown Range Image Database (BRID), a database of 197 range images collected by Ann Lee, Jinggang Huang and David Mumford at Brown University (see also [Huang et al., 2000]) is used. The range images are recorded with a laser range-finder[2]. The data of each point consist of 4 values: the distance, the horizontal angle and the vertical angle in spherical coordinates and a value for the reflected intensity of the laser beam (see figure 3.7). The knowledge about the 3D data structure allows for a simulation of a moving camera in a scene and is used to estimate the correct flow for nearly all pixel positions of a frame of an image sequence. It should be noted that this approach cannot produce correct flow for occluded areas.

The simulated motion is forward translation. Different motion types (such as rotation, and rotation and translation) may produce different global motion types. Therefore, the results in this chapter are valid only for translational motions, and other types of motions should be expected to yield quantitatively if not qualitatively different results.

Different flow estimation algorithms yield flow fields with different densities; i.e., they can make an estimation of the motion for a certain proportion of the image data. By adjusting the parameters of the flow algorithms that have been used in this chapter, the flow fields were made as dense as possible for our analysis, which happened to be on the average 100%, 90% and 86% respectively for the Nagel-Enkelmann, the Lucas-Kanade algorithms and the phase-based approach.

The quality of optic flow estimation is displayed in a histogram over the iD triangle (see figures 3.8 and 3.9). The error is calculated using the well known measure:

$$e(\mathbf{u},\mathbf{v}) = acos\Big(\frac{\mathbf{u}\cdot\mathbf{v}+1}{(\mathbf{u}\cdot\mathbf{u}+1)(\mathbf{v}\cdot\mathbf{v}+1)}\Big), \tag{3.4}$$

where $\mathbf{u}$ and $\mathbf{v}$ are the flow vectors (see also [Barron et al., 1994])[3] This measure is called the *combined error* in this chapter.

[2]Each image contains 44×1440 measurements with an angular separation of 0.18 degree. The field of view is 80 degree vertically and 259 degree horizontally. The distance of each point is calculated from the time of flight of the laser beam, where the operational range of the sensor is $2 - 200$m. The laser wavelength of the laser beam is $0.9\mu m$ in the near infrared region.

[3]Measurements using angular and magnitudal errors with the formulas $e_{ang}(\mathbf{u},\mathbf{v}) = acos(\frac{\mathbf{u}\cdot\mathbf{v}}{|\mathbf{u}|.|\mathbf{v}|})$, $e_{mag}(\mathbf{u},\mathbf{v}) = \frac{abs(|\mathbf{u}|-|\mathbf{v}|)}{|\mathbf{u}|+|\mathbf{v}|}$ yield similar results (for details, see [Kalkan et al., 2004a]).

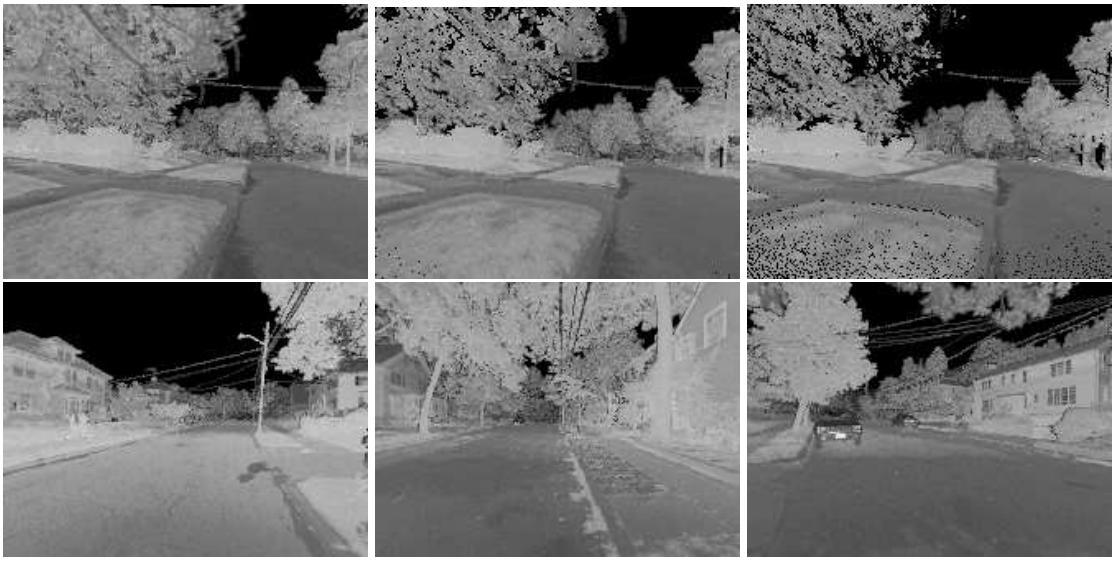
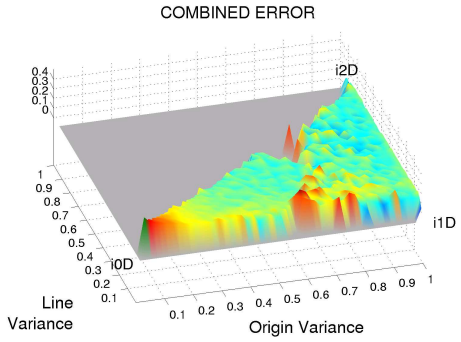


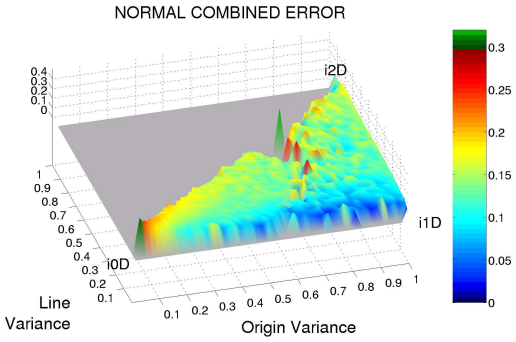
Figure 3.7: Sequences with ground truth optic flow. Reflected intensity of the laser beam is shown. The first line is from the same sequence (the starting image, the middle image and the last image). Other images are taken from the other sequences.

For the Nagel–Enkelmann algorithm with 10 iterations, the combined error computed using the original ground truth (see figure 3.8(a)) is high for signals close to the i0D corner of the triangle as well as on the horizontal stripe from the i0D to the i1D corner. In the other parts, there is a smooth surface which shows that the error decreases towards the i2D corner (note that the peaks in the middle of the triangle are due to only few samples and can be ignored). This is in accordance with the notion that optic flow estimation at corner-like structures is more reliable than for edges and homogeneous image patches (A2). However, in figure 3.8(a), it becomes obvious that the area where more reliable flow vectors can be computed is very broad and covers also i0D and i1D signals. Furthermore, the decrease of error is rather slight which points to the fact that the quality of flow computation is limited in these areas, as well. When the number of iterations is increased (figure 3.8(c)), the estimation of flow becomes better. In fact, it is observable that the area where optic flow is estimated with low error covers almost the whole triangle except some parts in the i1D area of the triangle.

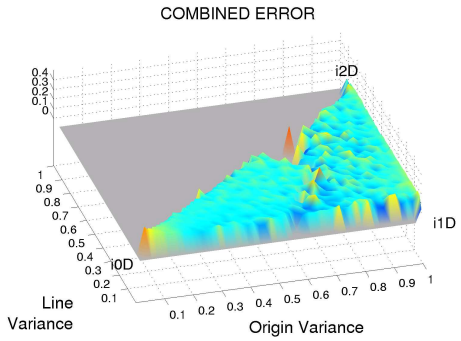
Among all, the phase-based approach produces the best results for quite a large area (figure 3.9(c)). However, for many i1D signals the estimate is more unreliable than for most i0D and i2D cases. For the Lucas-Kanade algorithm, it is observed also that the area with low error smoothly extends to some i0D and i1D signals (figure 3.9(a)). The figures 3.8(a),3.8(c), 3.9(a) and 3.9(c) suggest that this only slight



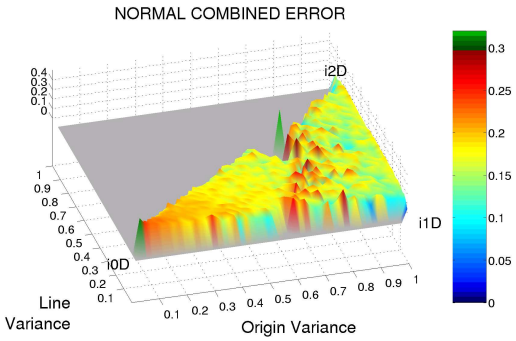
(a) Nagel–Enkelmann with 10 iterations.



(b) Nagel–Enkelmann with 10 iterations with the projected ground truth.



(c) Nagel–Enkelmann with 100 iterations.



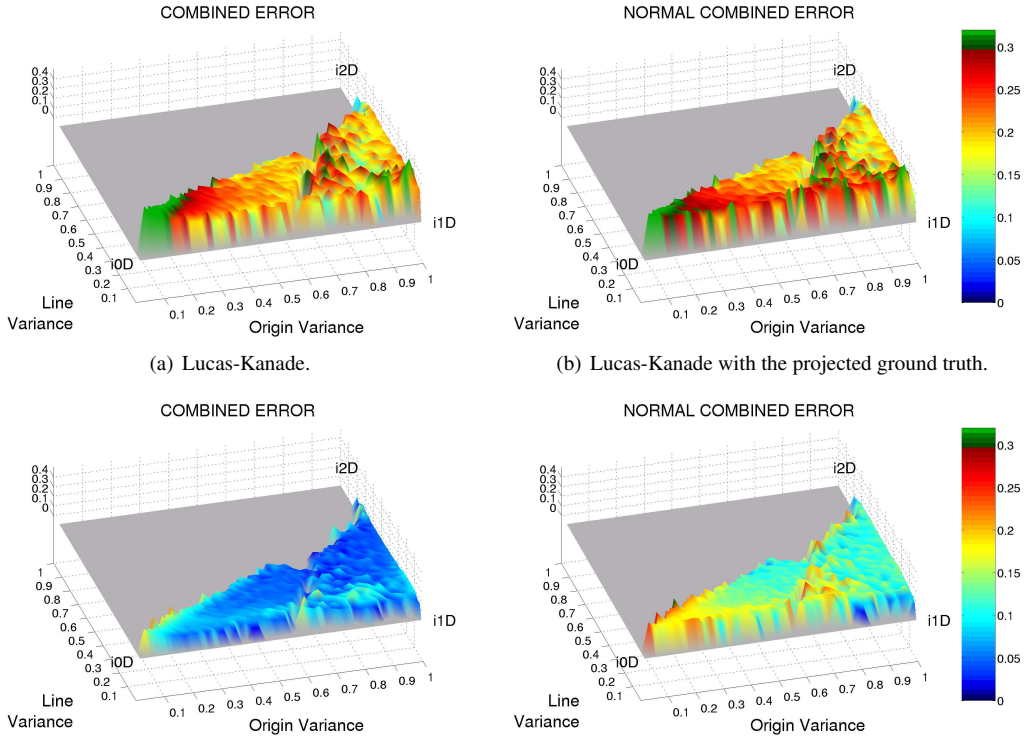
(d) Nagel–Enkelmann with 100 iterations with the projected ground truth.

Figure 3.8: Qualities of the optic flow algorithms depending on iD. Left column shows the errors between computed flow and the ground truth. Right column shows the errors between computed flow and the ground truth projected orthogonal to the orientation of signals. Color bars show the error values for corresponding colors of corresponding graphs. **(a,b)** Nagel–Enkelmann with 10 iterations. **(c,d)** Nagel–Enkelmann with 100 iterations.

decrease is a general property of the investigated optic flow algorithms.

For better analysis of the aperture problem, the computed flow is also compared against the projection of the ground truth over the normal vectors (i.e., the true normal flow). For this, the normal vector of the image patch is computed using the local orientation; then, the ground truth is projected over this vector, and the error is computed between the optic flow vector and this projected ground truth. In figures 3.9 and 3.8, this is called as the *normal combined error*.

For the error computed using the normal ground truth (see figures 3.8(b), 3.8(d)), a different picture occurs. For the Nagel–Enkelmann algorithm with 10 iterations (figure 3.8(b)), the error is very low for a horizontal stripe from the middle point between the i0D and i1D corners to the i1D corner. When compared to figure 3.8(a), this figure reflects the effect of the aperture problem when only local information



(a) Lucas-Kanade.

(b) Lucas-Kanade with the projected ground truth.

(c) The phase-based approach.

(d) The phase-based approach with the projected ground truth.

Figure 3.9: Qualities of the optic flow algorithms depending on iD. Left column shows the errors between computed flow and the ground truth. Right column shows the errors between computed flow and the ground truth projected orthogonal to the orientation of signals. Color bars show the error values for corresponding colors of corresponding graphs. **(a,b)** The Lucas-Kanade. **(c,d)** The phase-based approach.

is used. When the number of iterations is increased, it is observable that using more global information decreases the effect of the aperture problem (figures 3.8(c),3.8(d)). However, it is visible that the quality of the estimated error area in the i1D area of the triangle is always significantly lower than the quality of the estimated normal flow with a small number of iterations (i.e., using only very local information).

Comparing the results for the Nagel–Enkelmann algorithm with 10 and 100 iterations suggests that increasing the region of influence means an increase in the overall quality of optic flow estimation. However, the error in estimation of flow in the i1D area of the triangle using more global information (figure 3.8(c)) is always significantly higher than the error of the estimated normal flow using more local information (see, figure 3.8(b)), which can be computed with high reliability. The information for such signals is of great importance since (1) there exists a large number of local image patches corresponding to such

edge–like structures (see, figure 3.2) and (2) constraints for global motion estimation can be defined on line–correspondences (see, e.g., [Rosenhahn, 2003, Krüger and Wörgötter, 2004]), i.e., correspondences that only require normal flow. For these tasks, a reliably estimated normal flow might be a better basis than an unreliably estimated true flow.

3.5 Discussion

This chapter analyzed the distribution of local image patches and the quality of optic flow estimates using intrinsic dimensionality.

A continuous understanding of intrinsic dimensionality (see section 2.1) allows for a more precise characterization of established structures in terms of their statistical manifestation in natural images (see figures 3.1(a) and 3.2). Moreover, such a continuous formulation can be used for a more quantitative investigation and characterization of the quality of optic flow estimation depending on local image structures. The current chapter could justify and more precisely quantify generally acknowledged ideas about such estimates (see A0-A2 in the introduction). More specifically, it showed that (see also figure 3.1(b), figure 3.8, and 3.9):

Q0. In general, homogeneous structures lead to low quality optic flow estimation. However, for many i0D signals, true flow can be estimated with good accuracy. In fact, the algorithms compute the flow with quite a low error for most of the i0D structures.

Q1.1. There exist significantly more horizontal and vertical structures in natural images (see also, e.g. [Krueger, 1998, Coppola et al., 1998]). The strength of the dominance of these structures depends crucially on the intrinsic dimension. Furthermore, the distribution of orientations is directly reflected in the distribution of the estimates of optic-flow directions as an effect of the aperture problem. However, the degree of this reflection is observed to be very much dependent on the particular optic flow algorithm and the parameters used.

Q1.2. The optic flow estimates in the stripe-shaped cluster with high origin–variance and low line–variance (corresponding to edges) are in general worse than for i2D signals for all of the investigated algorithms. However, for the Nagel–Enkelmann algorithm, the normal flow can be estimated

reliably for this stripe-shaped cluster in the i1D signal domain. It is important to note that this reliably computed normal flow is an important information as such. For example, line–line correspondences that can be derived from the normal flow play an important role in Rigid Body Motion estimation (see, e.g., [Shevlin, 1998, Rosenhahn and Sommer, 2002, Krüger and Wörgötter, 2004]). Using the Nagel–Enkelmann algorithm, the effect of using more global information on the quality of optic flow estimation for i1D signals is also provided.

Q2. The quality of optic flow estimation is higher for i2D signals. However, in analogy to the lack of a cluster for i2D signals, there exists a continuous signal domain (covering also sub-areas of i0D and i1D signals) for which a higher quality in the optic flow estimation can be achieved. The increase of the quality, on the other hand, is only slight which suggests that the role of i2D structures for motion estimation might not be as important as suggested in the literature (see, e.g., [Mota and Barth, 2000]). Observing this behaviour for different optic flow algorithms suggests that this is a general property of optic flow algorithms.

It should be stressed that the aim of this chapter was not to find the 'best' algorithm but to make general properties of optic flow algorithms explicit. The choice of the 'best' algorithm depends a lot on the context determined by time or hardware constraints. Furthermore, this chapter could show that even different parameter settings leading to qualitatively different estimates are plausible for different tasks.

3.6 Acknowledgements

We thank Joachim Weickert and Michael Felsberg for fruitful discussions, and Fabio Solari (and his student Javier Diaz), Temujin Gautama and Marc van Hulle for providing computed flow fields for the Lucas-Kanade and the phase-based approach, respectively. The relevant publications from the author are [Kalkan et al., 2004a, Kalkan et al., 2004b, Kalkan et al., 2005].

Chapter 4

Improving Junction Detection by Semantic Interpretation

The previous chapter showed that optic flow estimation at edge-like structures is biased, and it suggested that this bias can be removed by using optic flow information at junction-like and other 2D structures. However, extraction of junction-like structures is biased and ambiguous itself, and this bias and ambiguity might cause further bias and ambiguity if they are not corrected. The current chapter proposes a solution in the form of a feedback mechanism for this bias and ambiguity in junction detection.

Junctions are utilized in computer vision and image processing for tasks that especially require finding correspondences between different views of the same scene, mainly due to their *distinctiveness*, *seldomness* and *stability*. Correct localization of junctions[1] is crucial because even small errors in localization lead to wrong interpretations of the scene [Rohr, 1992]. Nevertheless, it is shown in [Deriche and Giraudon, 1993, Rohr, 1992] that energy-based junction detection methods smooth out junctions and face the problem of wrong localization.

Junctions also have the property of being *interpretable*: *i.e.*, you can construct a meaningful interpretation about how the junction is formed, as proposed in [Parida et al., 1998, Rohr, 1992]. Such a semantic interpretation can be utilized in rigid body motion estimation [Pilz et al., 2007], depth estimation [Waltz, 1975], feature matching etc. and is more informative than just a junctionness measure and

[1] In this chapter, corners are considered to be a special case of junctions, and the term 'corner' is avoided. While doing so, the understanding of corner as texture-like structure is excluded.

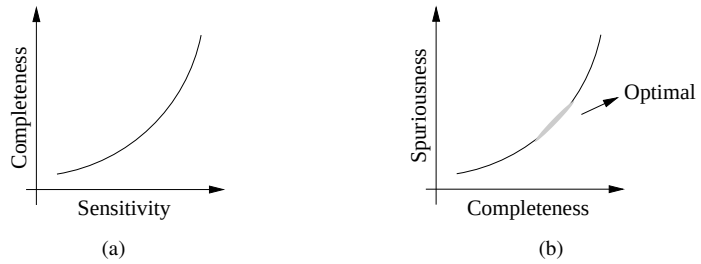


Figure 4.1: The relations between sensitivity, completeness and spuriousness.

can be used in identification of junctions and in correspondence finding.

Junction detectors, no matter what the underlying methods are, have to make a decision about the 'junctionness' of an image area. The decision is made by a set of automatically or manually set thresholds (on a set of measures) that determine the *sensitivity* of the algorithm to contrast (in most of the cases, a high threshold means low sensitivity and vice versa). On the other hand, a method that utilizes a junction detector requires the detector to be *complete*: *i.e.*, the detector should be able to detect all the junctions that represent the image.

The relation between sensitivity and completeness presumably looks like as plotted in figure 4.1(a). Increasing the sensitivity increases not only the completeness of a detector[2] but also increases the amount of false-positives, or 'spuriousness', of the detector as illustrated in figure 4.1(b). These observations suggest that spuriousness and completeness are two competing objectives that make the problem of junction detection a multi-objective optimization problem, and it is known that a multi-objective optimization problem with competing objectives does not have a global optimum, but a set of optimal solutions which are called Pareto-optimal (see, figure 4.1(b) and *e.g.*, [Coello, 1999]).

Junction detection algorithms face this completeness-spuriousness 'problem' (called CS-problem in the rest of the chapter) because detecting junctions in real images is an ill-posed problem at the level of feature-processing due to the fact that identifying *accurate* and *complete* boundaries and junctions of the objects requires an object-recognition step which is supposed to happen at a higher level in a vision system.

This chapter shows that junction detectors can be followed by a 'semantic interpretation' step as a feedback mechanism to achieve a better completeness-to-spuriousness ratio. This is achieved by (1)

[2] The exact shape of this relation might be different in real world; however, the authors claim that completeness should be still an increasing function of sensitivity in any case.

increasing the sensitivity of the junction detectors (by decreasing their thresholds), (2) improving the positioning of the detection step using a regularity or intersection consistency step and then (3) extracting the semantic interpretation of the junctions to filter spurious junctions.

The intersection-consistency, or regularity measure that is implemented in this chapter is based on the observation that the position of a junction is defined by the intersection of its edges [Parida et al., 1998]. This measure is similar to the $\mathcal{R}()$ function in equation 5 of [Parida et al., 1998] and the regularity function $S()$ in equation 4 of [Forstner, 1994]. Both of these functions are based on the local image gradient whereas the method of this chapter utilizes another edgeness measure called intrinsic dimensionality (see section 2.1 and [Krüger and Felsberg, 2003] for details).

For semantic interpretation of junctions, this chapter proposes representing junctions in terms of their constituents (*i.e.*, the edges that form the junctions) and how they form the junctions (*i.e.*, the directions of the constituent edges). There have already been studies related to the representation of the junctions (see, *e.g.*, [Baker et al., 1998, Hahn and Krüger, 2000, Parida et al., 1998, Rohr, 1992, Simoncelli and Farid, 1996]): In [Simoncelli and Farid, 1996], steerable wedge filters are developed for analyzing the orientation maps of edges and junctions, without creating an explicit representation of these features; in [Parida et al., 1998], by assuming that the number of junctions is known, a junction model is fitted to the data by minimizing an energy function; in [Baker et al., 1998], parameters of junctions with just two edges (*i.e.*, corners) are extracted by using dimensionality reduction techniques. In [Rohr, 1992], assuming that the number of edges is known, a junction is extracted as a composition of L-junctions by fitting a parametric model to the image data. In [Hahn and Krüger, 2000], corners are detected, and their representations are created using Hough lines, and these corners are merged to create junction representations. The current chapter employs a simple method that extracts the representation of a junction by analyzing the clusters in its orientation histogram. While doing so, *it does not make any assumptions* about the junction and is able to create representations of any junction configuration.

The contributions of this chapter are (1) proposal of a new method for creating representations of junctions and (2) pinpointing a common problem in all junction detectors (namely, the CS-problem) and (3) proposing a way to improve junction detectors with respect to this problem. The method is tested on natural images, using three different junction detectors: SUSAN, Harris operators and the intrinsic dimensionality. The aim of this chapter is not to compare the performance of these methods but to propose a feedback mechanism to improve them. For a comparison of a set of interest point and junction

detectors, the interested reader is directed to [Schmid et al., 2000].

As mentioned in chapter 1, biological vision systems can cope with the ambiguities in the visual information in the early stages of visual processing by using feedback mechanisms. This chapter is considered to be application of such a feedback mechanism for a better detection of junctions.

4.1 Junction Detection Algorithms

This section briefly describes the main approaches for junction detection without any claim of being complete (see, *e.g.*, [Deriche and Giraudon, 1993, Schmid et al., 2000, Smith, 1997] for more detailed reviews; [Harris and Stephens, 1988] for the Harris operator, and [Smith and Brady, 1997] for the SUSAN operator, respectively).

Since the first attempts around late 1970s, there have been quite a number of works on the detection of junctions. The methods can be roughly divided into three main categories:

- **Contour-based:** These methods involve extracting an edge representation and then processing the maxima curvature or the linking of the edges to find the junctions (see, *e.g.*, [Asada and Brady, 1986, Deriche and Giraudon, 1990, Horaud and Veillon, 1990]).
- **Signal-based:** These methods involve finding the junctions by directly using the image intensities. The second order derivatives of intensities, usually called the Hessian matrix [Beaudet, 1978, Dreschler and Nagel, 1982], autocorrelation function of the image patch [Forstner, 1994, Harris and Stephens, 1988, Moravec, 1980] are the main tools used by such approaches.
- **Template-based:** These methods detect junctions that match certain templates [Parida et al., 1998, Rohr, 1992].

4.1.1 Harris Operator

The Harris operator [Harris and Stephens, 1988] is an improvement of the Moravec operator [Moravec, 1980]. The Moravec operator extracts image features by shifting the image patch in a set of directions and measuring the correlation between the original image patch and the shifted image patch.

$$E^m(x,y) = \sum_{u,v} \left| I(x+u, y+v) - I(u,v) \right|^2, \tag{4.1}$$

where (u, v) are the image patch coordinates; I is the image intensity; and, (x, y) is the shift. The Moravec operator considered the shifts (x, y) from the set $\{(1, 0), (1, 1), (0, 1), (-1, 1)\}$.

The shifts fall into three cases: (1) the image patch is homogeneous, and the shifts result in small changes; (2) the image patch is edge-like, and there is a big change in one direction; and, (3) the image patch is a junction, and there are changes in every directions.

The main improvement of the Harris operator over the Moravec operator was to consider the shifts in all directions by making use of the derivatives of the image patch in u and v directions instead of a set of discrete directions [Harris and Stephens, 1988]:

$$E^h(x, y) = (x, y)\, \mathbf{M}\, (x, y)^T, \tag{4.2}$$

where $\mathbf{M}$ is the second derivative matrix of $E^m(x, y)$:

$$\mathbf{M} = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}. \tag{4.3}$$

Differences between image structures are reflected then in the eigenvalues λ_1 and λ_2 of $\mathbf{M}$: (1) λ_1 and λ_2 are close to zero when the image patch is homogeneous; (2) One of the eigenvalues is high and the other eigenvalue is close to zero when the image patch is edge-like; and (3) Both λ_1 and λ_2 are high when the image patch is junction-like.

4.1.2 SUSAN Operator

The Smallest Univalue Segment Assimilating Nucleus (SUSAN) operator is based on placing a circular mask M at each pixel and comparing the center pixel with the pixels of the mask: (1) If the image patch is homogeneous, the mask should contain similar elements, and intensity differences between the pixels and the center of the mask should approximately sum up to zero. (2) If the image patch is edge-like, approximately half of the mask should have different intensity than the center of the mask. (3) If the image patch is junction-like, 1/4 of the mask should have different intensity than the center of the mask.

The area that have the similar intensity with the center (u_0, v_0) of the mask is measured with:

$$n(u_0, v_0) = \sum_{u,v \in M} e^{-(I(u,v)-I(u_0,v_0))^6/t}, \tag{4.4}$$

where t determines the width of the exponent function and is chosen experimentally.

4.2 Improving Localization

The approach of the current chapter is to detect junctions (using Harris, Susan or *iD*), and then to compute a junction regularity measure, called intersection-consistency (*IC*), in the neighborhood of the detected junctions. The new *improved* position of a junction is determined by the local maximum of *IC* in the 3x3-neighborhood.

IC is measured by checking whether the pixels in the image patch point towards the center of the patch or not. Pointing towards the center of the local image patch is measured by the distance between the center $\mathbf{p}_c$ and the line going through the pixel. The line is defined according to the position of the pixel $\mathbf{p}$ and the computed orientation information $\theta_{\mathbf{p}}$. The weighted average of these distances then defines the intersection consistency at $\mathbf{p}_c$:

$$ic(\mathbf{p}_c) = \int [c_{i1D}(\mathbf{p})]^2 [1 - d(l^{\mathbf{p}}, \mathbf{p}_c)/d(\mathbf{p}, \mathbf{p}_c)] \, \mathrm{d}\mathbf{p}, \tag{4.5}$$

where $\mathbf{p}$ is the index of the pixels in image patch P; $c_{i1D}(\mathbf{p})$ is the confidence for i1D of pixel $\mathbf{p}$; $l^{\mathbf{p}}$ is the line going through pixel $\mathbf{p}$ with a slope defined according to the orientation $\theta_{\mathbf{p}}$; $d(l^{\mathbf{p}}, \mathbf{p}_c)$ is the distance between $l^{\mathbf{p}}$ and $\mathbf{p}_c$; and, $d(\mathbf{p}, \mathbf{p}_c)$ is the distance between $\mathbf{p}$ and $\mathbf{p}_c$. Note that $d(l^{\mathbf{p}}, \mathbf{p}_c)$ is normalized with $d(\mathbf{p}, \mathbf{p}_c)$ in order to make the weights be in the range $[0, 1]$

The distances between the center of the local image patch and the lines through the pixels is weighted by $(c_{i1D})^2$ because the computed orientation information is defined only for edge-like structures, and *IC* by definition involves intersection consistency of edge-like structures.

The $ic(P)$ value will be high (1) if the image patch has only one edge which goes through the center of the patch or (2) if the image patch has a junction whose intersection point is located at the center of the patch.

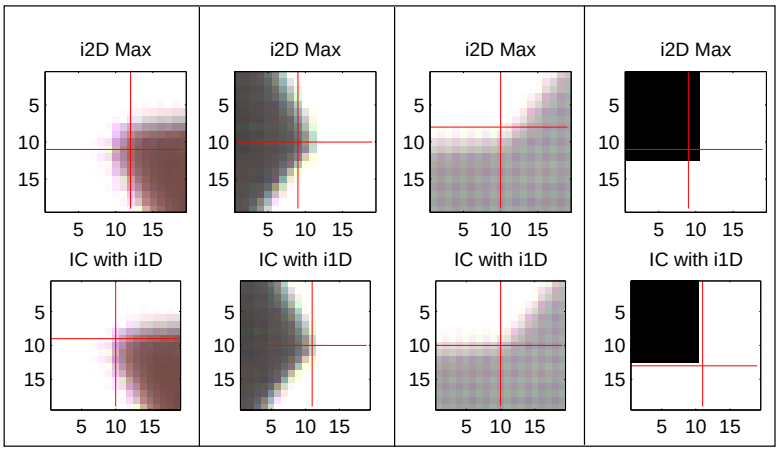


Figure 4.2: Illustration of the maximum *IC* for a few examples.

The max of *IC* is shown for a few examples in figure 4.2.

For comparison, $S()$ function of [Forstner, 1994] is given below which is similar to equation 4.5 (a different function for the same purpose can be found in [Parida et al., 1998]):

$$S(p,\sigma) = \iint d^2(p,l_q) \parallel \nabla g(q) \parallel^2 G_\sigma(p-q)\, \mathrm{d}q. \tag{4.6}$$

where p is the center of the image patch; q denotes image points in the image patch; $d(p,l_q)$ is the distance of center point p to the line l_q defined by q; and, $\nabla g(q)$ is the intensity gradient (g_x, g_y).

4.3 Semantic Interpretation of Junctions

This section describes how the semantic interpretation (*SI*) of a junction is estimated. This estimation process does not make any assumptions on the configuration of the edges (interested reader is directed to [Waltz, 1975] for different classes of junctions and their properties). The *SI* of a junction that does not fall into any meaningful junction category can be used to detect false-positives.

For computing the *SI*, a junction is represented by a set of rays $r_1 ... r_n$ corresponding to the set of n edges that intersect at the junction. Each ray r_i represents a specific edge i, defined as a half-line expanding from the intersection point in a certain direction $\tilde{\theta}_i$. Another parameter c_i can be introduced denoting the confidence of an edge which can be used as a weight when the *SI* of a junction is utilized.

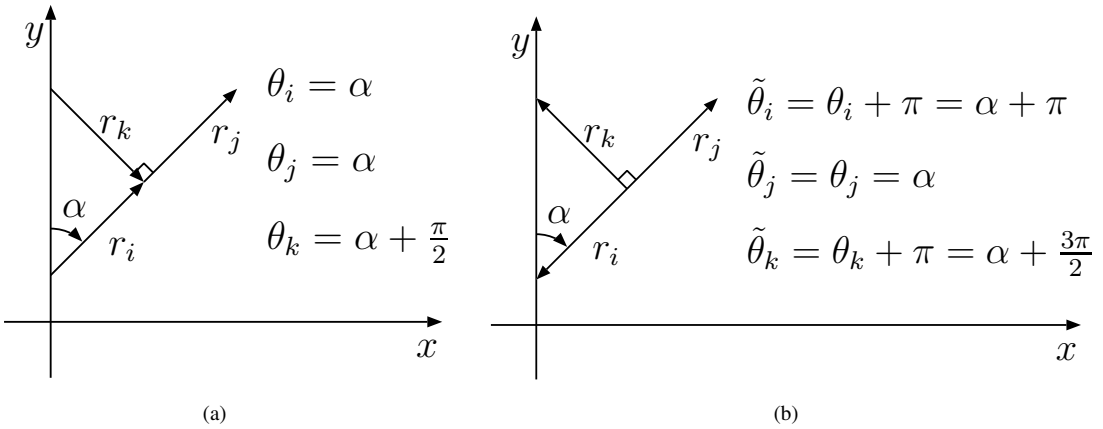


Figure 4.3: Image- and junction-relative representations of the directions of the edges of an example junction. (a) Image-relative directions. (b) Junction-relative directions. Note that the direction of the vectors that represent the orientation of the rays change in (b).

With these parameters, the semantic interpretation $SI(\mathcal{J})$ of a junction $\mathcal{J}$ can be defined as follows:

$$SI(\mathcal{J}) = \{r_1, ..., r_n\} = \left\{\left(c_1, \tilde{\theta}_1\right), ..., \left(c_n, \tilde{\theta}_n\right)\right\}. \tag{4.7}$$

$\tilde{\theta} \in [0, 2\pi)$ is the junction-relative orientation; *i.e.*, it is the orientation defined with respect to the center of the junction. The image relative orientation $\theta_p \in [0, \pi)$ of a pixel p at (x_p, y_p) needs to be transformed to junction-relative orientation for junction $\mathcal{J}$ at (x, y) as follows:

$$\tilde{\theta}_p = \begin{cases} \theta_p, & \text{if } tan^{-1}[(x - x_p)/(y - y_p)] < \pi, \\ \theta_p + \pi, & \text{if } tan^{-1}[(x - x_p)/(y - y_p)] \geq \pi. \end{cases} \tag{4.8}$$

In figure 4.3, the image-relative and junction-relative orientations of two edges are shown for a junction.

The junction-relative orientation $\tilde{\theta}_i$ for each ray r_i is extracted by finding the dominant orientations in the neighborhood N of the junction $\mathcal{J}$. The set of pixels in N that point towards the center of the junction $\mathcal{J}$ can be constructed as follows:

$$O_{\mathcal{J}} = \left\{\tilde{\theta}_p \ : \ p \in N \text{ and } d(l_p, (x, y)) < T\right\}, \tag{4.9}$$

where l_p is the line defined by the pixel p with orientation $\tilde{\theta}_p$.

The number of rays and their orientations are determined by the clusters in the histogram $H_l(O_{\mathcal{J}})$ where l is the index of the bins. The set of clusters $\{C_m\}$ is the set of H_l (1) where the first derivative $\delta H_l/\delta l$ changes sign, and (2) where the energy (*i.e.*, the number of elements in the bin) is above a threshold. Figure 4.4 shows the rays extracted from an example junction.

A junction is marked as false-positive if $n < 2$ or $n = 2$ and $\tilde{\theta}_1 \simeq \tilde{\theta}_2$.

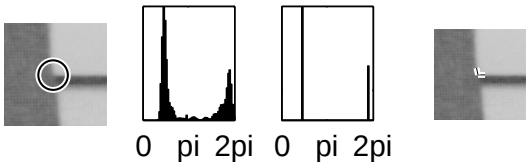


Figure 4.4: Illustration of the *SI* of a junction. From left to right: the junction marked with a circle; the distribution of junction-relative orientation; detected ray orientations; estimated *SI*.

4.4 Results and Discussions

In this section, we evaluate the intersection consistency function and semantic interpretation on real examples. The SUSAN implementation is taken from the author of [Smith and Brady, 1997], and the Harris implementation is taken from [Noble, 1989]. The parameters of the SUSAN, the Harris operators and *iD* are provided in table 4.1.

Table 4.1: The parameters used in the experiments.

Algorithm	Low sensitivity	High sensitivity
SUSAN	$brightness > 20$	$brightness > 13$
Harris	$E > 1000$	$E > 300$
iD	$c_{i2D} > c_{i1D}$ & $c_{i2D} > c_{i0D}$	$c_{i2D} > 0.3$

In figure 4.5, the results of the three junction detection methods are presented for several image patches extracted from real images. For each example and each method, original detection results, improved positions and the *SI* are plotted.

The examples demonstrate that junction detection methods face the problem of wrong localization as pointed out in [Deriche and Giraudon, 1993, Rohr, 1992]. Moreover, it is very likely that the methods produce false positives especially visible in the case of SUSAN and *iD*. However, figure 4.5 shows that

the effect of the positioning problem can be decreased, and false positives are removed by using the *SI* of the junctions.

As mentioned at the beginning of the chapter, junction detectors have a level of sensitivity that cannot be made universal; *i.e.*, it is not possible to adjust the parameters of a junction detector in order to detect every junction without producing a big ratio of false positives.

Figure 4.6 displays a set of examples for SUSAN, Harris and *iD* with low thresholds (table 4.1). The thresholds have been decreased so that the detectors can detect the junctions that they have missed with their default parameter values (*e.g.*, the junction in the center of figure 4.6(a)). Figure 4.6(a) suggests that increasing sensitivity of a detector can help in detecting low contrast however important junctions. On the other hand, figure 4.6 shows that all methods produce spurious results when the sensitivity is high, especially in the case of SUSAN and *iD*. However, by making use of *IC* and *SI*, it is possible to get rid of most of the spurious junctions and detect a wider range of junctions with more accuracy even for high sensitivity levels.

4.5 Summary

This chapter proposed two methods for improving the detection and the representation of junctions: (1) an operator called *intersection consistency* that measures how consistent a junction is with its neighborhood, and (2) a way to create semantic interpretation of junctions.

As shown in [Deriche and Giraudon, 1993, Rohr, 1992], energy-based junction detectors face the problem of wrong positioning. Using the intersection consistency method introduced in this chapter, better positioning has been achieved for the different junction detection algorithms.

The issue of having thresholds on the detection of junctions with respect to the 'completeness' and 'spuriousness' of the detection has also been addressed. By making use of the semantic interpretation as a feedback mechanism, it has been shown that the performance of junction detectors can be improved.

4.6 Acknowledgements

This work is a product of close colloboration with Shi Yan from Aalborg University Copenhagen, who worked on this topic for his bachelor project [Yan, 2005].

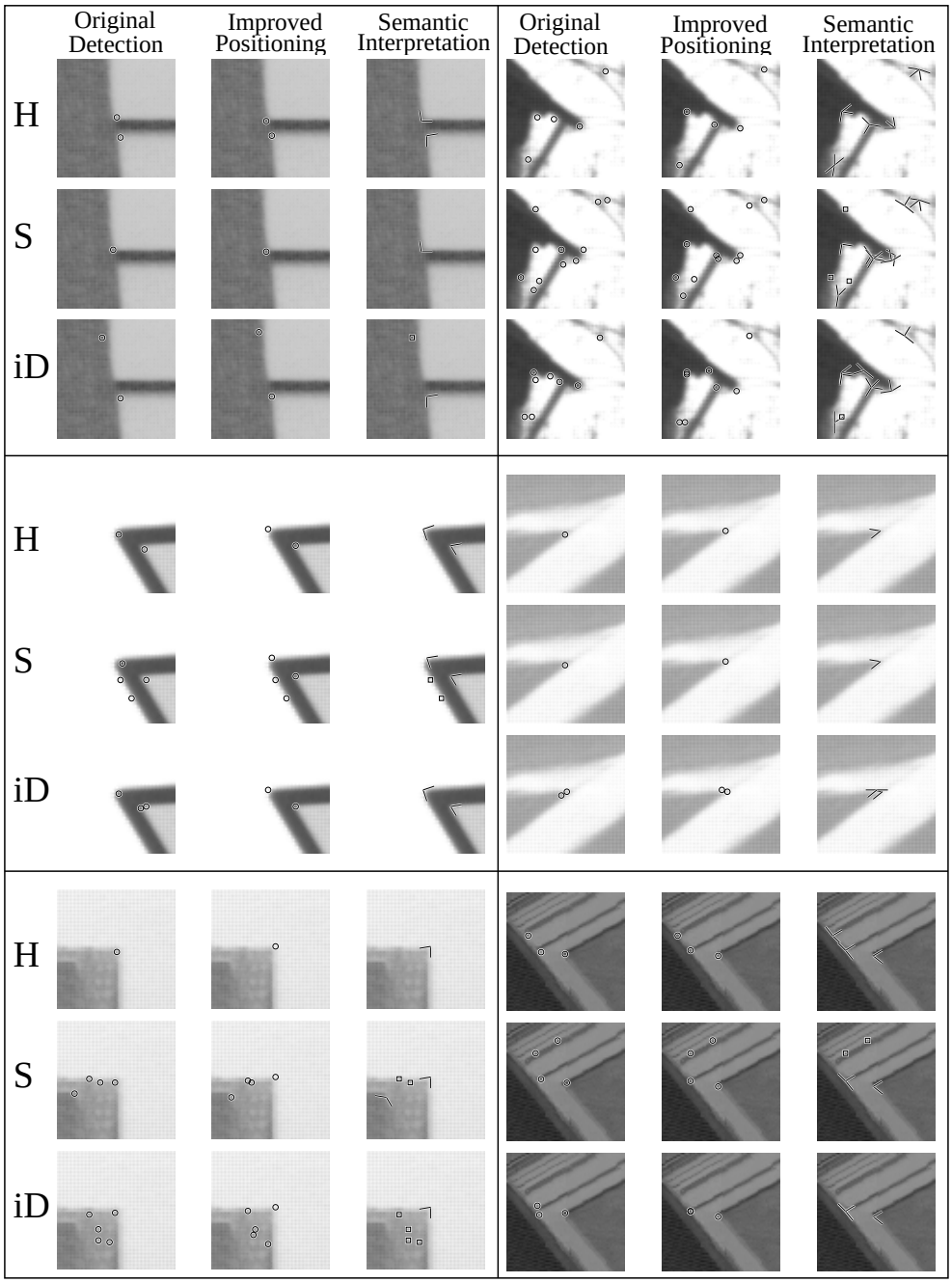


Figure 4.5: A set of example junctions and the results of junction detectors and the results of *IC* and *SI* on these results. For each example, S, H and iD denote SUSAN, Harris and *iD* respectively. In each example, the first column shows the original detections of the algorithms; the second column shows the effect of improved positioning via *IC*; and, the third column shows the estimated *SI* and how it can be used to get rid of spurious junctions. Spurious junctions that are estimated with *SI* are marked in small squares.

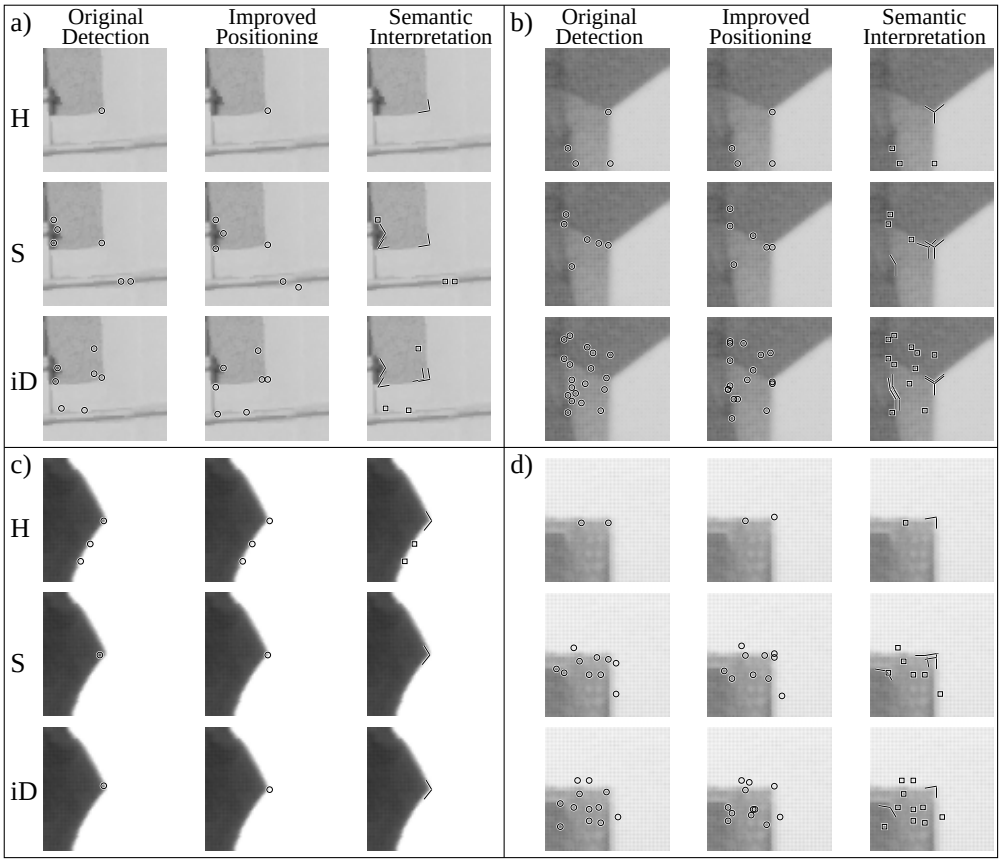


Figure 4.6: The effect of high 'sensitivity' on the performance of junction detectors. Junction detectors can now detect low contrast junctions that they miss with low sensitivity. H and S denote Harris and SUSAN respectively. For each subfigure, the first column shows original detection results, the second the results of improved positioning with *IC* and the third the *SI*. Spurious junctions that are estimated with *SI* are marked in small squares.

Chapter 5

Statistical Relation between Local Image Structures and Local 3D Structure

In section 2.2, 2D and 3D features were introduced to represent homogeneous and edge-like structures. The relation between the 2D and 3D structures are very crucial for understanding how these structures might be utilized better in depth extraction in early cognitive vision. In fact, it is argued already that different 2D structures can be made use of in different ways for 3D shape interpretation (see, *e.g.*, [Barrow and Tenenbaum, 1981]).

Surface interpolation studies widely make use of the assumption that two image points which do not have any contrast difference in between are on the same 3D surface (see, *e.g.*, [Grimson, 1983]). This assumption is usually called *'no news is good news'* in the literature.

The current chapter investigates the relation between 2D and 3D structures (1) to understand how these structures can be used better for depth extraction in early cognitive vision (in this sense, this chapter proves the hypothesis which chapter 7 uses) and (2) to make an analysis of the 'no news is good news' assumption. Namely, this chapter derives the likelihood of observing a certain local 3D structure, given its 2D projection (*i.e.*, underlying local image structure). For this, range data with real-world color information is used as ground truth data.

5.1 Relevant Studies

There have been only a few studies that have analyzed the 3D world from range data [Howe and Purves, 2004, Huang et al., 2000, Potetz and Lee, 2003, Yang and Purves, 2003], and these works have only been first-order. In [Yang and Purves, 2003], the distribution of roughness, size, distance, 3D orientation, curvature and independent components of surfaces was analyzed. Their major conclusions were: (1) local 3D patches tend to be saddle-like, and (2) natural scene geometry is quite regular and less complex than luminance images. In [Huang et al., 2000], the distribution of 3D points was analyzed using co-occurrence statistics and 2D and 3D joint distributions of Haar filter reactions. They showed that range images are much simpler to analyze than optical images and that a 3D scene is composed of piecewise smooth regions. In [Potetz and Lee, 2003], the correlation between light intensities of the image data and the corresponding range data as well as surface convexity were investigated. They could justify the event that brighter objects are closer to the viewer, which is used by shape from shading algorithms in estimating depth. In [Howe and Purves, 2002, Howe and Purves, 2004], range image statistics were analyzed for explanation of several visual illusions.

The first-order analysis of this chapter differs from the above-mentioned studies. For 2D local image patches, existing studies have only considered light intensity. As for 3D local patches, the most complex considered representation has been the curvature of the local 3D patch. In this chapter, however, a higher-order representation of the 2D local image patches and the 3D local patches are created; we represent 2D local image patches using homogeneous, edge-like, corner-like or texture-like structures, and 3D local patches using continuous surfaces and different kinds of 3D discontinuities. By this, established local image structures are related to their underlying 3D structures.

5.2 Local 2D and 3D Structures

This chapter distinguishes between the following local 2D structures (examples of each structure is given in figure 2.1) using intrinsic dimensionality (see section 2.1): homogeneous image patches, edge-like structures, corners and textures.

To our knowledge, there does not exist a systematic and agreed classification of local 3D structures like there is for 2D local image structures (*i.e.*, homogeneous patches, edges, corners and textures).

Intuitively, the 3D world consists of continuous surface patches and different kinds of 3D discontinuities. During the imaging process (through the lenses of the camera or the eye), 2D local image structures are generated by these 3D structures together with the illumination and the reflectivity of the environment.

With this intuition, any 3D scene can be decomposed geometrically into surfaces and 3D discontinuities. In this context, the local 3D structure of a point can be a:

- Surface Continuity: The underlying 3D structure can be described by one surface whose normal does not change or changes smoothly (see figure 5.1(a)).
- Regular Gap Discontinuity: The underlying 3D structure can be described by a small set of surfaces with a significant depth difference. The 2D and 3D views of an example gap discontinuity are shown in figure 5.1(d).
- Irregular Gap Discontinuity: The underlying 3D structure shows high depth-variation that can not be described by two or three surfaces. An example of an irregular gap discontinuity is shown in figure 5.1(e).
- Orientation Discontinuity: The underlying 3D structure can be described by two surfaces with significantly different 3D orientations that meet at the center of the patch. This type of discontinuity is produced by a change in 3D orientation rather than a gap between surfaces. An example for this type of discontinuity is shown in figure 5.1(c).

5.3 Methods

In this subsection, we define our measures for the three kinds of discontinuities that are described in section 5.2; namely, gap discontinuity, irregular gap discontinuity and orientation discontinuity. The measures for gap discontinuity, irregular gap discontinuity and orientation discontinuity of a patch P will be denoted by $\mu_{GD}(P)$, $\mu_{IGD}(P)$ and $\mu_{OD}(P)$, respectively. The reader who is not interested in the technical details can jump directly to section 5.4.

3D discontinuities are detected in studies which involve range data processing, using different methods and under different names like two-dimensional discontinuous edge, jump edge or depth discontinuity for gap discontinuity; and, two-dimensional corner edge, crease edge or surface discontinuity for orientation discontinuity [Bolle and Vemuri, 1991, Hoover et al., 1996, Shirai, 1987].

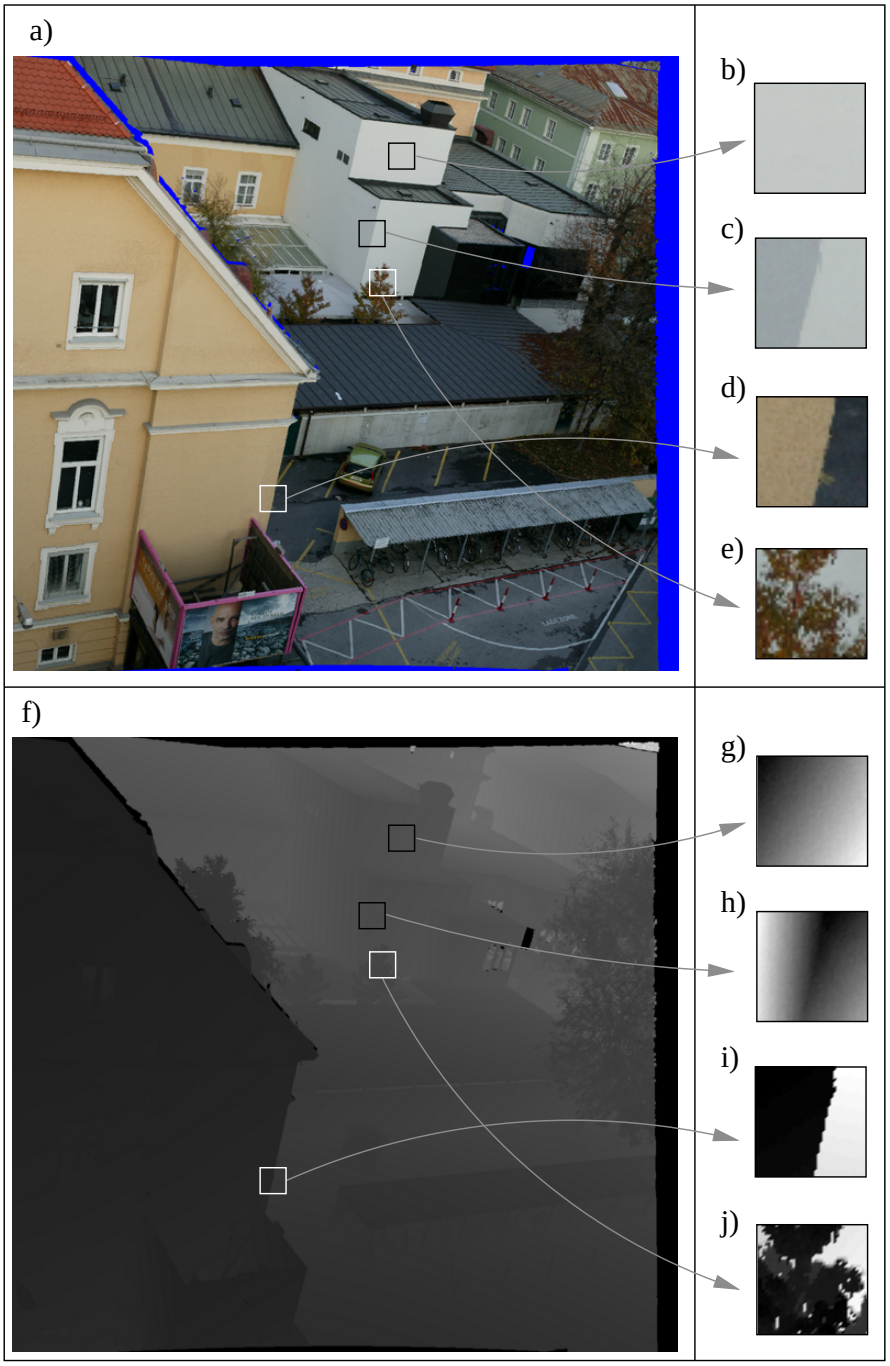


Figure 5.1: Illustration of the types of 3D discontinuities. **(a)** 2D image. **(b)** Continuity. **(c)** Orientation discontinuity. **(d)** Gap discontinuity. **(e)** Irregular gap discontinuity. **(f)-(j)** The range images corresponding to (a)-(e). Note that the range images are scaled independently for better visibility.

Figure 5.2: 10 of the 20 3D data sets used in the analysis. The points without range information are marked in blue. The gray image shows the range data of the top-left scene. The resolution range is [512-2048]x[390-2290] with an average resolution of 1140x1001.

In our analysis, we used chromatic range data of outdoor scenes which were obtained from Riegl UK Ltd. (`http://www.riegl.co.uk/`). There were 20 scenes in total, 10 of which are shown in figure 5.2. The range of an object which does not reflect the laser beam back to the scanner or is out of the range of the scanner cannot be measured. These points are marked with blue in figure 5.2 and are not processed in our analysis. The resolution range of the data set is [512-2048]x[390-2290] with an average resolution of 1140x1001.

5.3.1 Measure for Gap Discontinuity: μ_{GD}

Gap discontinuities can be measured or detected in a similar way than edges in 2D images; edge detection processes RGB-coded 2D images while for a gap discontinuity, one needs to process XYZ-coded 2D images [1]. In other words, gap discontinuities can be measured or detected by taking the second order derivative of XYZ values [Shirai, 1987].

Measurement of a gap discontinuity is expected to operate on both the horizontal and the vertical axes of the 2D image; that is, it should be a two dimensional function. The alternative is to discard the topology and do an 'edge-detection' in sorted XYZ values, *i.e.*, to operate as a one-dimensional function.

[1] Note that XYZ and RGB coordinate systems are not the same. However, detection of gap discontinuity in XYZ coordinates can be assumed to be a special case of edge detection in RGB coordinates.

Although we are not aware of a systematic comparison of the alternatives, for our analysis and for our data, the topology-discarding gap discontinuity measurement captured the underlying 3D structure better (of course, qualitatively, *i.e.*, by visual inspection). Therefore, we have adopted the topology-discarding gap discontinuity measurement in the rest of the chapter.

For an image patch P of size $N \times N$, let,

$$\begin{aligned} \mathcal{X} &= \text{ascending\_sort}(\{X_i \mid i \in P\}), \\ \mathcal{Y} &= \text{ascending\_sort}(\{Y_i \mid i \in P\}), \\ \mathcal{Z} &= \text{ascending\_sort}(\{Z_i \mid i \in P\}), \end{aligned} \tag{5.1}$$

and also, for $i = 1, .., (N \times N - 2)$,

$$\begin{aligned} \mathcal{X}^{\Delta} &= \{ \mid (\mathcal{X}_{i+2} - \mathcal{X}_{i+1}) - (\mathcal{X}_{i+1} - \mathcal{X}_i) \mid \}, \\ \mathcal{Y}^{\Delta} &= \{ \mid (\mathcal{Y}_{i+2} - \mathcal{Y}_{i+1}) - (\mathcal{Y}_{i+1} - \mathcal{Y}_i) \mid \}, \\ \mathcal{Z}^{\Delta} &= \{ \mid (\mathcal{Z}_{i+2} - \mathcal{Z}_{i+1}) - (\mathcal{Z}_{i+1} - \mathcal{Z}_i) \mid \}, \end{aligned} \tag{5.2}$$

where $\mathcal{X}_i, \mathcal{Y}_i, \mathcal{Z}_i$ represents 3D coordinates of pixel i. Equation 5.2 takes the absolute value of the $[+1, -2, +1]$ operator.

The sets $\mathcal{X}^{\Delta}, \mathcal{Y}^{\Delta}$ and $\mathcal{Z}^{\Delta}$ are the measurements of the jumps (*i.e.*, second order differentials) in the sets $\mathcal{X}, \mathcal{Y}$ and $\mathcal{Z}$, respectively. A gap discontinuity can be defined simply as a measure of these jumps in these sets. In other words:

$$\mu_{GD}(P) = \frac{h(\mathcal{X}^{\Delta}) + h(\mathcal{Y}^{\Delta}) + h(\mathcal{Z}^{\Delta})}{3}, \tag{5.3}$$

where the function $h : \mathcal{S} \rightarrow [0, 1]$ over the set $\mathcal{S}$ measures the homogeneity of its argument set (in terms of its 'peakiness') and is defined as follows:

$$h(\mathcal{S}) = \frac{1}{\#(\mathcal{S})} \times \sum_{i \in \mathcal{S}} \frac{s_i}{\max(\mathcal{S})}, \tag{5.4}$$

where $\#(\mathcal{S})$ is the number of the elements of $\mathcal{S}$, and s_i is the i^{th} element of the set $\mathcal{S}$. Note that as a

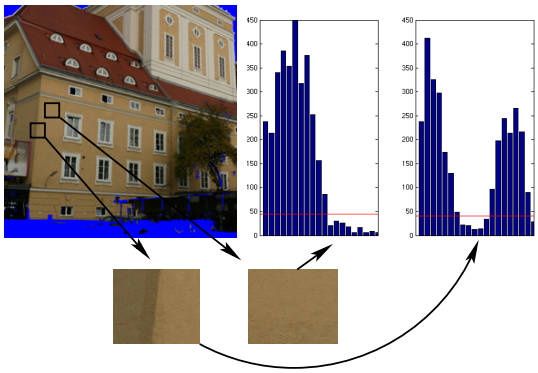


Figure 5.3: Example histograms and the number of clusters that the function $\psi(S)$ computes. $\psi(S)$ finds one cluster in the left histogram and two clusters in the right histogram. Red line marks the threshold value of the function. X axis denotes the values for 3D orientation differences.

homogeneous set (*i.e.*, a non-gap discontinuity) $\mathcal{S}$ produces a high $h(\mathcal{S})$ value, a gap discontinuity causes a low μ_{GD} value. Figure 5.5(c) shows the performance of μ_{GD} on one of our scenes shown in figure 5.2.

It is known that derivatives like in equations 5.1 and 5.2 are sensitive to noise. Gaussian-based functions could be employed instead. In this chapter, we chose simple derivatives for their faster computation times, and instead employed a more robust processing stage (*i.e.*, analyzing the uniformity of the distribution of derivatives) to make the measurement more robust to noise. As shown in figure 5.5(c), this method can capture the underlying 3D structure well.

5.3.2 Measure for Orientation Discontinuity: μ_{OD}

The orientation discontinuity of a patch P can be detected or measured by taking the 3D orientation difference between the surfaces that meet in P. If the size of the patch P is small enough, the surfaces can be, in practice, approximated by 2-pixel wide unit planes[2]. The histogram of the 3D orientation differences between every pair of unit planes forms one cluster for continuous surfaces and two clusters for orientation discontinuities.

For an image patch P of size $N \times N$ pixels, the orientation discontinuity measure is defined as:

$$\mu_{OD}(P) = \psi(H^n(\{\alpha(i,j) \mid i,j \in planes(P), i \neq j\})), \tag{5.5}$$

[2]Note that using bigger planes have the disadvantage of losing accuracy in positioning which is very crucial for the current analysis.

where $H^n(S)$ is a function which computes the n-bin histogram of its argument set $\mathcal{S}$; $\psi(\mathcal{S})$ is a function which finds the number of clusters in $\mathcal{S}$; $planes(P)$ is a function which fits 2-pixel-wide unit planes to 1-pixel apart points in P using Singular Value Decomposition[3]; and, $\alpha(i, j)$ is the angle between planes i and j.

For a histogram H of size N_H, the number of clusters is given by:

$$\psi(S) = \frac{\sum_{i=1}^{N_H+1} neq([H_i > \max(H)/10], [H_{i-1} > \max(H)/10])}{2}, \tag{5.6}$$

where the function neq returns 1 if its parameters are not equal and returns 0, otherwise; H_i represents the i^{th} element of the histogram H; H_0 and H_{N_H+1} are defined as zero; and, $\max(H)/10$ is an empirically set threshold. Figure 5.3 shows two example clusters for a continuous surface and an orientation discontinuity.

Figure 5.5(d) shows the performance of μ_{OD} on one of the scenes shown in figure 5.2.

5.3.3 Measure for Irregular Gap Discontinuity: μ_{IGD}

Irregular gap discontinuity of a patch P can be measured using the observation that an irregular-gap discontinuous patch in a real image usually consists of small surface fragments with different 3D orientations. Therefore, the spread of the 3D orientation histogram of a patch P can measure the irregular gap discontinuity of P.

Similar to the measure for orientation discontinuity defined in sections 5.3.1 and 5.3.2, the histogram of the differences between the 3D orientations of the unit planes (which are of 2 pixels wide) is analyzed. For an image patch P of size $N \times N$ pixels, the irregular gap discontinuity measure is defined as:

$$\mu_{IGD}(P) = h(H^n(\{\alpha(i, j) \mid i, j \in planes(P), i \neq j\})), \tag{5.7}$$

where $planes(P)$, $\alpha(i, j)$, $H^n(\mathcal{S})$ and $h(\mathcal{S})$ are as defined in section 5.3.2. Figure 5.5(e) shows the performance of μ_{IGD} on one of the scenes shown in figure 5.2.

[3] Singular Value Decomposition is a standard technique for fitting planes to a set of points. It finds the perfectly fitting plane if it exists; otherwise, it returns the least-square solution.

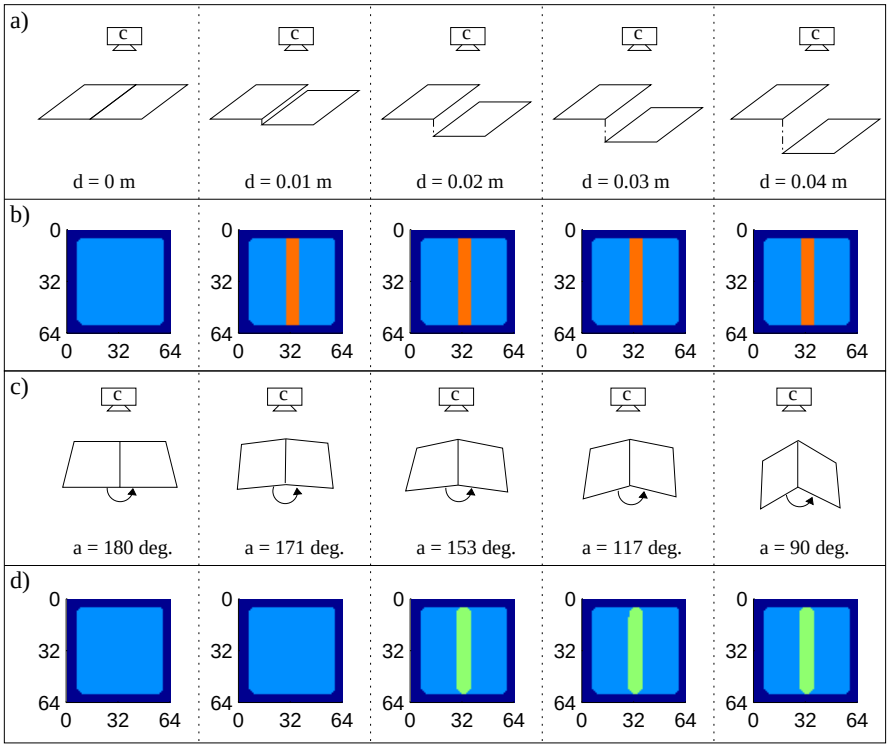


Figure 5.4: Results of the combined measures on artificial data. The camera and the range scanner are denoted by c. (a) Gap discontinuity tests. There are two planes which are separated by a distance d where d= $0, 0.01, 0.02, 0.03, 0.04$ meters. (b) The detected discontinuities. Dark blue marks the boundary points where the measures are not applicable. Blue and orange respectively correspond to detected continuities and gap discontinuities. (c) Orientation discontinuity tests. There are two planes which are connected but separated with an angle a where a=$180, 171, 153, 117, 90$ degrees. (d) The detected discontinuities. Dark blue marks the boundary points where the measures are not applicable. Blue and green respectively correspond to detected continuities and orientation discontinuities.

5.3.4 Combining the Measures

The relation between the measurements and the types of the 3D discontinuities are outlined in table 5.1 which entails that an image patch P is:

- gap discontinuous if $\mu_{GD}(P) < T_g$ and $\mu_{IGD}(P) < T_{ig}$,
- irregular-gap discontinuous if $\mu_{GD}(P) < T_g$ and $\mu_{IGD}(P) > T_{ig}$,
- orientation discontinuous if $\mu_{GD}(P) \geq T_g$ and $\mu_{OD} > 1$,
- continuous if $\mu_{GD}(P) \geq T_g$ and $\mu_{OD}(P) \leq 1$.

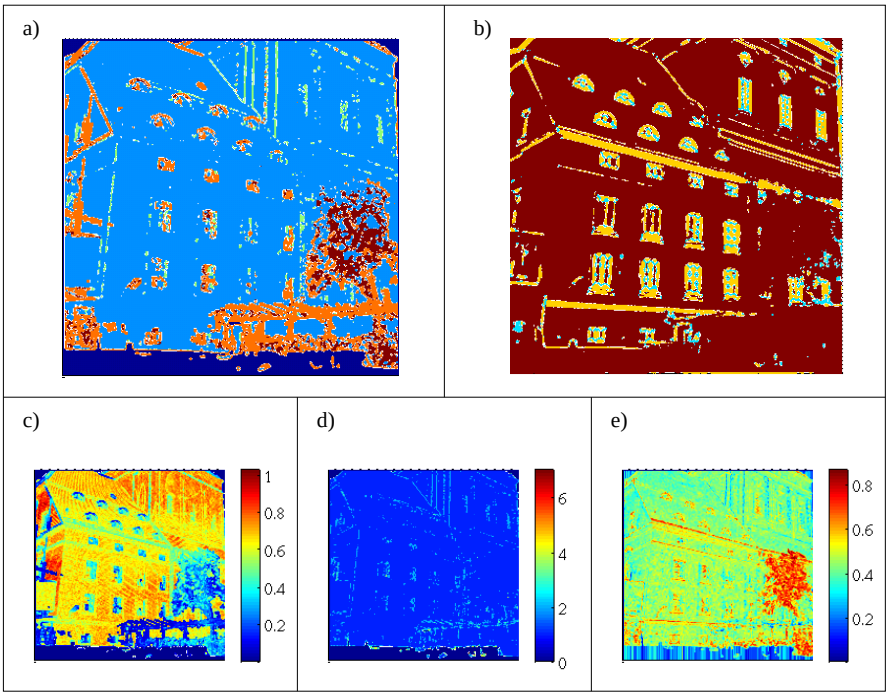


Figure 5.5: The 3D and 2D information for one of the scenes shown in figure 5.2. Dark blue marks the points without range data. **(a)** 3D discontinuity. Blue: continuous surfaces, light blue: orientation discontinuities, orange: gap discontinuities and brown: irregular gap discontinuities. **(b)** Intrinsic Dimensionality. Homogeneous patches, edge-like and corner-like structures are encoded in colors brown, yellow and light blue, respectively. **(c)** Gap discontinuity measure μ_{GD}. **(d)** Orientation discontinuity measure μ_{OD}. **(e)** Irregular gap discontinuity measure μ_{IGD}.

For our analysis, we have taken $N = 10$ and the threshold values $T_g = 0.4, T_{ig} = 0.6$ empirically. Bigger values for N means larger support region for the measures, in which case different kinds of 3D discontinuities might interfere in the patch. On the other hand, using smaller values would make the measures very sensitive to noise. Other thresholds T_g and T_{ig} are respectively set to 0.4 and 0.6. These values are empirically determined by testing the measures over a large set of samples. Different values for these thresholds may result in wrong classifications of local 3D structures and may lead to different results than presented in this chapter. Similarly, the number of bins, n, in H^n is empirically determined as 20.

Figure 5.4 displays the performance of the measures on two artificial scenes, one for gap discontinuity and one for orientation discontinuity for a set of depth and angle differences between planes. In the figure, the detected discontinuity type is shown for each pixel. The figure shows that gap discontinuity can be

Dis. Type	μ_{GD}	μ_{IGD}	μ_{OD}
Continuity	High value	Don't care	1
Gap Dis.	Low value	Low value	Don't care
Irregular Gap Dis.	Low value	High value	Don't care
Orientation Dis.	High value	Don't care	> 1

Table 5.1: The relation between the measurements and the types of the 3D discontinuities.

detected reliable even if the gap difference is low. The sensitivity of the orientation discontinuity measure is around 160 degrees. However, the sensitivity of the measures would be different in real scenes due to the noise in the range data.

For a real example scene from figure 5.2, the detected discontinuities are shown in figure 5.5(a), which suggests that the underlying 3D structure of the scene is reflected in figure 5.5(a).

An interesting example is smoothly curved surfaces. Such a surface would not produce *jumps* in equation 5.2 (since it is smooth), and therefore produce a high μ_{GD} value. Similarly, μ_{OD} would be 1 since there would be no peaks in the distribution of orientation differences. In other words, a curved surface would be classified as a continuity by the measures introduced above.

Note that this categorical combination of the measures appears to be against the motivation that has been provided for the classification of local 2D structures (section 2.1 and chapter 3) where we had advocated a continuous approach. There are two reasons: (1) With continuous 3D measures, the dimensionality of the results would be four (origin variance, line variance, a 3D measure and the normalized frequency of the signals), which is difficult to visualize and analyze. In fact, the number of triangles that had to be shown in figure 5.6 would be 12, and it would be very difficult to interpret all the triangles together. (2) It has been argued by several studies [Huang et al., 2000, Yang and Purves, 2003] that range images are much simpler and less complex to analyze than 2D images. This suggests that it might be safer to have a categorical classification for range images.

5.4 Results

For each pixel of the scene (except where range data is not available), the 3D discontinuity type and the intrinsic dimensionality are computed. Figure 5.5(a) and (b) shows the images where the 3D discontinuity and the intrinsic dimensionality of each pixel are marked with different colors.

Having the 3D discontinuity type and the information about the local 2D structure of each point, it

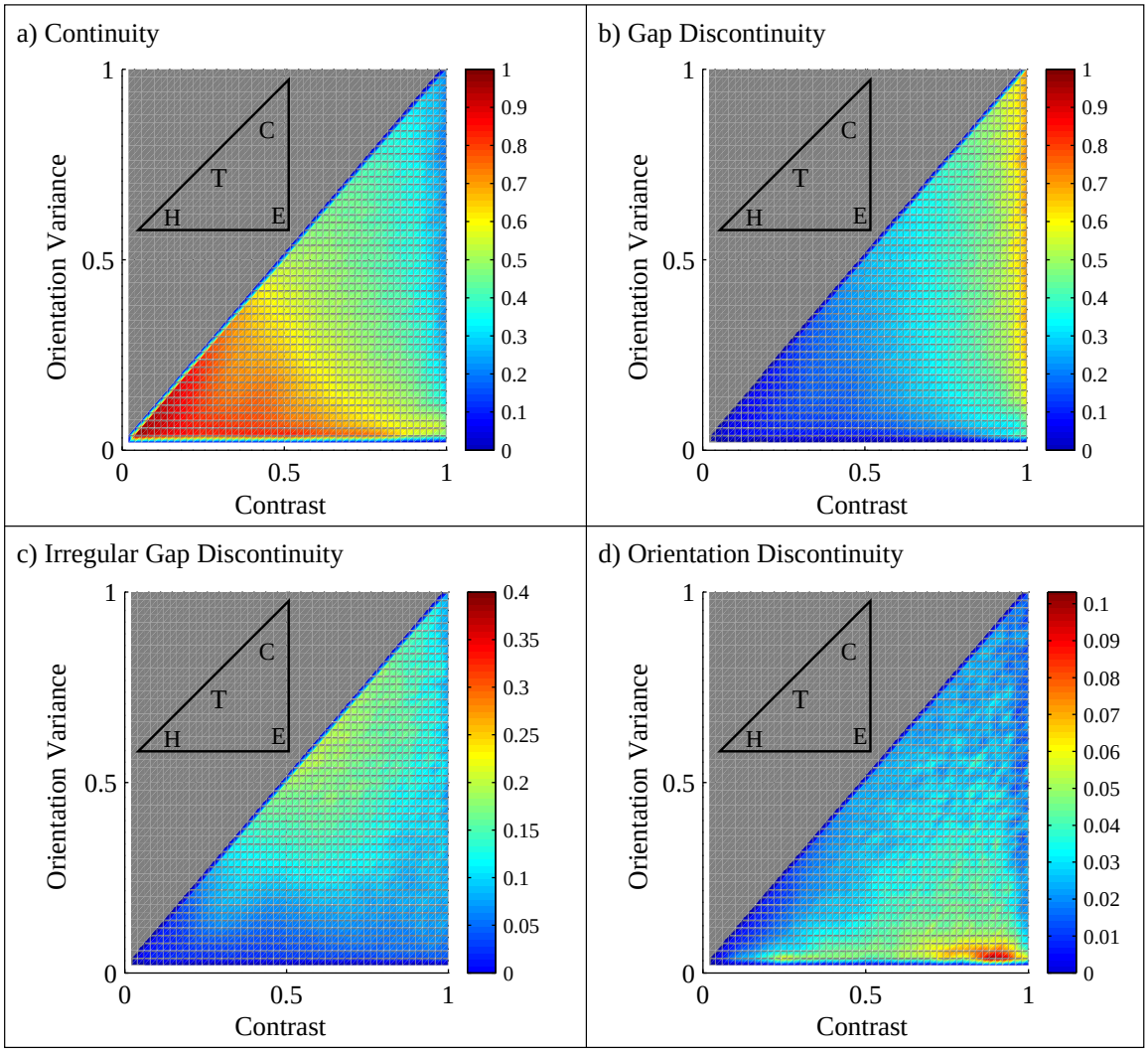


Figure 5.6: *P*(3D Discontinuity | 2D Structure). The distribution of local image structures is schematized for easy reference in the upper-left part of each subfigure (the letters C, E, H, T represent corner-like, edge-like, homogeneous and texture-like structures). **(a)** *P*(Continuity | 2D Structure). **(b)** *P*(Gap Discontinuity | 2D Structure). **(c)** *P*(Irregular Gap Discontinuity | 2D Structure). **(d)** *P*(Orientation Discontinuity | 2D Structure).

is straightforward to compute the conditional probability *P*(3D Discontinuity | 2D Structure), which is shown in figure 5.6. Note that the four triangles in figures 5.6(a), 5.6(b), 5.6(c) and 5.6(d) add up to one for all points of the triangle.

In figure 5.7, maximum likelihood estimates (MLE) of local 3D structures given local 2D structures are provided. Figure 5.7(a) shows the MLE from the distributions in figure 5.6. Due to high likelihoods,

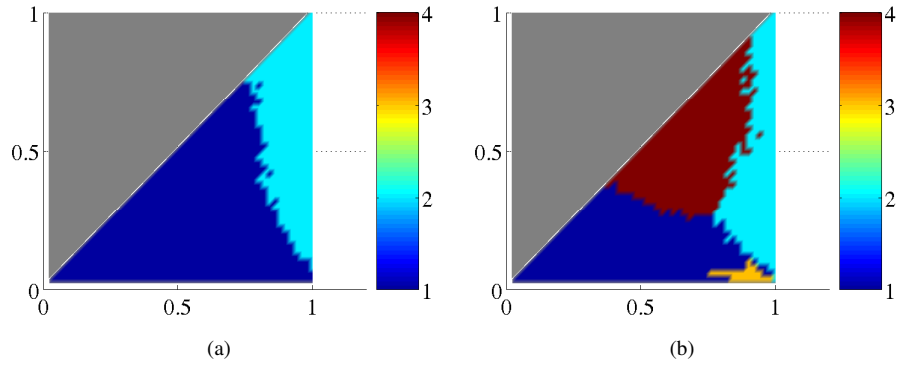


Figure 5.7: Maximum likelihood estimates of local 3D structures given local 2D structures. Numbers 1, 2, 3 and 4 represent continuity, gap discontinuity, orientation discontinuity and irregular gap discontinuity, respectively. **(a)** Raw maximum likelihood estimates. Note that the estimates are dominated by continuities and gap discontinuities. **(b)** Maximum likelihood estimates from normalized likelihood distributions: the triangles provided in figure 5.6 are normalized within themselves so that the maximum likelihood of $P(\ X \mid$ 2D Structure) is 1 for X being continuity, gap discontinuity, irregular gap discontinuity and orientation discontinuity.

gap discontinuities and continuities are the most likely estimates given local 2D structures. Figure 5.7(b) shows the MLE from the *normalized* distributions: *i.e.*, each triangle in figure 5.6 is normalized within itself so that its maximum likelihood is 1. This way we can see the mostly likely *local 2D structures* for different local 3D structures.

- Figure 5.6(a) shows that homogeneous 2D structures are very likely to be formed by 3D continuities as the likelihood P(Continuity | 2D Structure) is very high (bigger than 0.85) for the area where homogeneous 2D structures exist (marked with H in figure 5.6(a)). This observation is confirmed in the MLE estimates of figure 5.7.

  Many surface reconstruction studies make use of a basic assumption that there is a smooth surface between any two points in the 3D world, if there is no contrast difference between these points in the image. This assumption has been first called as 'no news is good news' in [Grimson, 1983]. Figure 5.6(a) quantifies 'no news is good news' and shows for which structures and to what extent it holds: In addition to the fact that no news is in fact good news, figure 5.6(a) shows that news, especially texture-like structures and edge-like structures, can also be good news (see below).

Homogeneous image patches cannot be used for depth extraction by correspondence-based methods, and only weak or no information from these structures is processed by the cortex. Unfortunately, the vast majority of local image structure is of this type (see, *e.g.*, [Kalkan et al., 2005] and chapter 3). On the other hand, homogeneous patches indicate 'no change' in depth which is the underlying assumption of interpolation algorithms.

- Edges are considered as important sources of information for object recognition and reliable correspondence finding. Approximately 10% of local image structures are of that type (see, *e.g.*, [Kalkan et al., 2005] and chapter 3). Figures 5.6(a), (b) and (d) together with the MLE estimates in figure 5.7 show that most of the edges are very likely to be formed by continuous surfaces or gap discontinuities. Looking at the decision areas for different local 2D structures shown in figure 2.1(d), we see that the edges formed by continuous surfaces are mostly low-contrast edges (figure 5.6(a)); *i.e.*, the origin variance is close to 0.5. Little percentage of the edges are formed by orientation discontinuities (figure 5.6(d)).

- Figures 5.6(a) and (b) show that well-defined corner-like structures result from either gap discontinuities or continuities.

- Textures also map with high likelihood to surface continuities but also to irregular gap discontinuities.

  Finding correspondences becomes more difficult with the lack or repetitiveness of the local structure. The estimates of the correspondences at texture-like structures are naturally less reliable. In this sense, the likelihood that certain textures are caused by continuous surfaces (shown in figure 5.6(a)) can be used to model stereo matching functions that include interpolation as well as information about possible correspondences based on the local image information.

It is remarkable that local image structures mapping to different sub-regions in the triangle are caused by rather different 3D structures. This clearly indicates that these different image structures should be used in different ways for surface reconstruction.

5.5 Discussion

This chapter analyzed which local image structures suggest a depth interpolation process. Using natural images, it showed that homogeneous image structures correspond to continuous surfaces, as suggested and utilized by some computational theories of surface interpolation (see, *e.g.*, [Grimson, 1983]). On the other hand, a considerable proportion of edge-like structures lie on continuous surfaces (see figure 5.6(a)); *i.e.*, a contrast difference does not necessarily mean a depth discontinuity. This suggests that interpreting edges in combination with neighboring corners or edges is important for understanding the underlying 3D structure [Barrow and Tenenbaum, 1981].

The results from section 5.4 are useful in several contexts:

- Depth interpolation studies assume that homogeneous image regions are part of the same surface. Such studies can be extended with the statistics provided here as priors in a Bayesian framework. This extension would allow making use of the continuous surfaces that a contrast difference (caused by textures or edge-like structures) might correspond to.

  Acquiring range data from a scene is a time-consuming task compared to image acquisition, which lasts on the order of seconds even for high resolutions. In [Torres-Mendez and Dudek, 2006], for mobile robot environment modeling, instead of making a full-scan of the whole scene, only partial range scan is performed due to time constraints. This partial range data is completed by using a Markov Random Field which is trained from a pair of complete range and the corresponding image data. In [Torres-Mendez and Dudek, 2006], the partial range data is produced in a regular way; *i.e.*, every nth scan-column is neglected. This assumption, however, may introduce aliasing in the 3D data acquired from natural images using depth cues, and therefore, their method may not be applicable. Nevertheless, it could possibly be improved by utilizing the priors introduced in this chapter.

- Automated registration of range and color images of a scene is crucial for several purposes like extracting 3D models of real objects and scenes. Methods that align edges extracted from the intensity image with the range data already exist (see, *e.g.*, [Laycock and Day, 2006]). These methods can be extended with the results presented in this chapter in a way that not only edges but also other image structures are used for alignment. Such an extension also allows a probabilistic framework

by utilizing the probability P(3D Structure | 2D Structure). Moreover, making use of local 3D structure types that are introduced in this chapter can be more robust than just a gap discontinuity detection.

Such an extension is possible by maximizing the following energy function:

$$E(R,T) = \int_{u,v} P(\text{ 3D Structure at } (u,v) \mid \text{2D Structure at } (u,v))\, \mathrm{d}u\, \mathrm{d}v, \tag{5.8}$$

where R and T are translation and rotation of the range data in 3D space.

By extracting a more complex representation than existing range-data analysis studies, the current chapter could point to the intrinsic properties of the 3D world and its relation to the image data. This analysis is important because (1) it may be that the human visual system is adapted to the statistics of the environment [Brunswik and Kamiya, 1953, Knill and Richards, 1996, Krueger, 1998, Olshausen and Field, 1996, Purves and Lotto, 2002, Rao et al., 2002], and (2) it may be used in several computer vision applications (for example, depth estimation) in a similar way as in [Elder and Goldberg, 2002, Elder et al., 2003, Pugeault et al., 2004, Zhu, 1999].

5.5.1 Limitations of the current work

The first limitation is due to the type of scenes that have been used; *i.e.*, scenes of man-made environments which also included trees. Alternative scenes could include pure forest scenes or scenes taken from an environment with totally round objects. However, we believe that our dataset captures the general properties of the scenes that a human being encounters in daily life.

Different scenes might produce quantitatively different but qualitatively similar results. For example, forest scenes would produce much more irregular gap discontinuities than the current scenes; however, our conclusions regarding the link between textures and irregular gap discontinuities would still hold.

It should be noted that acquisition of range data with color images is very hard for forest scenes since the color image of the scene is taken after the scene is scanned with the scanner. During this period, the leaves and the trees may move (due to wind etc.), making the range and the color data inconsistent. In office environments, a similar problem arises: due to lateral separation between the digital camera and range scanner, there is the parallax problem, which again produces inconsistent range-color association.

For an office environment, a small-scale range scanner needs to be used.

The statistics presented in this chapter can be extended by analyzing forest scenes, office scenes etc. independently. The comparison of such independent analyses should provide more insights into the relations that this chapter have investigated but we believe that the qualitative conclusions of this chapter would still hold.

It would be interesting to see the results presented in the chapter by changing the measure for surface continuity so that it can separate planar and curved surfaces.

5.6 Acknowledgements

We would like to thank RIEGL U.K. Ltd. for providing us with chromatic 3D range data. The publications of the author which are relevant for this chapter are [Kalkan et al., 2006, Kalkan et al., 2007c].

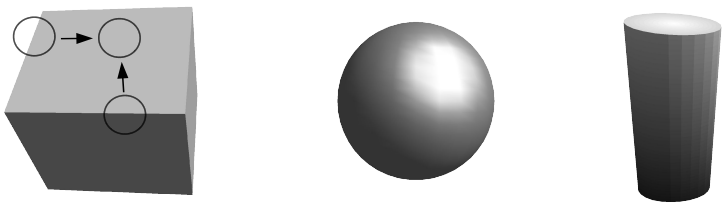
Chapter 6

Statistical Relation between Local 3D Structures

Chapter 5 investigated the relation between local 2D and 3D structures and suggested that different structures should be used in different ways for 3D depth extraction. Once depth is extracted from available features, however, it will be incomplete most of the time. One important reason for this is the correspondence problem of the multi-view depth cues as already discussed in chapter 1 as a problem of early vision.

The current chapter investigates *whether* the depth information available at edge-like structures can be used for *filling in* the missing depth information at homogeneous image areas (chapter 7 deals with the *how* part in an early cognitive vision framework).

Using the ground truth range data, this chapter investigates co-planarity relations between at homogeneous image structures and the edges that bound them. In other words, given two proximate co-planar edges, we compute the 'probability field' of finding co-planar surface patches which project as homogeneous image structures in the 2D image. This probability field is similar to the 'association field' [Field et al., 1993] which is a probability field also based on natural image statistics. The 'probability field', which is developed in this chapter, provides important information about (1) the predictability of depth at homogeneous image structures using the depth available at the bounding edges and (2) the relative complexity of 3D geometric structure compared to the complexity of 2D image structures.



(a) (b) (c)

Figure 6.1: Illustration of the relation between the depth of homogeneous image structures and the bounding edges. **(a)** In the case of cube, the depth of homogeneous image area and the bounding edges are related. However, in the case of round surfaces, **(b)** the depth of homogeneous image structures may not be related to the depth of the bounding edges since the depth information is sometimes given by other depth cues such as shading. **(c)** In the case of a cylinder, we see both cases of the relation as illustrated in (a) and (b).

Depth relation between edges and homogeneous image structures is illustrated for a few examples in figure 6.1.

The ground truth data is the chromatic range data that has also been used in chapter 5. A small subset of the data set is displayed in figure 5.2.

The following section provides the details of the analysis. The results are presented and discussed in section 6.2. See section 5.1 for relevant studies on the analysis of range data statistics.

6.1 Methods

This section provides the procedural details of how the analysis is performed.

The analysis is performed in three stages: First, local 2D and 3D representations of the scene are extracted from the chromatic range data. Second, a data set is constructed out of each pair of edge features, associating the monos that are likely to be coplanar to those edges to them (see section 6.1.2 for what is meant by relevance). Third, the coplanarity between the monos and the edge features that they are associated to are investigated. An overview of the analysis process is sketched in figure 6.2, which roughly lists the steps involved.

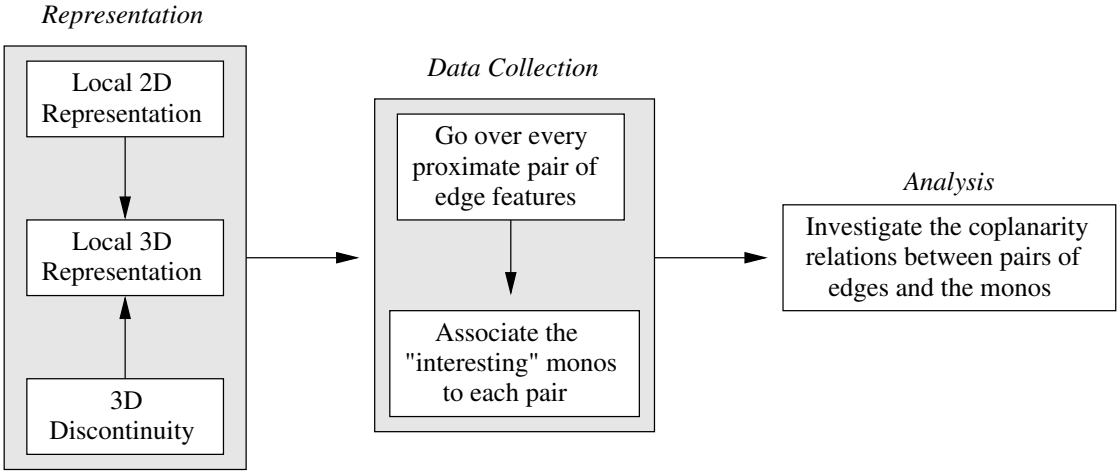


Figure 6.2: Overview of the analysis process. First, local 2D and 3D representations of the scene are extracted from the chromatic range data. Second, a data set is constructed out of each pair of edge features, associating the monos that are likely to be coplanar (*i.e.*, "interesting") to them (see section 6.1.2 for what is meant by relevance). Third, the coplanarity between the monos and the edge features that they are associated to are investigated.

6.1.1 Representation

Using the 2D image and the associated 3D range data, a representation of the scene is created in terms of local compository 2D and 3D features denoted by π. In this process, first, 2D features are extracted from the image information, and at the locations of these 2D features, 3D features are computed. The complementary information from the 2D and 3D features are then merged at each valid position, where validity is only defined by having enough range data to extract a 3D representation.

For homogeneous and edge-like structures, different representations are needed due to different underlying structures (in the rest of the chapter, a homogeneous image structure that corresponds to a 3D continuity will be called a *mono*). For this reason, there are two different definitions of π denoted respectively by π^e (for edge-like structures) and π^m (for monos) and formulated as:

$$\pi^m = (\mathbf{X}_{3D}, \mathbf{X}_{2D}, \mathbf{c}, \mathbf{p}), \tag{6.1}$$

$$\pi^e = (\mathbf{X}_{3D}, \mathbf{X}_{2D}, \phi_{2D}, \mathbf{c}_1, \mathbf{c}_2, \mathbf{p}_1, \mathbf{p}_2), \tag{6.2}$$

where $\mathbf{X}_{3D}$ and $\mathbf{X}_{2D}$ denote 3D and 2D positions of the 3D entity; ϕ_{2D} is the 2D orientation of the 3D entity; $\mathbf{c}_1$ and $\mathbf{c}_2$ are the 2D color representation of the surfaces of the 3D entity; $\mathbf{c}$ represents the color

of π^m; $\mathbf{p}_1$ and $\mathbf{p}_2$ are the planes that represent the surfaces that meet at the 3D entity; and $\mathbf{p}$ represents the plane of π^m (see figure 6.3). Note that π^m does not have any 2D orientation information (because it is undefined for homogeneous structures), and π^e has two color and plane representations to the 'left' and 'right' of the edge.

The process of creating the representation of a scene is illustrated in figure 6.3.

For the current analysis, the entities are regularly sampled from the 2D information. The sampling size is 10 pixels. See [Krüger et al., 2003, Krüger and Wörgötter, 2005] for details.

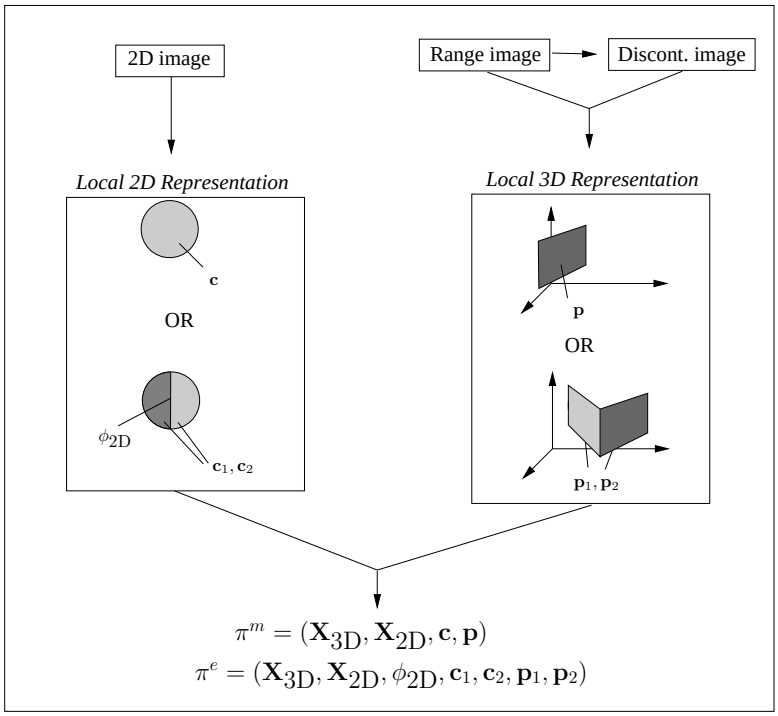


Figure 6.3: Illustration of the representation of a 3D entity. From the 2D and 3D information, local 2D and 3D representation is extracted.

Extraction of the planar representation requires knowledge about the type of local 3D structure of the 3D entity (see figure 6.3). Namely, if the 3D entity is a continuous surface, then only one plane needs to be extracted; if the 3D entity is an orientation discontinuity, then there will be two planes for extraction; if the 3D entity is a gap discontinuity, then there will also be two planes for extraction.

In the case of a continuous surface, a single plane is fitted to the set of 3D points in the 3D entity in question. For orientation discontinuous 3D structures, extraction of the planar representation is not

straight-forward. For these structures, our approach was to fit unit-planes[1] to the 3D points of the 3D entity and find the two clusters in these planes using k-means clustering of the 3D orientations of the small planes. Then, one plane is fitted for each of the two clusters, producing the bi-fold planar representation of the 3D entity.

Color representation is extracted in a similar way. If the image patch is a homogeneous structure, then the average color of the pixels in the patch is taken to be the color representation. If the image patch is edge-like, then it has two colors separated by the line which goes through the center of the image patch and which has the 2D orientation of the image patch. In this case, the averages of the colors of the different sides of the edge define the color representation in terms of c_1 and c_2. If the image patch is corner-like, the color representation becomes undefined.

6.1.2 Collecting the Data Set

In our analysis, we form pairs out of π^es that are close enough (see below), and for each pair, we check whether monos in the scene are coplanar to the elements of the pair or not. As there are plenty of monos in the scene, we only consider a subset of monos for each pair of π^e that are suspected to be relevant to the analysis because otherwise, the analysis becomes computationally intractable. The situation is illustrated in figure 6.4(a). In this figure, two π^e and three regions are shown; however, only one of these regions (*i.e.*, region A) is likely to have coplanar monos (*e.g.*, see figure 6.1(a)). This *assumption* is based on the observation of how objects are formed in the real world: objects have boundaries which consists of edge-like structures who bound surfaces, or image areas, of the object. The image area that is bounded by a pair of edge-like structures is likely to be the area that has the normals of both structures. For convex surfaces of the objects, the area that is bounded belongs to the object; however, in the case of concave surfaces, the area covered may also be from other objects, and the extent of the effect of this is part of the analysis.

Let $\mathcal{P}$ denote the set of pairs of proximate π^es whose normals intersect. $\mathcal{P}$ can be defined as:

$$\mathcal{P} = \left\{(\pi_1^e, \pi_2^e) \mid \forall \pi_1^e, \pi_2^e,\ \pi_1^e \in \Omega(\pi_2^e),\ I(\perp(\pi_1^e), \perp(\pi_2^e))\right\}, \tag{6.3}$$

where $\Omega(\pi^e)$ is the N-pixel-2D-neighborhood of π^e; $\perp(\pi^e)$ is the 2D line orthogonal to the 2D orientation

[1] Unit-planes mean planes that are fitted to the 3D points that are 1-pixel apart in the 2D image.

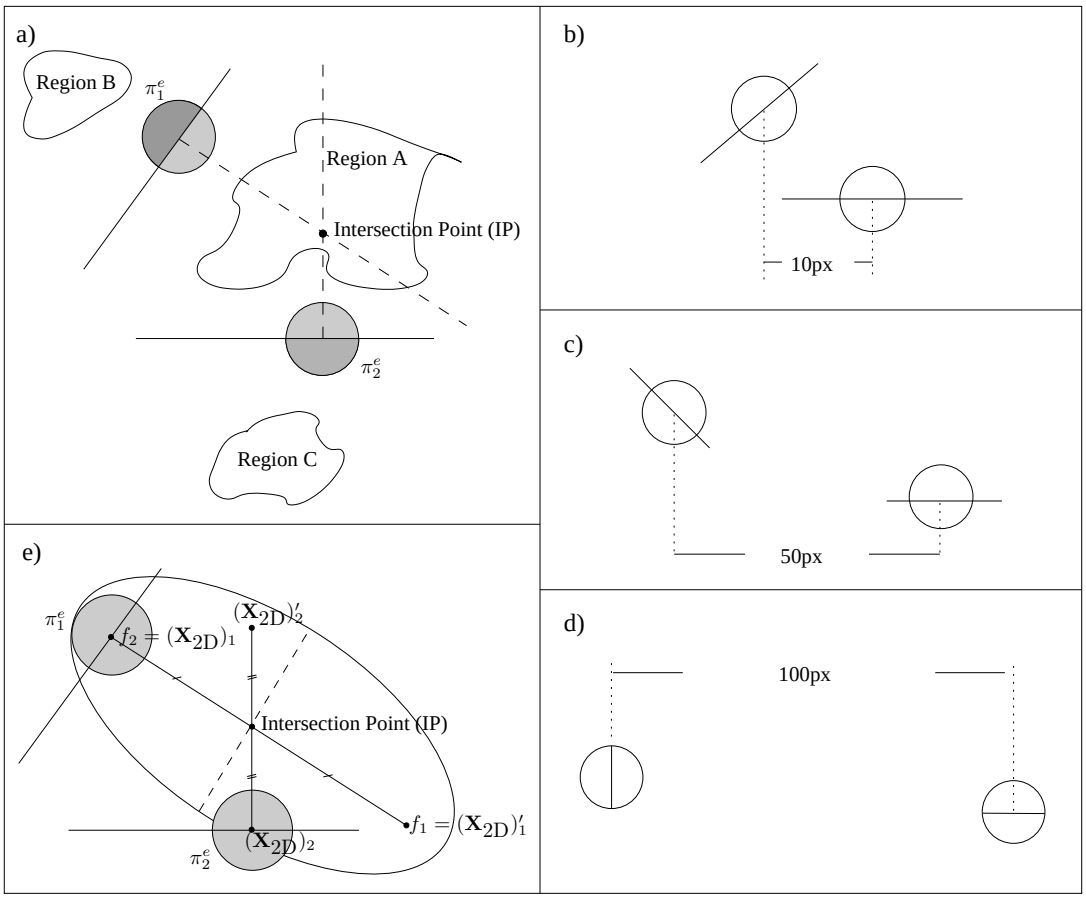


Figure 6.4: (a) Given a pair of edge features, coplanarity relation can be investigated for homogeneous image patches inside regions A, B and C. However, due to computational intractability reasons, this chapter is concerned in making the analysis only in region A (see the text for more details). (b)-(d) A few different configurations of edge features that might be encountered in the analysis. The difficult part of the investigation is to make these different configurations comparable, which can be achieved by fitting a shape (like square, rectangle, circle, parallelogram, ellipse) to these configurations. (e) The ellipse, among the alternative shapes (*i.e.*, square, rectangle, circle, parallellogram) turns out to describe the different configurations shown in (b)-(d) better. For this reason, ellipse is for analyzing coplanarity relations in the rest of the chapter.

of π^e, i.e., the normal of π^e; and, $I(l_1, l_2)$ is true if the lines l_1 and l_2 intersect. N is set to 100.

It turns out that there are a lot of different configurations possible for a pair of edge features based on relative position and orientation, which are illustrated for a few cases in figure 6.4(b)-(d). The difficult part of the investigation is to be able to compare these different configurations. One way to achieve this is to fit a shape to region A which can *normalize* the coplanarity relations by its size in order to make them comparable (see section 6.2 for more information).

The possible shapes would be square, rectangle, parallelogram, circle and ellipse. Among the alternatives, it turns out that an ellipse (1) is computationally cheap and (2) fits to different configurations of π_1 and π_2 under different orientations and distances *without* leaving region A much. Figure 6.4(e) demonstrates the ellipse generated by an example pair of edges in figure 6.4(a). The center of the ellipse is at the intersection of the normals of the edges, which is called *the intersection point* (IP) in the rest of the chapter.

The parameters of an ellipse are composed of two focus points f_1, f_2 and the minor axis b. In the current analysis, the more distant 3D edge feature determines the foci of the ellipse (and, hence, the major axis), and the other 3D edge feature determines the length of the minor axis. Alternatively, the ellipse can be constructed by minimizing an energy functional which optimizes the area of the ellipse inside region A and going through the features π_1 and π_2. However, for the sake of speed issues, the ellipse is constructed without optimization.

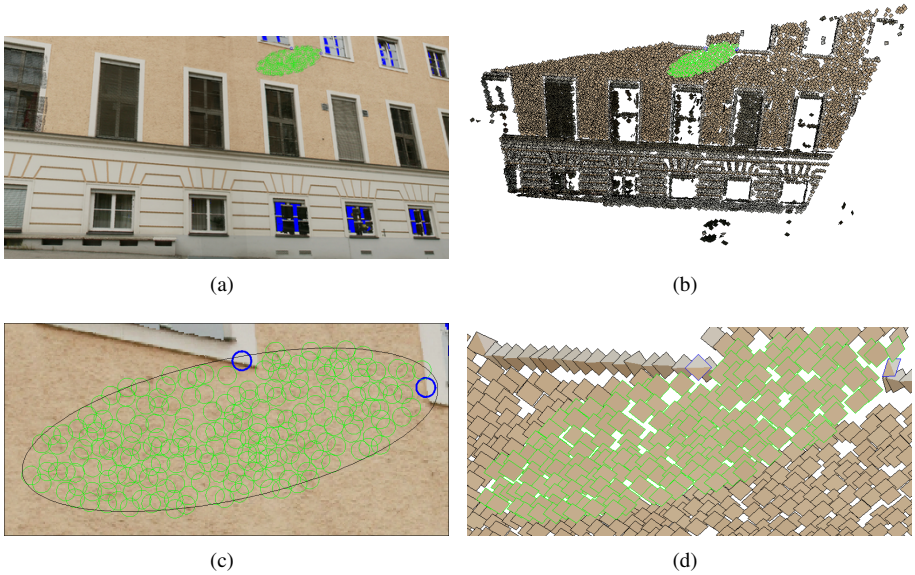
See appendix C.1 for details on how we determine the parameters of the ellipse.

For each pair of edges in $\mathcal{P}$, the region to analyze coplanarity is determined by intersecting the normals of the edges. Then, the monos inside the ellipse are associated to the pair of edges.

Note that a π^e has two planes that represent the underlying 3D structure. When π^es become associated to monos, only one plane, the one that points into the ellipse, remains relevant. Let π^{se} denote the semi-representation of π^e which can be defined as:

$$\pi^{se} = (\mathbf{X}_{3D}, \mathbf{X}_{2D}, \mathbf{c}, \mathbf{p}). \tag{6.4}$$

Note that π^{se} is equivalent to the definition of π^m in equation 6.2.



(a) (b)

(c) (d)

Figure 6.5: Illustration of a pair of π^e and the set of monos associated to them. **(a)** The input scene. A pair of edges (marked in blue) and the associated monos (marked in green) with an ellipse (drawn in black) around them shown on the input image. See (c) for a zoomed view. **(b)** The 3D representation of the scene in our 3D visualization software. This representation is created from the range data corresponding to (a) and is explained in the text. **(c)** The part of the input image from (a) where the edges, the monos and the ellipse are better visible. **(d)** A part of the 3D representation (from (b)) corresponding to the pair of edges and the monos in (c) is displayed in detail where the edges are shown with blue margins; the monos with the edges are shown in green (all monos are coplanar with the edges). The 3D entities are drawn in rectangles because of the high computational complexity for drawing circles.

Let $\mathcal{T}$ denote the data set which stores $\mathcal{P}$ and the associated monos which can be formulated as:

$$\mathcal{T} = \{(\pi_1^{se}, \pi_2^{se}, \pi^m) \mid (\pi_1^e, \pi_2^e) \in \mathcal{P}, \pi^m \in \mathcal{S}^m, \pi^m \in E(\pi_1^e, \pi_2^e)\}, \tag{6.5}$$

where $\mathcal{S}^m$ is the set of all π^m.

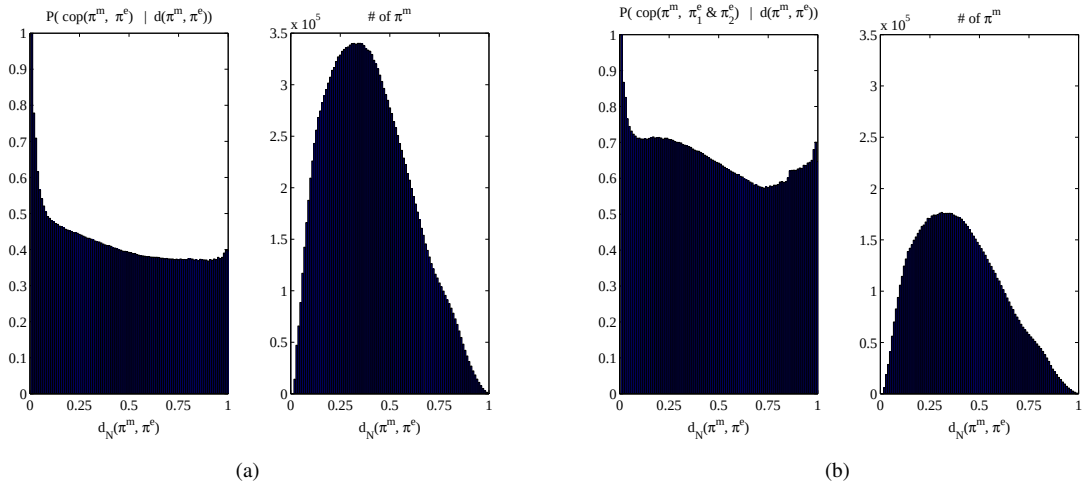
A pair of π^es and the set of monos associated to them are illustrated in figure 6.5. The edges are shown in blue, and the coplanar and non-coplanar monos are shown in green and red, respectively.

6.1.3 Definition of coplanarity

Two entities are coplanar if they are on the same plane. Coplanarity of edge features and monos is equivalent to coplanarity of two planar patches: two planar patches A and B are coplanar if (1) they are parallel and (2) the planar distance between them is zero.

See appendix C.2 for more information.

6.2 Results



(a) (b)

Figure 6.6: Likelihood distribution of coplanarity of monos. In each sub-figure, left-plot shows the likelihood distribution whereas right-plot shows the frequency distribution. (a) The likelihood of the coplanarity of a mono with π^e_1 *or* π^e_2 against the distance to π^e_1 *or* π^e_2. This is the unconstrained case; *i.e.*, the case where there is no information about the coplanarity of π^e_1 and π^e_2. (b) The likelihood of the coplanarity of a mono with π^e_1 *and* π^e_2 against the distance to π^e_1 *or* π^e_2.

The data set $\mathcal{T}$ defined in equation 6.5 consists of pairs of π^e_1, π^e_2 and the associated monos. Using this set, we compute the likelihood that a mono is coplanar with $\pi^e_1$and/or π^e_2 against a distance measure.

The results of the current analysis are shown in figures 6.6 and 6.8 and 6.9.

In figure 6.6(b), the likelihood of the coplanarity of a mono against the distance to π^e_1 *or* π^e_2 is shown. This likelihood can be denoted formally as $P(cop(\pi^m, \pi^e_1 \ \& \ \pi^e_2) \mid d_N(\pi^m, \pi^e))$ where $cop(\pi^m, \pi^e_1 \ \& \ \pi^e_2)$ is defined as $cop(\pi^e_1, \pi^e_2) \ \wedge \ cop(\pi^m, \pi^e)$, and π^e is either π^e_1 or π^e_2. The normalized distance measure[2] $d_N(\pi^m, \pi^e)$ is defined as:

$$d_N(\pi^m, \pi^e) = \frac{d(\pi^m, \pi^e)}{2\sqrt{d(\pi^e_1, IP)^2 + d(\pi^e_2, IP)^2}}, \tag{6.6}$$

where π^e is either π^e_1 or π^e_2, and IP is the intersection point of π^e_1 and π^e_2. We see in figure 6.6(b) that the likelihood decreases when a mono is more distant from an edge. However, when the distance measure

[2]In the following plots, the distance means the Euclidean distance in the image domain.

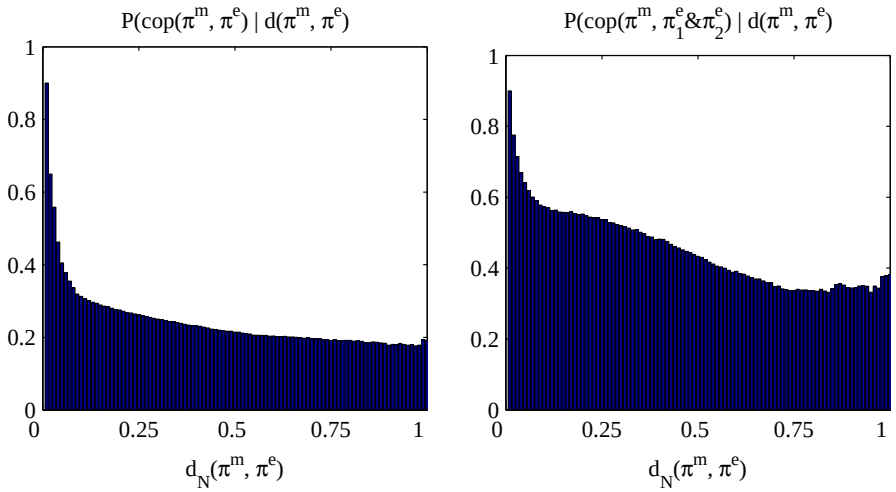


Figure 6.7: Likelihoods from figures 6.6(a) and 6.6(b) with a more *strict* coplanarity relation (namely, we set the thresholds T_p and T_d to 10 degrees and 0.2, respectively. See Appendix for more information about these thresholds). **(a)** Figure 6.6(a) with more strict coplanarity relation. **(b)** Figure 6.6(b) with more strict coplanarity relation.

gets closer to one, the likelihood increases again. This is because, when a mono gets away from either π^e_1 or π^e_2, it gets closer to the other π^e.

In figure 6.6(a), we see the unconstrained case of figure 6.6(b); *i.e.*, the case where there is no information about the coplanarity of π^e_1 and π^e_2; namely, the probability $P(cop(\pi^m, \pi^e) \mid d_N(\pi^m, \pi^e))$ where π^e is either π^e_1 or π^e_2. The comparison with figure 6.6(b) shows that the existence of another edge in the neighborhood increases the likelihood of finding coplanar structures. As there is no other coplanar edge in the neighborhood, the probability does not increase when the distance is close to one (compare with figure 6.6(b)).

It is intuitive to expect symmetries in figure 6.6. However, as (1) the roles of π^e_1 and π^e_2 in the ellipse are fixed, and (2) one π^e is guaranteed to be on the major axis, and the other π^e may or may not be on the minor axis, the symmetry is not observable in figure 6.6.

To see the effect of the coplanarity relation on the results, we reproduced figures 6.6(a) and 6.6(b) with a more *strict* coplanarity relation (namely, we set the thresholds T_p and T_d to 10 degrees and 0.2, respectively. See Appendix C.2 for more information about these thresholds). The results with more constrained coplanarity relation are shown in figure 6.7. Although the likelihood changes quantitatively, the

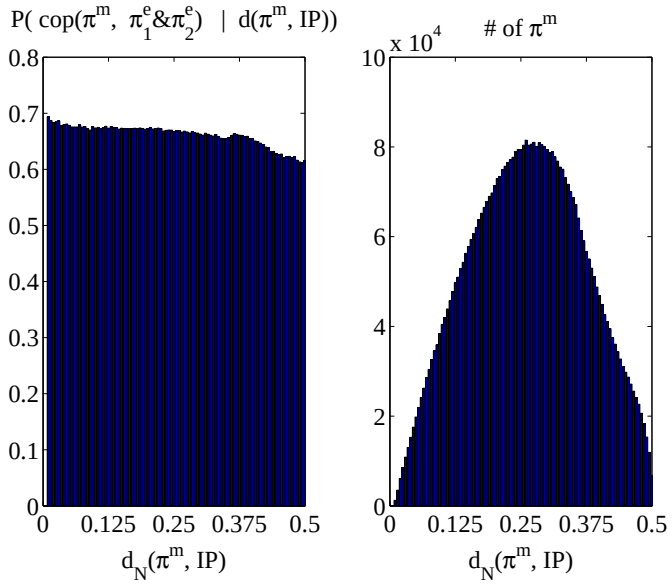


Figure 6.8: The likelihood of the coplanarity of a mono against the distance to IP. Left-plot shows the likelihood distribution whereas right-plot shows the frequency distribution.

figure shows the qualitative behaviours that have been observed with the standard thresholds. Moreover, we cross-checked the results for subsets of the original dataset (results not provided here) and confirmed the same qualitative results.

In figure 6.8, the likelihood of the coplanarity of a mono against the distance to IP (*i.e.*, $P(cop(\pi^m, \pi_1^e \,\&\, \pi_2^e) \mid d_N(\pi^m, IP))$) is shown. We see in the figure that the likelihood shows a flat distribution against the distance to IP.

In figure 6.9, the likelihood of the coplanarity of a mono against the distance to π_1^e *and* π_2^e (*i.e.*, $P(cop(\pi^m, \pi_1^e \,\&\, \pi_2^e) \mid d_N(\pi^m, \pi_1^e), d_N(\pi^m, \pi_2^e))$) is shown. We see that when π^m is close to π_1^e or π_2^e, it is more likely to be coplanar with π_1^e and π_2^e than when it is equidistant to both edges. The reason is that, when π^m moves away from an equidistant point, it becomes closer to the other edge, in which case the likelihood increases as shown in figure 6.6(b).

The results, especially figures 6.6(b) and 6.6(a) confirm the importance of the relation illustrated in figure 6.1(a).

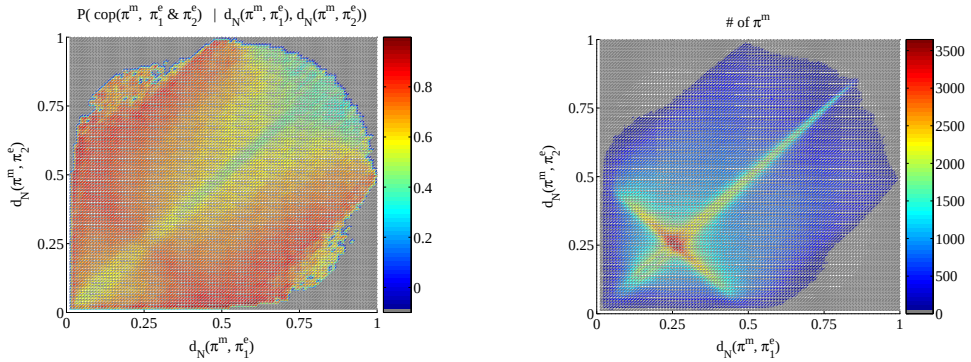


Figure 6.9: The likelihood of the coplanarity of a mono against the distance to π_1^e *and* π_2^e. Left-plot shows the likelihood distribution whereas right-plot shows the frequency distribution.

6.3 Discussion

This chapter analyzed whether depth at homogeneous image structures is related to the depth of edge-like structures in the neighborhood. Such an analysis is important for understanding the possible mechanisms that could underlie depth interpolation processes. Our findings show that an edge feature provides significant evidence for making a depth prediction at a homogeneous image patch that is in the neighborhood. Moreover, the existence of a second edge feature in its neighborhood which is not collinear with the first edge feature increases the likelihood of the prediction.

Using second order relations and higher order features for representing the 2D image and 3D range data, we produce confirming results that the natural scene geometry is simpler compared to 2D images (see, [Yang and Purves, 2003]).

By extracting a more complex representation than existing range-data analysis studies, we could point to the intrinsic properties of the 3D world and its relation to the image data. This analysis is important because (1) it may be that the human visual system is adapted to the statistics of the environment [Brunswik and Kamiya, 1953, Knill and Richards, 1996, Krueger, 1998, Olshausen and Field, 1996, Purves and Lotto, 2002, Rao et al., 2002], and (2) it may be used in several computer vision applications (for example, depth estimation) in a similar way as in [Elder and Goldberg, 2002, Elder et al., 2003, Pugeault et al., 2004, Zhu, 1999].

6.3.1 Limitations of the current work

An important limitation has already been mentioned in section 5.5.1, regarding the type of scenes that have been used. It was said that alternative scenes like pure forest scenes or scenes taken from an environment with totally round objects. However, we believe that our dataset captures the general properties of the scenes that a human being encounters in daily life.

Different scenes might produce quantitatively different but qualitatively similar results. For example, coplanarity relations would be harder to predict for forest scenes since (depending on the scale) surface continuities are harder to find; however, on a bigger scale, some forest scenes are likely to produce the same qualitative results presented in this chapter because of piecewise planar leaves which are separated by gap discontinuities.

It should be noted that acquisition of range data with color images is very hard for forest scenes since the color image of the scene is taken after the scene is scanned with the scanner. During this period, the leaves and the trees may move (due to wind etc.), making the range and the color data inconsistent. In office environments, a similar problem arises: due to lateral separation between the digital camera and range scanner, there is the parallax problem, which again produces inconsistent range-color association. For an office environment, a small-scale range scanner needs to be used.

The statistics presented in this chapter can be extended by analyzing forest scenes, office scenes etc. independently. The comparison of such independent analyses should provide more insights into the relations that this chapter have investigated but we believe that the qualitative conclusions of this chapter would still hold.

6.4 Acknowledgements

We would like to thank RIEGL U.K. Ltd. for providing us with chromatic 3D range data. The publications of the author which are relevant for this chapter are [Kalkan et al., 2007d, Kalkan et al., 2007c].

Chapter 7

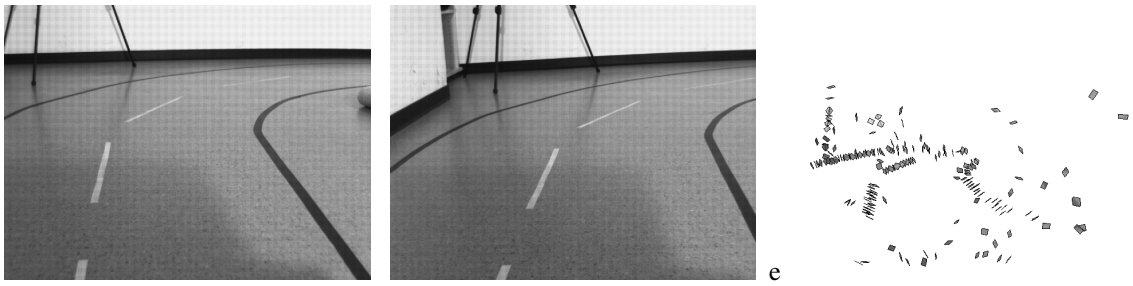
A Model for Depth Prediction from 3D Edge Features

Chapter 6 suggested that a 3D edge feature with a surface representation is able to predict the depth of a homogeneous image patch. It quantified that the strength of this prediction decreases with the distance between the homogeneous image patch and the edge feature, and that the existence of a second 3D edge feature in the neighborhood of a homogeneous image patch increases the likelihood of this prediction.

Motivated by these results in chapter 6, the current chapter is interested in the prediction of depth at homogeneous image patches (called *mono*s; see section 2.2) from the depth of the edges[1] in the scene using a voting model. The model starts by creating a representation of the input stereo image pair in terms of local features corresponding to edge-like structures and monos (as introduced in [Krüger et al., 2004b] and in section 2.2). The depth at edge-like features is extracted using a feature-based stereo method introduced in [Pugeault and Krüger, 2003]. This provides a 3D-silhouette of the scene which however can include strong outliers and ambiguous interpretations in particular when large disparities and low thresholds on matching similarities are used (figure 7.1). The depth of a certain mono, then, is voted by the 3D edge-like features that are part of this 3D-silhouette.

A typical scenario with extracted 3D information (using stereo) is shown in figure 7.1. We see that

[1] Note that the 3D edges in chapter 6 are extracted from range data, and therefore, includes surface information on two sides of an edge. The 3D edge features in this chapter, however, are extracted from stereo, and includes only 3D line orientation (see section 2.2).



(a) (b) (c)

Figure 7.1: (a-b) An input stereo pair. (c) Results of a feature-based stereo algorithm taken from [Pugeault and Krüger, 2003]. This is a view from our 3D displaying software which shows the 3D edge features as rectangles for the sake of simplicity. Note that stereo information can contain outliers.

stereo computation may produce strong outliers which prohibit a direct application of a *surface interpolation* process as it is not trivial to differentiate between the outliers and the reliable stereo information. Moreover, the 3D features at the edges of the road that should be reliable turn out not to share a common surface nor a common 3D line (see figure 7.1(c)). Therefore, applying a surface interpolation method on such input data is expected to lead to an erroneous interpretation of the scene. In this chapter, we will show that our depth prediction method is able to cope with these situations.

We compare our depth prediction method with several dense stereo methods (with local as well as global optimizations) on real and artificial scenes where the amount of texture can be controlled to see the effect of texture on the performance of the different approaches. We show that dense stereo methods are best suited for textured image areas whereas our method performs well on homogeneous or weakly-textured image areas or edges. The results suggest a combination of the two different approaches into a single model that can perform well at both textured and homogeneous or weakly-textured image areas.

The contributions of this chapter can be listed as:

- A novel voting-based method for predicting depth at homogeneous image areas using just the 3D line orientation at 3D local edge-features.
- Our votes have reliability measures which are based on the co-planarity statistics of 3D local surface patches provided in [Kalkan et al., 2007e] and chapter 6.
- Comparison with dense stereo on real and artificial scenes where we control the amount and the type of texture to see the effect on performance of the different approaches.

Depth prediction should be regarded as one depth cue which utilizes the 3D information at the edges. The utilization of the edges for depth prediction is along the lines of 3D surface interpretation from line drawings of objects [Barrow and Tenenbaum, 1981]. As motivated in chapter 1, depth prediction can be understood as a feedback mechanism which completes the missing information in early vision. This makes depth prediction a part of an early cognitive vision framework where different cues interact with each other to remove the ambiguities and the missing information in early vision.

7.1 Cues for depth extraction

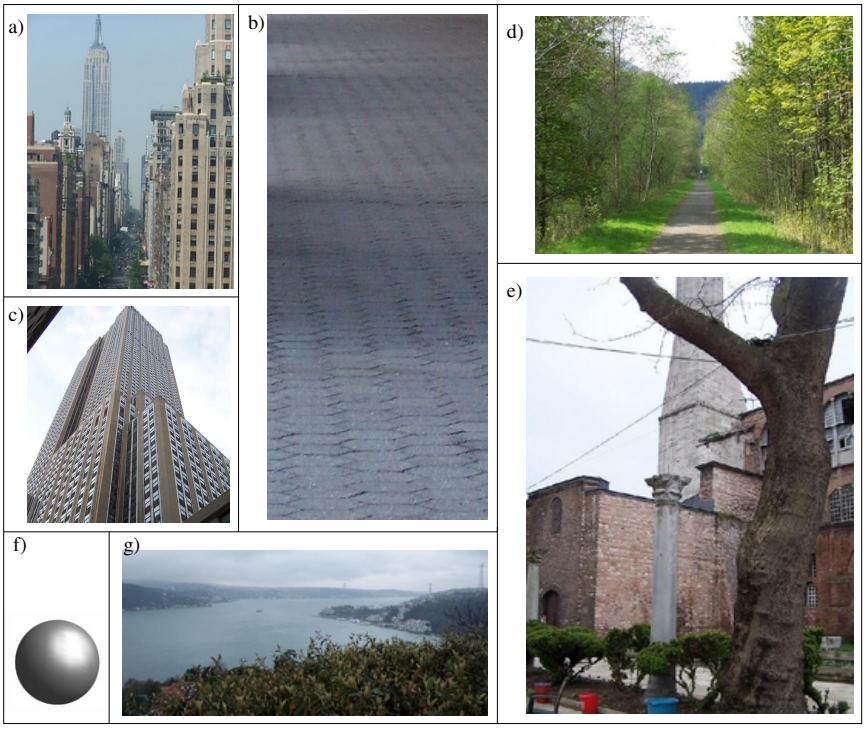


Figure 7.2: Pictorial, *i.e.*, monocular cues for depth extraction. Perspective (a) and texture-gradient (b) are important cues since (1) due to perspective projection, farther objects appear smaller in an image, and (2) most surfaces have a texture whose gradient changes with the surface normal and the distance to the camera. In (c), we see the cases of both a perspective cue and the texture gradient. In most cases, monocular cues are not able to provide exact depth, or surface, information like in (d), where we see texture-gradient (due to leaves of the trees) and perspective cue. Occlusion (e) is another widely observed cue; occlusion provides a depth cue by making statements about the ordering of the objects, *i.e.*, the occluding object is in front of the occluded object. The shading on a surface can be used also as a depth cue (f), which is called shape from shading in the literature. Another example for monocular cue is atmospheric effect (g), where the far away objects and surfaces look less sharp and more blurred.

Figure 7.3: Line drawing of a scene. Picture courtesy of [van Diepen and Graef, 1994].

In this section, we provide an overview of the different cues for depth extraction and briefly elaborate on the problems associated with some of these cues. It can be argued that one important task of vision is derive the 3D interpretation of a scene from its image projections. For this, two main types of depth cues are used: monocular, or pictorial, cues where the 3D interpretation is made using only one view of the scene; and multi-view, or correspondence-based, cues where at least two views of the scene are used for the 3D interpretation.

Examples of monocular cues include perspective distortion, texture-gradient, occlusion, shading and atmospheric effect (figure 7.2). Perspective (figure 7.2(a)) and texture-gradient (figure 7.2(b)) are important cues since (1) due to perspective projection, farther objects appear smaller in an image, and (2) most surfaces have a texture whose gradient changes with the surface normal and the distance to the camera. In (figure 7.2(c)), we see the cases of both a perspective cue and the texture gradient. In most cases, monocular cues are not able to provide exact depth, or surface, information like in (figure 7.2(d)), where we see texture-gradient (due to leaves of the trees) and perspective cue. Occlusion (figure 7.2(e)) is another widely observed cue; occlusion provides a depth cue by making statements about the ordering of the objects, *i.e.*, the occluding object is in front of the occluded object. The shading on a surface can be used also as a depth cue (figure 7.2(f)), which is called shape from shading in the literature [Ragheb and Hancock, 2002]. Another example for a monocular cue is the atmospheric effect (figure 7.2(g)), where the far away objects and surfaces look less sharp and more blurred. Psychophysical experiments have demonstrated that the 3D information about surfaces can also be recovered from the line drawings of an object (see, *e.g.*, [Barrow and Tenenbaum, 1981] and figure 7.3). Monocular cues can most of the time provide only relative depth information, or depth ordering between different surfaces or objects unless prior information about the scene or the objects

are used. Computational modelling of monocular cues has been limited since sophisticated interactions of the different cues are difficult to formulate; for example, from extensive research on shape from shading [Ragheb and Hancock, 2002, Robles-Kelly and Hancock, 2004], it turned out that these methods only work under very controlled illumination conditions. Similar observations can be made for texture-gradient [Clerc and Mallat, 2002].

Multi-view cues such as motion and stereo, on the other hand, can yield absolute depth values without prior information about the scene or the objects. Such cues use the projections of a 3D point in the different views to reconstruct the depth information by using simple trigonometric relations [Faugeras, 1993, Hartley and Zisserman, 2000]. For this, the corresponding projections of a 3D point from the different views need to be found. Usually called the *correspondence problem* in the literature, this search is not trivial since for each image point, or a feature, in a first frame, the matching point, or feature, in a second view is sought. This requires that a *globally-distinguishable* local structure exists at each image point, which does not always hold in natural scenes. For example, homogeneous or weakly-textured image areas are hard to match since they do not have distinguishable structures. Repetitive textures or patterns also face difficulties even though they carry image structure. In the case of a stereo setup, this search problem is simplified due to the geometry of the cameras, *i.e.*, the epipolar constraint, which states that the matching point of an image point in the first view can only be on a line (called the epipolar line), which is defined by the geometry of the cameras [Faugeras, 1993, Hartley and Zisserman, 2000].

There are two main computational approaches for stereo computation: dense and sparse methods. The dense methods tackle the correspondence problem at the signal level and try to compute stereo information for every pixel. They are complete, computationally expensive and are limited to small images, small disparities and baselines. The sparse methods, on the other hand, make use of image features rather than pixels for finding the correspondences, and therefore, produce only sparse depth information. Due to sparsity, they are computationally cheap, and they can work with big images, big disparities and big baselines. Dense methods require textured surfaces, where the textures are not repetitive (unless there is a global optimization step; see section 7.5) whereas sparse methods are applicable for scenes where the utilized features are meaningful.

Attention and the utilization of existing information about objects and scenes are important mechanisms in the human visual system for interpreting the 3D information of a scene. Object knowledge, for example, is argued to increase the accuracy and the speed of performing several tasks that require visual

perception [Bruce et al., 2003]. Such high-level information are likely to be exploited by using feedback to the lower visual processing levels.

As mentioned in chapter 1, the experiments suggest that depth cues which are not directly based on correspondences evolve rather late in the development of the human visual system [Kellman and Arterberry, 1998]. This indicates that experience may play an important role in the development of these cues, *i.e.*, that we have to understand depth perception as a statistical learning problem [Knill and Richards, 1996, Purves and Lotto, 2002, Rao et al., 2002].

An interesting question is about the reason of the existence of numerous depth cues. This is likely to be due to the redundancy of information in natural scenes. However, in general, a single cue is not enough for a full 3D reconstruction of a scene and more than one cue needs to be fused. For example, accuracy of stereo drops quickly with the distance from the camera (Grimson [Grimson, 1993] also questioned the extent of computational stereo vision as a depth cue). Moreover, there exist human beings without stereo vision, who are able to perceive the 3D world.

The current chapter introduces a depth prediction method which utilizes the 3D information at the edges in order to recover the missing depth information at weakly-textured image areas. For this, it uses a feature-based (*i.e.*, sparse) stereo algorithm; however, the stereo algorithm can be replaced by any other depth cue which produces the required 3D information at the edges of an image. Completion of the missing depth information in early vision using interactions between 3D edge features makes the depth prediction method a part of early cognitive vision, *i.e.*, at a higher processing stage than stereo. The proposed method can be considered as one novel depth cue that, of course, needs to interact with other cues.

7.2 Related studies

The work of Grimson [Grimson, 1982] can be regarded as the pioneer of surface interpolation studies. In [Grimson, 1982], Grimson proposed fitting square Laplacian functionals to surface orientations at existing 3D points utilizing a *surface consistency constraint* called 'no news is good news'. The constraint argues that if two image points do not have a contrast difference in-between, then they can be assumed to be on the same 3D surface (see [Kalkan et al., 2006] and chapter 5 for a quantification of this assumption). The work of [Grimson, 1982] is extended in [Grimson, 1984] with shape from shading. In

[Grimson, 1982], it is assumed that 3D orientation is available, and the input 3D points are dense enough for second order differentiation.

Interpretation of line drawings for recovering surface information is of relevance to our work. In [Barrow and Tenenbaum, 1981], lines are classified as extremal (where a surface turns away from the camera smoothly like at the edges of a sphere) or discontinuity (where a smooth surface terminates or intersects with another) by making use of the junction labels and global relations like symmetry and parallelism. They assume that (1) extremal points (the boundaries of the objects) in an image correspond to surface orientations which are normal to the image curve and the line of sight, and that (2) discontinuities (lines other than extremal points) lead to a possible set of surface orientations which are normal to 3D edge at the discontinuity point. The underlying assumptions of [Barrow and Tenenbaum, 1981] are that (1) a clean contour of the scene is provided, and that (2) the object is separated from the background. Moreover, the results provided in [Kalkan et al., 2006] (and in chapter 5) suggest that it may not be a good idea to assume that edges correspond to only one type of surface orientation. Other methods that are based on line drawing interpretations (*e.g.*, [Nalwa, 1989, Stevens, 1981, Ulupinar and Nevatia, 1991, Ulupinar and Nevatia, 1993]) are similar to [Barrow and Tenenbaum, 1981].

In [Guy and Medioni, 1994], 3D points with surface orientation are interpolated using a perceptual constraint called *co-surfacity* which produces a 3D association field (which is called the Diabolo field by the authors) similar to the association field used in 2D perceptual contour grouping studies. The association field casts votes around existing 3D points, and these votes are combined. If the points do not have 3D orientation, they estimate the 3D orientation first (by fitting a surface model locally) and then apply the surface interpolation step. Our work is different from [Guy and Medioni, 1994] in that they used the association field for filling in the missing depth information *only around* an existing 3D point whereas in our work, we allow long range interactions between edge descriptors, making use of only the 3D line orientation at the edges. Moreover, we utilize the votes of a pair of 3D edge descriptors rather than individual votes of 3D points.

Two other relevant studies are [Hoff and Ahuja, 1989, Lee et al., 2002]. They both argued that stereo matching and surface interpolation should not be sequential but rather simultaneous. [Hoff and Ahuja, 1989] fits local planes to disparity estimates from the zero-crossings of a stereo pair to make rough surface estimates which are then interpolated taking the occlusions into account. The disadvantages of [Hoff and Ahuja, 1989] are that their model fits a local plane to the disparity of a set of zero-crossings,

which is not accurate since the disparity at zero-crossings can reconstruct a 3D surface normal only up to a certain accuracy [Faugeras, 1993]. On the other hand, we are concerned with predictions (1) of 3D line orientations of the edges, (2) using long-range relations and (3) voting mechanisms. Moreover, as Hoff and Ahuja have tested their approach only on very textured scenes, the applicability of the approach to homogeneous image areas is not clear. [Lee et al., 2002] employs the following steps: A *dense* disparity map is computed, and the disparities corresponding to inliers, surfaces and surface discontinuities are marked and combined using tensor voting. The surfaces are then extracted from the dense disparities using marching cubes approach.

Our method is related to shape from silhouette methods which try to estimate the 3D information from the occluding edges of a single object (see, *e.g.*, [Kang et al., 2001, Liu et al., 2007]). As put forward in [Liu et al., 2007], these methods are limited to objects which are generated by a full rotation of a 1D curved structure around an axis, and the underlying principles are valid only for occluding edges. Such methods are usually combined with stereo in order to have a more efficient and a better performance [Matsumoto et al., 1999, Esteban and Schmitt, 2004].

In [Terzopoulos, 1982, Terzopoulos, 1988], stereo is computed at different scales, and instead of collapsing the results of these different scales into a single layer of disparity estimation and then applying surface interpolation, surface interpolation is applied separately for each scale and the results are combined.

Our work is different from the above mentioned works since it does not assume that the input stereo points are dense enough to compute their 3D orientation. Instead, our method relies on the 3D line-orientations of the edge segments which are extracted using a feature-based stereo algorithm (proposed in [Pugeault and Krüger, 2003]). The second difference is that we employ a voting method which is different from tensor-voting ([Lee and Medioni, 1998, Lee et al., 2002]) in that it allows long-range interactions in empty image areas and only in certain directions in much less computations than tensor-voting, in order to predict *both* the depth and the surface orientation.

We would like to distinguish *depth prediction* from *surface interpolation* because surface interpolation assumes that there is already a dense depth map of the scene available in order to estimate the 3D orientation at points (see, *e.g.*, [Grimson, 1982, Guy and Medioni, 1994, Lee and Medioni, 1998, Lee et al., 2002, Terzopoulos, 1988]) whereas our understanding of depth prediction makes use of only 3D line-orientations at edge-segments which are computed using a feature-based stereo proposed in

[Pugeault and Krüger, 2003].

7.3 Relations between Primitives

The sparse and the symbolic nature of primitives allows the following relations to be defined on them. These relations are used in deciding which features are used to make a depth prediction. See [Kalkan et al., 2007b] for more information about these relations.

[Pugeault et al., 2008] showed that the uncertainty of the 3D position of points, which are reconstructed using stereo, increases with the distance from the camera. A similar case occurs for the reconstruction of 3D orientation: line orientations parallel to the epipolar line have big uncertainties [Pugeault et al., 2008]. These results suggest that the relations defined in this section are suitable for close objects and non-epipolar edges[2] in a scene. This is in turn a limitation of the depth prediction method proposed in this chapter; *i.e.*, the depth prediction method is not suitable for distant surfaces or surfaces which are defined only by epipolar edges.

7.3.1 Co–planarity

Two 3D edge primitives $\mathbf{\Pi}_i^e$ and $\mathbf{\Pi}_j^e$ are defined to be co–planar if their orientation vectors t_i and t_j lie on the same plane, *i.e.*:

$$cop(\mathbf{\Pi}_i^e, \mathbf{\Pi}_j^e) = 1 - |\mathbf{proj}_{t_j \times v_{ij}}(t_i \times v_{ij})|, \tag{7.1}$$

where v_{ij} is the vector $(\boldsymbol{X}_i - \boldsymbol{X}_j)$; Θ_i and Θ_j are the 3D orientations; and, $\mathbf{proj_u}(\mathbf{a})$ is the projection of vector $\mathbf{a}$ over vector $\mathbf{u}$ and defined as:

$$\mathbf{proj_u}(\mathbf{a}) = \frac{\mathbf{a} \cdot \mathbf{u}}{\| \mathbf{u} \|^2}\mathbf{u}. \tag{7.2}$$

Among the perceptual relations used in this thesis, the co-planarity relation is affected by the uncertainty of reconstruction the most since it uses both the 3D position and the 3D line orientation. The co–planarity relation is illustrated in figure 7.4.

[2]Epipolar edges are those who are parallel to the epipolar line.

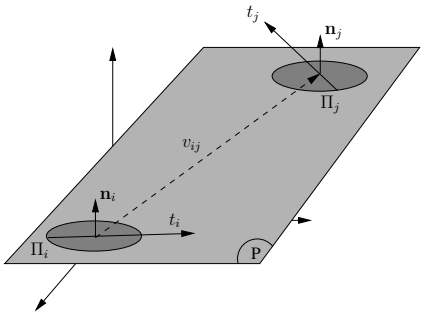


Figure 7.4: Co–planarity of two 3D primitives $\mathbf{\Pi}_i^e$ and $\mathbf{\Pi}_j^e$.

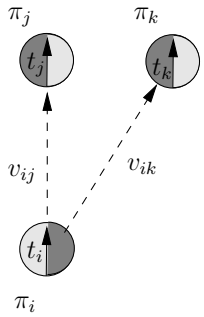


Figure 7.5: Linear dependence of three π_i^e, π_j^e and π_k^e. In this example, π_i^e is linearly dependent with π_j^e whereas π_k^e is linearly independent of other primitives.

7.3.2 Linear dependence

Two 3D primitives $\mathbf{\Pi}_i^e$ and $\mathbf{\Pi}_j^e$ are defined to be linearly dependent if the *three* lines which are defined by (1) the 3D orientation of $\mathbf{\Pi}_i^e$, (2) the 3D orientation of Π_j^e and (3) v_{ij} are identical. Due to uncertainty in the 3D reconstruction process, in this work, the linear dependence of two spatial primitives $\mathbf{\Pi}_i^e$ and $\mathbf{\Pi}_j^e$ is computed using their 2D projections π_i^e and π_j^e. We define the linear dependence of two 2D primitives π_i^e and π_j^e as:

$$lin(\pi_i^e, \pi_j^e) = |\mathbf{proj}_{v_{ij}} t_i| \times |\mathbf{proj}_{v_{ij}} t_j|, \tag{7.3}$$

where t_i and t_j are the vectors defined by the orientations θ_i and θ_j. Linear dependence is illustrated in figure 7.5.

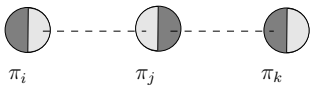


Figure 7.6: Co–colority of three 2D primitives π_i^e, π_j^e and π_k. In this example, π_i^e and π_j^e are co-color, so are π_i^e and π_k^e; however, π_j^e and π_k^e are not co-color.

7.3.3 Co–colority

Two 3D primitives $\mathbf{\Pi}_i^e$ and $\mathbf{\Pi}_j^e$ are defined to be co–color if their parts that face each other have the same color. In the same way as linear dependence, co–colority of two spatial primitives $\mathbf{\Pi}_i^e$ and $\mathbf{\Pi}_j^e$ is computed using their 2D projections π_i^e and π_j^e. We define the co–colority of two 2D primitives π_i^e and π_j^e as:

$$coc(\pi_i^e, \pi_j^e) = 1 - \mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j), \tag{7.4}$$

where $\mathbf{c}_i$ and $\mathbf{c}_j$ are the RGB representation of the colors of the parts of the primitives π_i^e and π_j^e that face each other; and, $\mathbf{d}_c(\mathbf{c}_i, \mathbf{c}_j)$ is Euclidean distance between RGB values of the colors $\mathbf{c}_i$ and $\mathbf{c}_j$. Co-colority between an edge primitive π^e and and a mono primitive π^m, and between two monos can be defined similarly (not provided here). In figure 7.6, a pair of co–color and not co–color primitives are shown.

7.4 Formulation of the Model

For the prediction of the depth at monos, we developed a voting model. Voting models are suitable for producing a result from data which includes outliers. In a voting model, there are a set of voters that state their *opinion* about a certain event e. A voting model combines these votes in a reasonable way to make a decision about the event e.

In the depth prediction problem, the event e to be voted for is the depth and the 3D orientation of a mono π^m, and the voters are the edge primitives $\{\pi_i^e\}$ (for $i = 1, ..., N_E$) that bound the mono. In this chapter, we are interested in the predictions of pairs of π_i^es, which are denoted by P_j for $j = 1, ..., N_P$. While forming a pair P_j from two edges π_i^e and π_k^e from the set of the bounding edges of a mono π^m, we have the following restrictions:

1. π_i^e and π_k^e should share the same color with the mono π^m (*i.e.*, the following relations should hold: $coc(\pi_i^e, \pi_k^e) > T_{coc}$ and $coc(\pi_i^e, \pi^m) > T_{coc}$).

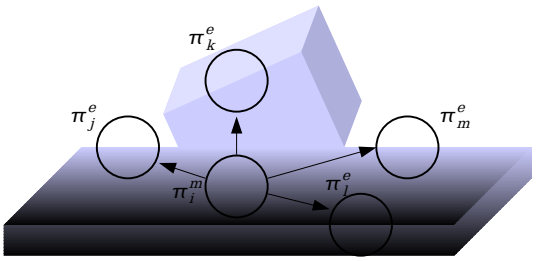


Figure 7.7: A set of primitives for illustrating why the relations co-planarity, co-colority and linear dependence are required as restrictions for forming pairs from edges.

2. The 3D primitives $\mathbf{\Pi}^e_i$ and $\mathbf{\Pi}^e_k$ of π^e_i and π^e_k should be on the same plane (*i.e.*, $cop(\mathbf{\Pi}^e_i, \mathbf{\Pi}^e_k) > T_{cop}$).

3. π^e_i and π^e_k should not be linearly dependent so that they define a plane (*i.e.*, $lin(\pi^e_i, \pi^e_k) < T_{lin}$).

In figure 7.7, such restrictions are illustrated for an example mono and a set of edge primitives that bound it. The primitives π^e_j and π^e_m are on the same line (*i.e.*, they are linearly dependent), and hence, they define infinitely many planes. As for primitives π^e_l and π^e_k, they cannot define a plane, nor do they share the same color.

The vote v_i by a pair P_j can be parameterized by:

$$v_i = (\boldsymbol{X}, \vec{n}), \tag{7.5}$$

where $\vec{n}$ is the normal of the mono π^m, and $\boldsymbol{X}$ is its depth.

Each v_i has an associated reliability or probability r_i. They denote how likely the vote is based on the believes of the pair P_i. It is suggested in [Kalkan et al., 2007e] and chapter 6 that the likelihood of a local surface patch being co-planar with a 3D edge feature decreases with the distance between them. Inspired from the results in [Kalkan et al., 2007e] and chapter 6, we define the reliability r_i of a vote v_i as:

$$r_i = 1 - \frac{1}{min(d(\pi^m, \pi^e_1), d(\pi^m, \pi^e_2))}, \tag{7.6}$$

where $d(.,.)$ is the Euclidean image distance between two features.

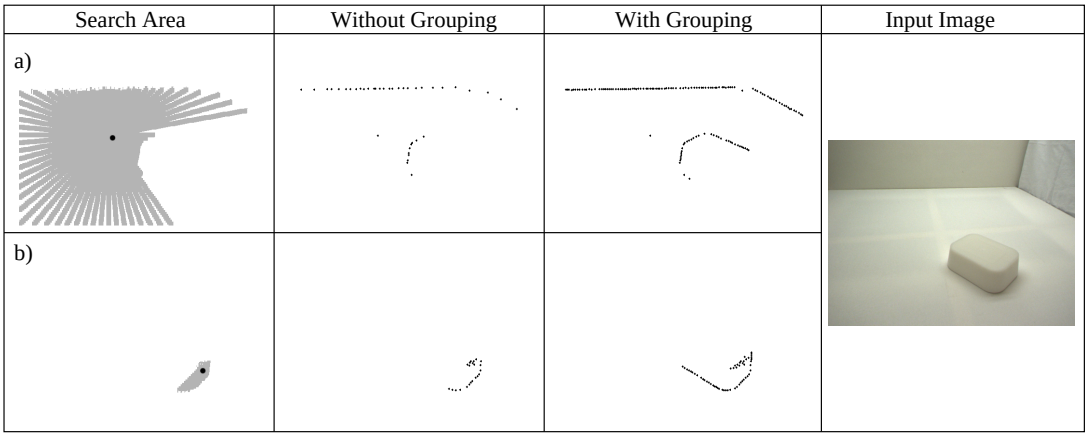


Figure 7.8: Finding bounding edge primitives with and without grouping information for two different monos which are marked in black in the first column. Using grouping information produces a more complete boundary finding as shown in (a). However, using grouping may include unwanted edge primitives in the boundary as shown in (b).

7.4.1 Bounding edges of a mono

Finding the bounding edges of a mono π^m requires making searches in a set of directions d_i, $i = 1, ..., N_d$ for the edge primitives. In each direction d_i, starting from a minimum distance R_{min}, the search is performed up to a distance of R_{max} in discrete steps s_j, $j = 1, ..., N_s$. If an edge primitive π^e is found in direction d_i in the neighborhood Ω of a step s_j, π^e is added to the list of bounding edges and the search continues with the next direction d_{i+1}.

The above mentioned method for finding the bounding edge primitives will lead to an incomplete and sparse boundary detection (see figure 7.8) because the search is performed only in a set of discrete directions. This can be improved by making use of the contour grouping information; when an edge primitive π^e is found in a direction d_i at step s_j, if π^e is part of a group G, then all the edge primitives in G can be added to the list of bounding edges (see [Pugeault et al., 2006] and appendix B for information about the grouping method we employ in this chapter).

Grouping information can lead to more complete and dense boundary finding as shown in figure 7.8(a); however, for certain objects, it may lead to worse results due to low contrast edges (see figure 7.8(b)).

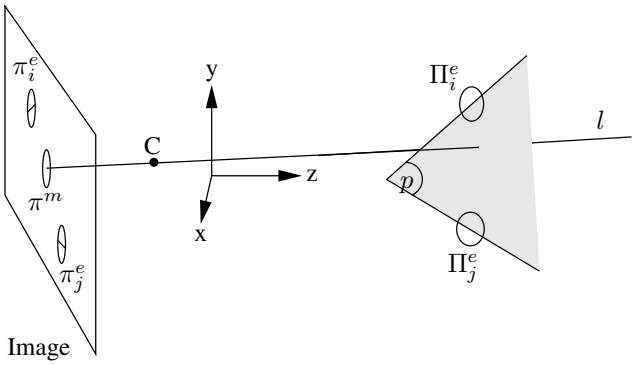


Figure 7.9: Illustration of how the vote of a pair of edge primitives is computed. The 3D primitives $\mathbf{\Pi}_i^e$ and $\mathbf{\Pi}_j^e$ corresponding to the 2D primitives π_i^e and π_j^e define the plane p. The intersection of p with the ray l that goes through the 2D mono π^m and the camera center C then determines the position of the estimated 3D mono Π^m. The 3D orientation of Π^m is set to be the orientation of the plane p.

7.4.2 The vote of a pair of edge primitives on a mono

A pair P_i of two edge primitives π_j^e and π_k^e with two corresponding 3D edge primitives $\mathbf{\Pi}_j^e$ and $\mathbf{\Pi}_k^e$, which are co-planar, co-color and linearly *independent*, defines a plane p with 3D normal $\mathbf{n}$ and position $\mathbf{X}$.

The vote v_l of $\mathbf{\Pi}_j^e$ and $\mathbf{\Pi}_k^e$ is computed by the intersection of the plane p with the ray l that goes through the mono, π^m, and the optical center of the camera (see figure 7.9). The ray l is computed using the following formula ([Faugeras, 1993], pg41):

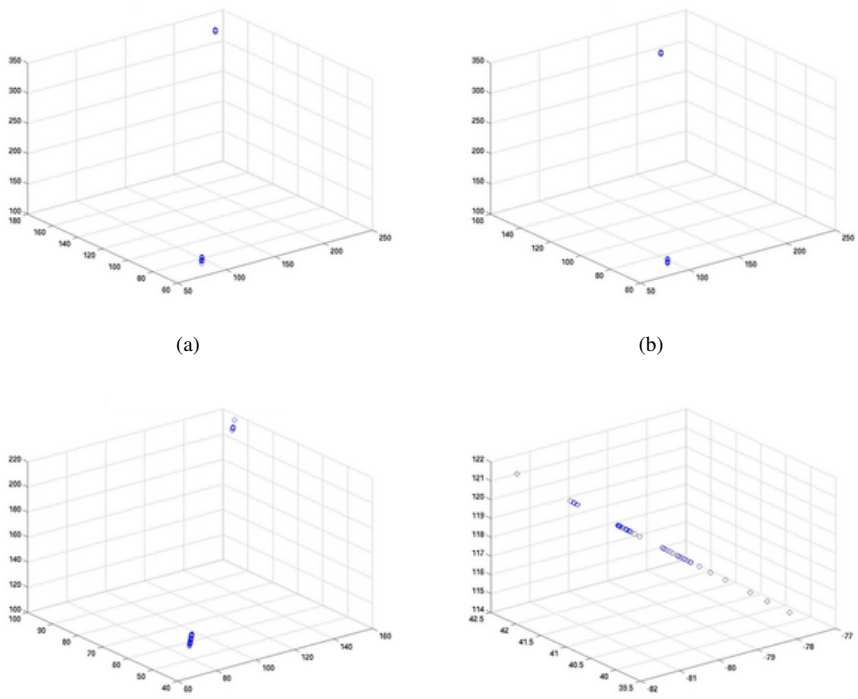
$$\mathrm{X_a} = M^{-1}(-\tilde{p} + \lambda\tilde{x}), \tag{7.7}$$

where $\tilde{x}$ is the homogeneous position of π^m; M and $\tilde{p}$ are respectively the 3x3 and the 3x1 sub-parts of the 3x4 projection matrix P_m so that $P_m = [M \;\; \tilde{p}]$; and, λ is an arbitrary number. By using two different values for λ, two different points on ray l are extracted which then are used to compute the ray l.

Because the ray l is unique for a mono π^m, all the votes the mono π^m will be on ray l (figure 7.10). This property can be exploited for clustering the votes.

7.4.3 Combining the votes

The votes can be integrated using different ways to estimate the 3D representation Π^m of a 2D mono π^m. One way is to take the weighted average of the votes. Weighted averaging is adversely affected by the outliers. For this reason, we cluster the votes and do the averaging inside the *best* cluster. The votes



(a) (b)

(c) (d)

Figure 7.10: The distribution of the votes for a few monos shown in the 3D Euclidean space. Sub-figures (a)-(c) are clear examples where there are two clusters. However, such clear clusters do not always exist (d).

and how they cluster in 3D Euclidean space are shown in figure 7.10 for a few examples. Two clusters in figure 7.10 can be considered as two different depth hypotheses that can be distinguished by a higher level process.

Let us denote the clusters by c_i for $i = 1, ..., N_c$. Then,

$$\Pi^m = \arg\max_{c_i} \ \#c_i. \tag{7.8}$$

where # is the cardinality of a cluster. Equation 7.8 defines the best cluster as the most crowded one. The best cluster can be alternatively chosen to be a cluster which has the highest reliability. In this paper, we adopted the definition in equation 7.8.

Clustering the votes can filter outliers out whereas it is slow. Moreover, it is not trivial to determine the number of clusters from the data points that will be clustered.

In this paper, we implemented (1) a histogram-based clustering where the number of bins is fixed, and the best cluster is considered to be the bin with the most number of elements, and (2) a clustering algorithm where the number of clusters is determined automatically by making use of a cluster-regularity measure and maximizing this measure iteratively.

(1) is a simple but fast approach whereas (2) is considerably slower due to the iterative-clustering step. Our investigations showed that (1) and (2) produce similar results (the comparative results are not provided in this paper). For this reason, we have adopted (1) as the clustering method for the rest of the paper.

The best two clusters of the votes (figure 7.10) at a mono can be considered as two different depth hypotheses. Our depth model can be extended easily so that it keeps two different hypotheses at each mono until they can be disambiguated by a higher-level process.

7.4.4 Combining the predictions using area information

3D surfaces project as areas into 2D images. Although one surface may project as many areas in the 2D image, it can be assumed most of the time that the image points in an image area are part of the same 3D surface.

Figure 7.11(a) shows the predictions of a surface. Due to possible outliers in the stereo computation, depth predictions are scattered around the surface that they are supposed to represent. We show that it is possible to segment the 2D image into areas based on intensity similarity and combine the predictions in areas to get a cleaner and more complete surface prediction.

We segment an input image $\boldsymbol{I}$ into areas A_i, $i = 1, .., N_A$ using co-colority (see section 7.3) between primitives utilizing a simple region-growing method; the areas are grown until the image boundary or an edge-like primitive is hit. Figure 7.11(b) shows the segmentation of one of the images from figure 7.1. For more examples, see figure 7.12.

In this chapter, we assume that each A_i has a corresponding surface S_i defined as follows:

$$S_i(x, y, z) = \mathrm{a}x^2 + \mathrm{b}y^2 + \mathrm{c}z^2 + \mathrm{d}xy + \mathrm{e}yz + \mathrm{f}xz + \mathrm{g}x + \mathrm{h}y + \mathrm{i}z = 1. \tag{7.9}$$

Such a surface model allows a wide range of surfaces to be represented, including spherical, ellipsoid, quadratic, hyperbolic, conic, cylindric and planar surfaces.

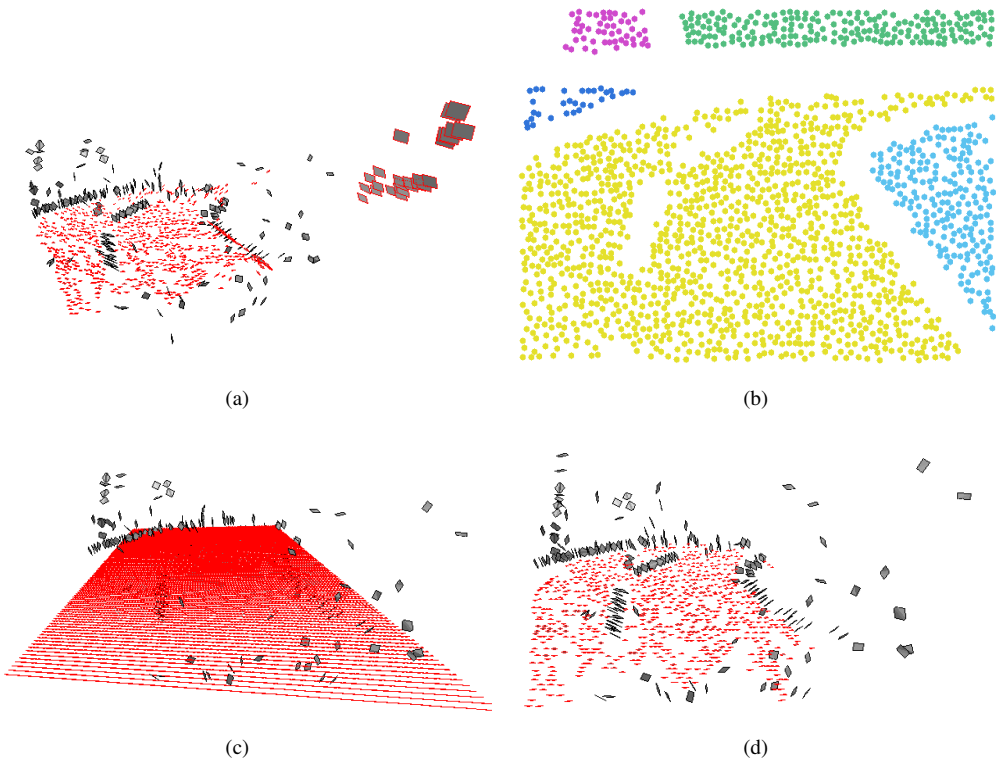


Figure 7.11: **(a)** The predictions on the surface of the road for the input images shown in figure 7.1 (predictions are marked with red boundaries). The predictions are scattered around the plane of the road, and there are wrong predictions due to possible outliers in the computed stereo. The figure is a snapshot from our 3D displaying software. **(b)** Segmentation of one of the input images given in figure 7.1 into areas using region-growing based on primitives. **(c)** The surface extracted from the predictions shown in (a). **(d)** The predictions from (a) that are corrected using the extracted surface shown in sub-figure (c).

S_i is estimated from the predictions in A_i by solving for the coefficients using a least-squares method. As there are nine coefficients, such a method requires at least nine predictions to be available in area A_i. For the predictions shown in figure 7.11(a), the estimated surface is shown in figure 7.11(c) using a sparse sampling.

Having an estimated S_i for an area A_i makes it possible to *correct* the mono predictions using the estimated surface S_i: Let $\mathbf{X}_n$ be the intersection of the surface S_i with the ray that goes through π^m and the camera, and $\mathbf{n}_n$ be the surface normal at this point (defined by $\mathbf{n}_n = (\delta S_i/\delta_x, \delta S_i/\delta_y, \delta S_i/\delta_z)$). $\mathbf{X}_n$ and $\mathbf{n}_n$ are respectively the corrected position and the orientation of mono Π^m.

Corrected 3D monos for the example scene is shown in figure 7.11(d). Comparison with the initial predictions which are shown in figure 7.11(a) concludes that (1) outliers are *corrected* with the extracted

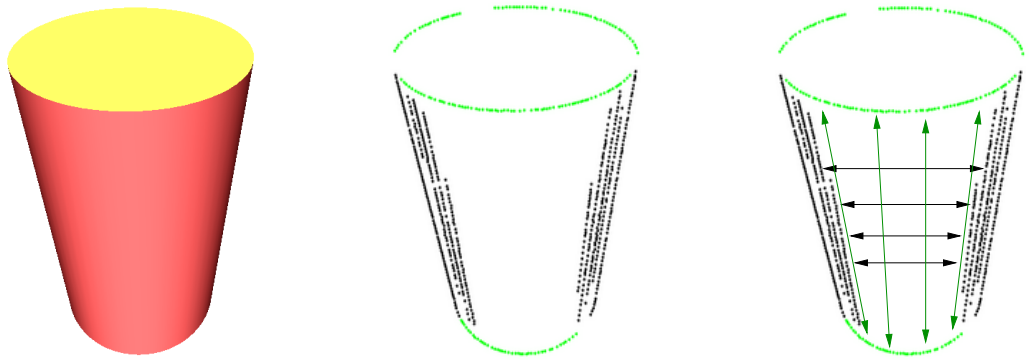
Figure 7.12: A set of images (on the left) and the extracted areas (on the right). Homogeneous primitives belonging to different areas are encoded in different colors.

surface representation, and (2) orientations and positions are qualitatively better. Surface information can further be used to remove the possible outliers in the 3D edge features that are extracted using stereo.

7.4.5 Round object mode

Since the votes of every pair of edge features in the boundary of an image area are considered for making a depth prediction, the above described method is biased towards estimating a planar surface even on round surfaces. In this section, a solution involving the curved groups of image areas is proposed. The reason for using curved groups is because curvature is known to be a non-accidental feature [Biederman, 1987] which suggests the existence of a round surface. As will be shown in section 7.6.2, dense methods also fail at round surfaces due to implicit disparity smoothness constraints.

The proposed solution can reconstruct curved surfaces of round objects only if the cameras are positioned so that their 2D projections have curved edges. Otherwise, the 2D edges of the object can



(a) (b)

Figure 7.13: Detected curved groups (b) which are extracted from (a). The edges of a curved group are shown in green. There are several 'layers' of groups at the vertical edges of the cylinder due to shading. The vertical groups (associated with horizontal arrows) shown in (c) predict a planar surface, and in order to predict a round surface, the curved groups need to be used.

reconstruct only a straight edge. Moreover, the proposed solution works only if the curvature of the surface changes in one direction, because of which ovoidal (such as spherical and ellipsoidal) surfaces can not be recovered.

We add a *round object mode* to our model. In this mode:

1. the bounding edges of a mono are grouped,
2. these groups are *ordered* (see below),
3. the *curved* groups are found (see below),
4. if there are at least two curved groups, *only* the curved groups are allowed to make predictions,
5. a one-to-one association is established between ordered curved groups,
6. predictions are accepted only from the pairs of such one-to-one associations.

Extraction of the 3D surface by using the curvature of the groups is along the lines of 3D shape interpretation from the line drawings of objects [Barrow and Tenenbaum, 1981, Nalwa, 1989, Stevens, 1981, Ulupinar and Nevatia, 1991, Ulupinar and Nevatia, 1993].

The depth prediction method can switch to the the round object mode by detecting whether there are enough curved groups. If there are two or more curved contours, the method can run in the round object

mode. In case the detected curved groups do not correspond to a round surface, the method predicts a planar surface. Note that the round object mode depends on good-quality extraction of groups from an image, which however is not usually the case in real images. Even for the example simple scene in figure 7.13, it is not trivial to separate the bottom group and the vertical edges since a threshold of grouping similarity that separates them will also break up the group at the top of the cylinder.

Ordering of a group and creating one-to-one association between two groups.

Ordering of a group amounts to ordering of a set of (x, y) points. In the case of a group, however, having a 2D orientation at these points simplifies the problem. The process of ordering requires a *less-than* relation to be defined for its input points. For two edges π_i^e and π_j^e we define the following approximate relation:

$$\text{less-than}(\pi_i^e, \pi_j^e) = \begin{cases} \text{true} & \text{if } l_i^n \cdot (x_j, y_j, 1) > 0, \\ \text{false} & \text{otherwise,} \end{cases} \tag{7.10}$$

where l_i^n is the line $ax + by + c = 0$ orthogonal to the orientation defined by the primitive π_i^e. In words, this relation states that primitive π_i^e is less-than π_j^e if π_j^e lies to the right of the line defined by the normal orientation of π_i^e.

Note that if the the normal of a primitive inside the group self-intersects the group, then this ordering will not work. To tackle such situations, a computationally more expensive ordering method may be employed.

Having ordering allows for one-to-one[3] association between the groups. Two groups g_i and g_j with N_i and N_j being the number of elements are associated as follows:

1. The groups are aligned; *i.e.*, the beginning and the end of groups are adjusted so that the ascend inside each group are in the same direction.

2. If N_i equals N_j, then a one-to-one mapping is the resulting association.

3. Assuming g_i is the shorter group; each element of g_i is mapped to approximately N_j/N_i many elements of g_j in *order*.

[3] In fact, in mathematical terms, this is a one-to-many relationship as the number of elements in groups are not the same.

Detection of curved groups.

We define a group to be curved if the standard deviation of the orientations of the primitives inside the group is less than an empirical threshold 0.3. Curved groups for an example scene are shown in figure 7.13 in green.

7.5 Dense Stereo Methods

The depth prediction method proposed in this chapter is a stereo-based model which can produce depth information at *weakly-textured* image areas. Since dense methods are also stereo-based and functional at *textured* image areas, the depth prediction method needs to be compared to dense stereo methods.

Dense methods for stereo include in general the following two main steps: (1) computation of how similar an image point is to an image point in another view, and (2) disparity computation or optimization [Scharstein and Szeliski, 2001]. The most-widely used similarity functions are 'sum of squared intensity differences' (SSD) and 'sum of absolute intensity differences' (AD) [Brown et al., 2003]. The disparity computation step can be either (i) local, where the winner is determined simply by taking the match with the maximum similarity value (*i.e.*, winner-take-all (WTA)), or (ii) global, where the winner is determined by maximizing an energy function defined over the space of all possible disparities and the corresponding similarity functions [Scharstein and Szeliski, 2001]. Some of the global optimization methods include dynamic programming (DP), scanline optimization (SO), graph-cuts, belief propagation and intrinsic curves [Brown et al., 2003].

The dense methods with global optimization are more accurate; they yield smoother disparity maps, and they can integrate prior knowledge about the scene more easily. However, they are computationally expensive. The local dense methods, on the other hand, are faster but less accurate. Section 7.6.3 provides comparison of two local and two global dense methods. The various dense stereo methods that are used in this chapter are briefly introduced in the rest of the section.

7.5.1 Phase-based approach (PB)

The dense stereo method that uses the phase of the signals is taken from [Sabatini et al., 2007]. In [Sabatini et al., 2007], for a stereo pair of images $I^R(\boldsymbol{x})$ and $I^L(\boldsymbol{x})$ where $I^L(\boldsymbol{x}) = I^R[\boldsymbol{x}+d(\boldsymbol{x})]$, the disparity

$d(\boldsymbol{x})$ is computed as follows:

$$d(\boldsymbol{x}) = \frac{\left[\phi^L(\boldsymbol{x}) - \phi^R(\boldsymbol{x})\right]2\pi}{\omega(\boldsymbol{x})}, \tag{7.11}$$

where ϕ is the phase at point $\boldsymbol{x}$, and $\omega(\boldsymbol{x})$ is the average instantaneous frequency of the bandpass signal. [Sabatini et al., 2007] implements a coarse-to-fine control scheme to find the disparity estimates.

7.5.2 Region matching with squared sum of differences (SSD)

Region matching with squared sum of differences (SSD) is a local method which has squared sum of differences as the matching function $s(\boldsymbol{x})$ [Brown et al., 2003]:

$$s(\boldsymbol{x}, d) = \sum_{\boldsymbol{x}} \left[I^L(\boldsymbol{x}) - I^R(\boldsymbol{x} + d)\right]^2, \tag{7.12}$$

for different values of d. In this chapter, we have taken the implementation from [Scharstein and Szeliski, 2001] which uses winner-take-all optimization along the epipolar line.

7.5.3 Region matching with absolute differences and a scanline global optimization (SO)

Region matching with absolute differences and a scanline global optimization is a global method which has the following matching function [Brown et al., 2003]:

$$s(\boldsymbol{x}, d) = \sum_{\boldsymbol{x}} \left|I^L(\boldsymbol{x}) - I^R(\boldsymbol{x} + d)\right|, \tag{7.13}$$

for different values of d. The scanline optimization tries to minimize the total matching cost along a scanline (*i.e.*, a row) of the left image (assuming that the disparity space is scanned from the left image). In this chapter, we have taken the implementation from [Scharstein and Szeliski, 2001].

7.5.4 Region matching with absolute differences and a dynamic programming global optimization (DP)

This global method uses the absolute differences in equation 7.13 as the matching function, and optimizes the matching cost using dynamic programming. Dynamic programming is a method for solving problems

that involve overlapping subproblems. The classical example for demonstrating the concept of DP goes like this: Finding the minimum path between two nodes n_i and n_j through a node n_k is equivalent to finding the minimum paths between n_i and n_k, and between n_k and n_j. In the case of stereo, dynamic programming is applied to find the minimum path in the disparity space image (*i.e.*, $(\boldsymbol{x}, d)$).

DP may include inter-scanline optimization to optimize disparity values between different scanlines. In this chapter, we have the implementation from [Scharstein and Szeliski, 2001] which does not have inter-scanline optimization.

7.6 Results

This section evaluates the performance of the proposed depth prediction method and compares its performance to standard dense stereo methods. The aim is, however, not to claim that depth prediction is better than dense methods, rather to show that depth prediction is a *different cue* which can provide comparable and, at weakly-textured scenes, better results. For the comparison, the following dense methods are used: (1) our depth prediction method *without* surface corrections (DeP); (2) a phase-based (PB) dense stereo from [Sabatini et al., 2007]; (3) squared sum of differences (SSD) as the matching function with a winner-take-all approach; (4) absolute differences as the matching function with a scanline optimization (SO); and, (5) absolute differences with a dynamic programming optimization (DP).

The dense methods are expected to perform better at textured image areas whereas DeP should produce better results at weakly-textured image areas. Due to their global optimization stage, SO and DP should be better than SSD and PB. The images which the dense stereo algorithms are applied to were rectified and down-sampled when needed.

7.6.1 Results on road scenes

The results of our model as well as DP and PB (with two different thresholds) are shown in figure 7.14 for a real scene which includes occlusion and texture. We see that our method is able to provide comparable performance to dense stereo algorithms. Although our algorithm performs well on textured surfaces, the effect of the wrong predictions from the occluding edges are visible especially around the traffic sign. Moreover, due to the uncertainty on the left edge of the road and as least-squares fitting is affected by the outliers adversely, the surface on the left is badly reconstructed. Occlusions are a problem for dense

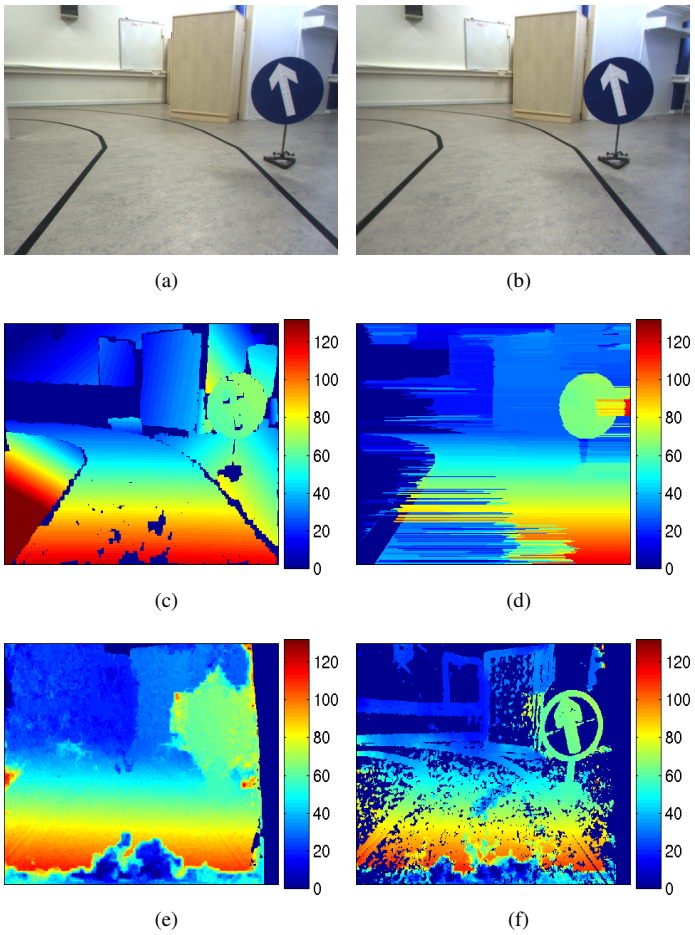


Figure 7.14: Experiment results on a road scene. **(a,b)** Input stereo pair. **(c)** The predictions of our model as a disparity map. **(d)** Disparity map from DP. **(e)** Disparity map from PB. **(f)** Subfigure (e) after a small threshold (0.001).

stereo algorithms as well (as seen in *e.g.*, figure 7.14(e)). DP however can perform better on occluded areas due to its global optimization; however, DP does not produce results on the left side of the scene. As shown in figure 7.14(f) for PB, using a reliability threshold on the disparity values can get rid of most of the outliers in figure 7.14(e), however, lowering the threshold decreases the most of the inliers of the disparity map.

Another example in figure 7.15 shows that in spite of limited 3D information from feature-based stereo, our method is able to predict the surfaces. Moreover, we see from figure 7.15 that our method is able to utilize the little information at the right side of the road to predict the 3D information.

The results on another road scene is shown in figure 7.16. The depth prediction model is able to

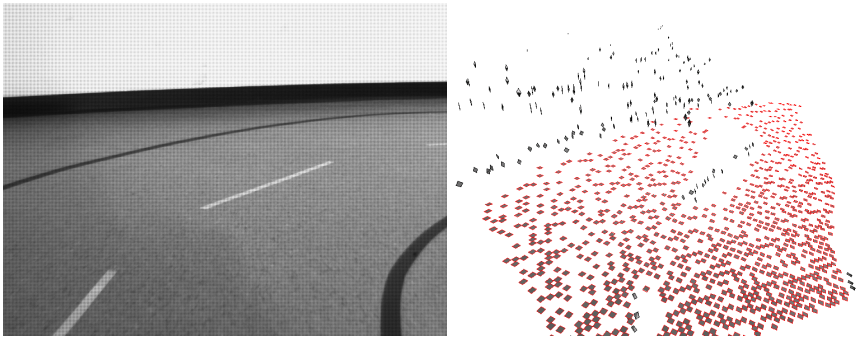


Figure 7.15: Experiment results on a lab road scene. **Left:** Left image of the input stereo pair. **Right:** The predictions of our model shown in our 3D displaying software.

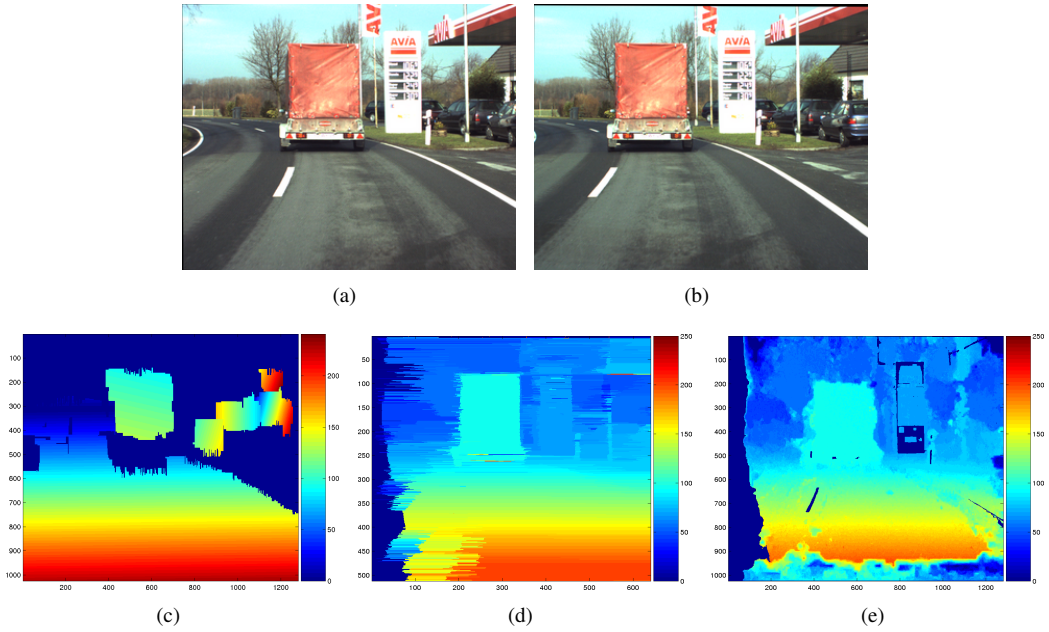


Figure 7.16: Experiment results on a road scene. **(a,b)** Input stereo pair. **(c)** The predictions of our model as a disparity map. **(d)** Disparity map from DP. **(e)** Disparity map from PB.

reconstruct the road better than other methods. DeP fails at the small areas since they are far from the camera and 3D orientation at the edges is not reliable. However, DP and PB perform better on small and textured surfaces.

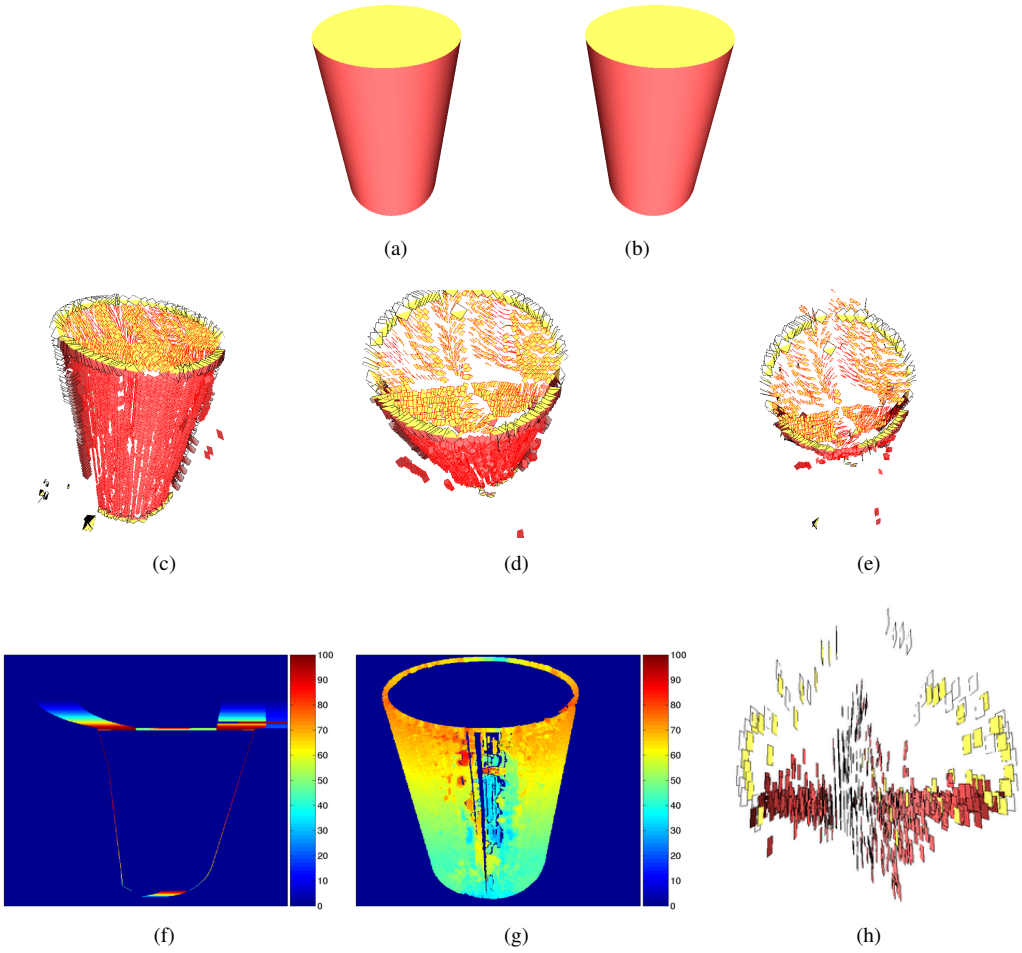


Figure 7.17: Experiment results on a cylinder. **(a,b)** Input stereo pair. **(c-e)** The predictions of our model shown as snapshots from our 3D displaying software. As surface fitting for curved surfaces in case of outliers is not trivial, we are unable to provide disparity maps for our results. **(f)** Disparity map from DP. **(g)** Disparity map from PB. **(h)** Top view of the results of PB.

7.6.2 Results on a round object

As mentioned in section 7.4.5, it is difficult to extract surfaces on round objects using stereo since the curvature is hidden in the shading or the texture-gradient of the surface, or depth extraction requires object knowledge. In this subsection, we evaluate the round object mode of the depth prediction method on a cylinder.

The results of DeP, DP and PB are shown in figure 7.17 for a cylinder. We see from the figures that DP and PB have problems on non-textured round objects; PB estimates disparities corresponding

to mainly a flat surface, whereas DP produces results only at parts of the edges and the shading. Dense methods fail at this scene because (1) the object surface does not contain any texture, which makes the correspondence problem unsolvable, and (2) dense methods assume implicitly some linearity assumption (through smoothing) that leads to disparity estimation.

7.6.3 Quantitative comparison with dense stereo

The depth prediction method proposed in this chapter is a stereo-based model which can produce depth information at *weakly-textured* image areas. Since dense methods are also stereo-based and functional at *textured* image areas, the depth prediction method needs to be compared to dense stereo methods. Extraction of 3D information at textured surfaces is difficult since a texture most of the time consists of repetitive structures, which are difficult to match locally across different views. Moreover, image noise and illumination increase the difficulty of texture matching locally. Consider, for example, a round surface whose curvature is provided in the texture on the surface. The 3D information can only be recovered by using the texture gradient, and therefore, matching textured image patches may not be sufficient to recover the depth at textured surfaces.

The comparisons are performed on an artificial scene where the texture could be controlled in order to see the behaviours of the different approaches. The texture is *white* noise, and the amount of texture is controlled by the frequency (n $\in [0, 0.2]$) of the white-noise. We tried n up to 0.2 because the images get over-textured for bigger values of n. A subset of the input images is shown in figure 7.18.

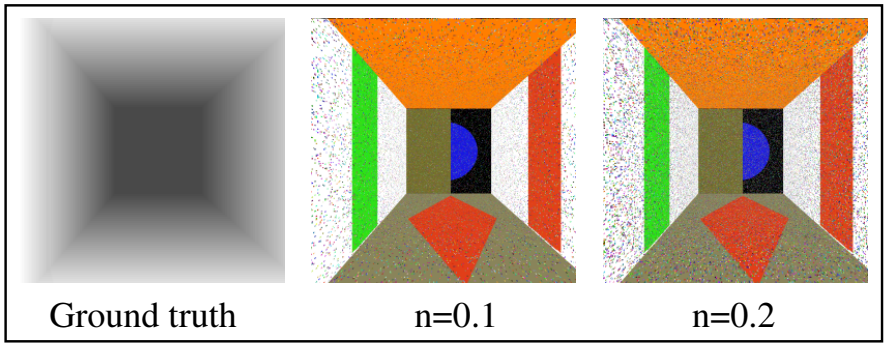


Figure 7.18: A subset of the textured artificial images that have been used. Added texture is *white* noise with a frequency n.

The expectation is to see that dense stereo methods perform poor on weakly-structured scenes where

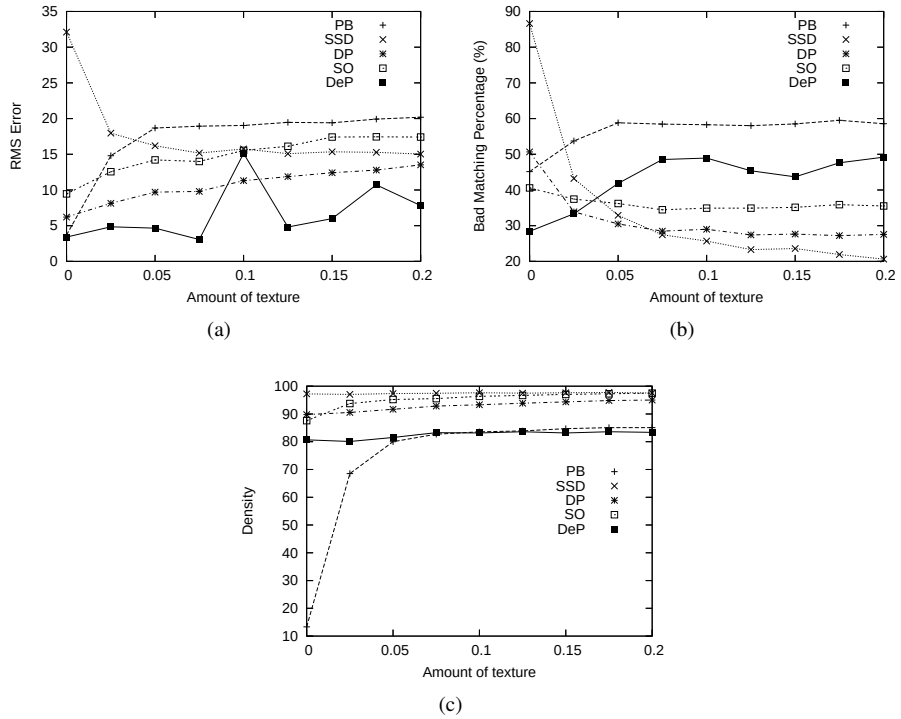


Figure 7.19: Performance of the different algorithms on the artificial scene in figure 7.18 for different amount of texture using RMS (a), BMP (b) measures. The densities are shown in (c).

our model should make good predictions. When the amount of texture is increased, dense stereo methods should perform better, and the predictions made by our model should degrade because an increase in texture causes the features to be less reliable and noisy.

For evaluation against a ground truth d_G, we used two disparity error measures: Root-Mean-Squares (RMS) and Bad-Matching-Percentage (BMP). RMS is the standard measure that has been used in the literature for evaluating the performance of stereo algorithms (see, *e.g.*, [Scharstein and Szeliski, 2001]):

$$\mathrm{RMS}(\mathcal{S}) = \Big(\frac{1}{\#\mathcal{S}} \sum_{\mathbf{p}\in\mathcal{S}} |d_C(\mathbf{p}) - d_G(\mathbf{p})|^2\Big)^{1/2}, \tag{7.14}$$

where $\mathcal{S}$ is the set of points with disparity information; and, $d_C(\mathbf{p})$ and $d_G(\mathbf{p})$ are respectively the computed and the ground truth disparity information at point $\mathbf{p}$.

(a) (b) (c)

Figure 7.20: Weak lines applied on the artificial scene from figure 7.18. Only portions of the images are provided for better visibility. (a) Irregular lines. (b) Regular horizontal lines. (c) Regular vertical lines.

The BMP measure (taken from [Scharstein and Szeliski, 2001]) is defined as follows:

$$\mathrm{BMP}(\mathcal{S}) = \frac{1}{\#\mathcal{S}} \sum_{\mathbf{p}\in\mathcal{S}} (|d_C(\mathbf{p}) - d_G(\mathbf{p})| > 1), \tag{7.15}$$

RMS errors in figure 7.19(a) shows that our method is more accurate than dense stereo methods. Comparison with BMP errors in figure 7.19(b) suggests RMS evaluation of dense methods are affected by the outliers. In general, we see that when there is no texture, our method is better than dense methods; the reverse is the case when the image is textured. The density plot in figure 7.19(c) confirms that our method can produce highly dense disparity maps at un-textured images.

We compared the performance of the different approaches using a different texture on the same artificial scene from figure 7.18. The type of texture is weak lines (see figure 7.20): regularly sampled vertical and horizontal lines, and irregularly sampled and sized lines. The reason for using additional types of textures is to see whether the methods are biased towards directed and repetitive textures and to the direction of the textures. The performance of dense stereo methods and our model are shown in figure 7.21. Again we observe that our depth prediction method can provide comparable results to DP, and better results than other approaches.

Finally, we compared the performance of the algorithms on noisy images (again using the artificial scene used above). This comparison is important because signal to noise ratio at weakly-textured image areas are higher than textured image areas, for the same amount of noise. We added white noise with a

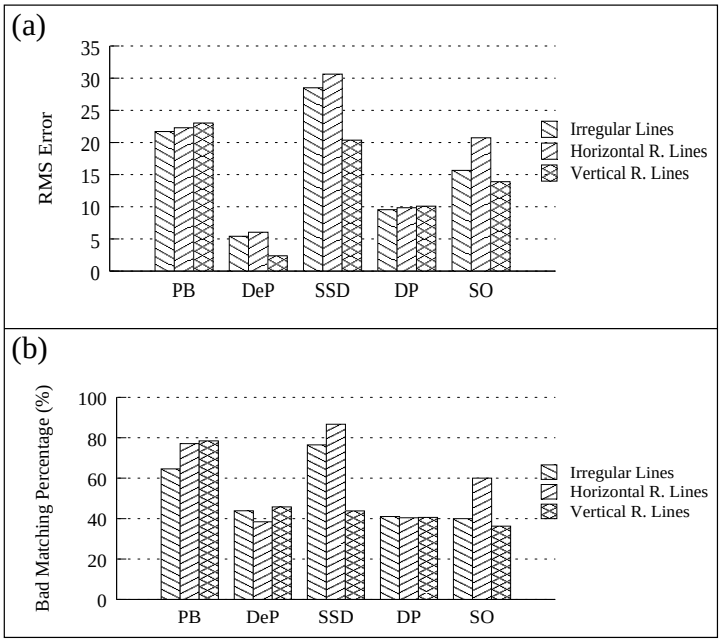


Figure 7.21: Performance of different algorithms on the artificial scene in figure 7.20 for different amount of texture (n) using RMS (a) and BMP (b) measures.

frequency between 0 and 0.2 and plotted the performance for different amount of texture (figure 7.22). The performance of dense methods are severely affected by noise since they work at the signal level. Our depth prediction method, on the other hand, is more robust because edge features are less sensitive to noise.

7.6.4 Integration with dense stereo information

Results from the previous section suggests that sparse and dense stereo (and, hence, DeP) perform well on different types of images: DeP performs well when there is not much texture whereas dense methods perform better when a scene is *well*-textured.

In this section, we combine DeP with the disparity information acquired from one of the dense methods:

- **if a primitive is textured:** The disparities in the patch are converted to 3D points. A plane is fitted to these points in 3D, and the intersection of this plane with the optic ray from the primitive defines the 3D position of the primitive. The normal of the primitive is defined to be the normal of the

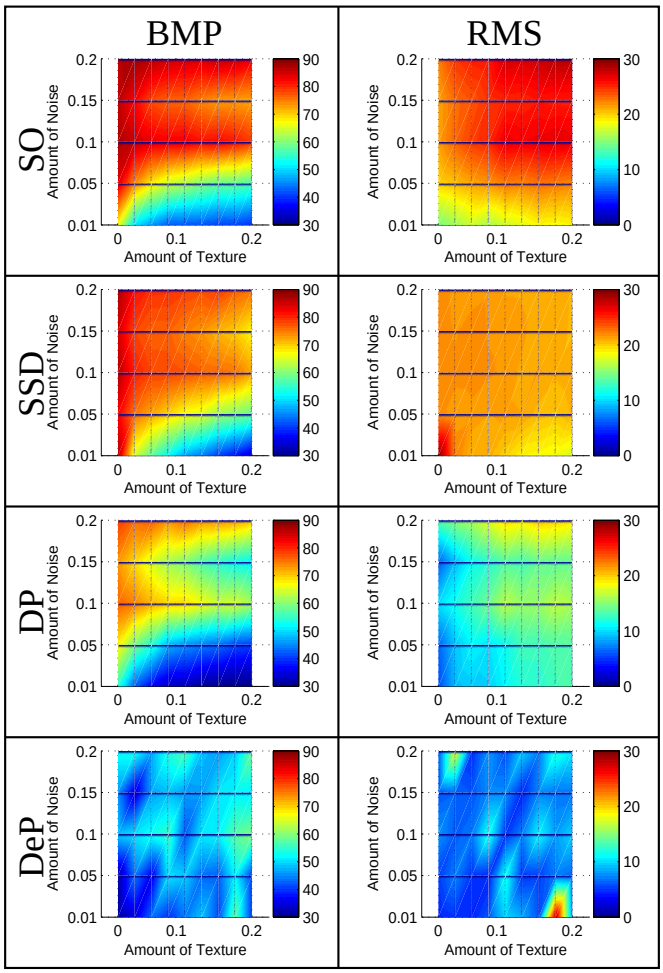


Figure 7.22: Performance of the different algorithms as a function of white noise and texture (white noise). Results of PB are skipped due to page limits.

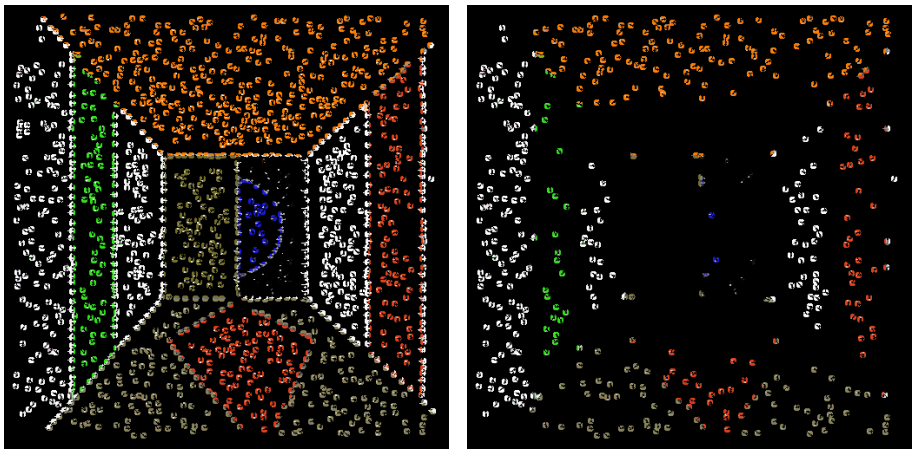
plane.

For detecting textured areas, we use intrinsic dimensionality (see section 2.1): an image point p is textured if origin variance is in $[0.2, 1.0]$ and line variance is in $[0.2, 1.0]$. Note that this scheme classifies corners as textured areas too, which is desirable because DeP does not utilize corners.

- **if a primitive is weakly-textured:** The predictions from DeP are used.

The features with texture for one artificial scene (taken from figure 7.18 with texture ratio $n = 0.025$) are shown in figure 7.23.

In figure 7.24, the results of the combination are provided for the artificial data set in figure 7.18



(a) (b)

Figure 7.23: Illustration of textured features. **(a)** All the features extracted from the artificial set shown in figure 7.18 where n is set to 0.025. **(b)** Detected textured features from (a).

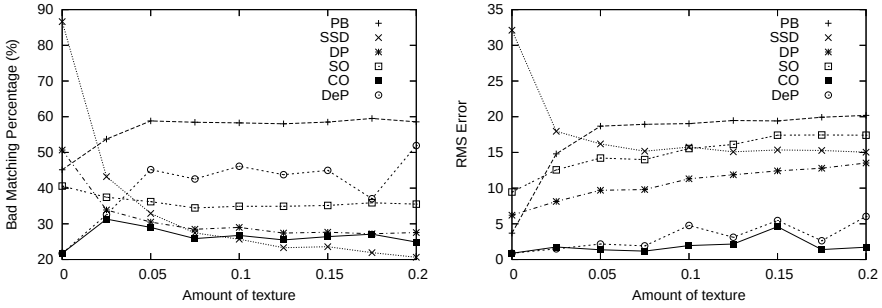


Figure 7.24: The results of combination (CO) with dense stereo (namely, DP) on the artificial data set from figure 7.18. (a) Bad matching performance. (b) RMS errors.

(combination is labeled with CO). We see from bad matching percentages that CO improves significantly over DeP and is slightly better than DP. In RMS errors, the reverse is the case: bad matching percentages show significant improvement of CO over DP while it is only slightly better than DeP. The figure shows that the combination of DP and DeP can make use of the benefits of both approaches and show better performance.

It is crucial to note that the proposed integration with dense stereo is naive and developed only for proof of concept. A better integration scheme should extensively make use of the compatibilities or the conflicts of the different hypotheses of the different approaches. For example, if the hypotheses are compatible with each other, their confidences should be amplified; otherwise, the two hypotheses can be

kept until they can be disambiguated by higher-level processes. See [Aloimonos and Shulman, 1989] for more reading about the integration of different cues.

7.6.5 Time issues

Although the execution time of the depth prediction method was not in the focus of our efforts since the implementation is not optimized, in this subsection, we evaluate the running times of the different methods. The amount of time the depth prediction method needs is dependent on the amount of homogeneous image areas, which depends on the scene as well as the image size.

For the example 512x512 pixel^2 scene in figure 7.18, it takes 40-50 seconds for the depth prediction method, including the extraction of the features and the computation of stereo. As for the dense methods, for example the phase-based approach, the computation time is in the same range. For a scene of 1024x768 pixel^2, the depth prediction methods takes 10-30 minutes, depending on the scale of the visual features that are used and the amount of homogeneous image areas. Dense methods other than the phase-based approach fails for such scenes due to the requirements of excessive memory usage, and the phase-based approach requires approximately 10 minutes.

Finding the bounding edges of a mono is the slowest part of the depth prediction implementation. Currently, the bounding edges for each mono in an area are found separately for each of them. Alternatively, the bounding edges can be found once and associated to the respective areas. When the bounding edges of a mono is needed, the bounding edges that has been associated to the area of the mono can be used without making a new search. However, this approach may produce undesirable predictions as suggested in figure 7.8.

7.6.6 Limitations of the current work

The proposed method currently depends on the 3D information from a feature-based stereo system, and therefore, the predictions are adversely affected at image areas where the stereo information is not accurate or available. In [Pugeault et al., 2008], it was shown that the uncertainity of reconstructed 3D points and 3D line orientation increases with the distance from the camera. The depth prediction method fails at small image areas since there are not enough surrounding edges or 3D information. Moreover, due to the geometry of the cameras, stereo can not match edges which are parallel to the epipolar line. For this

reason, the depth prediction method may function properly at image areas which are bounded by edges which are not parallel to the epipolar line. At its current status, this makes the depth prediction method a *near-field* approach, which is suited to closer objects. However, the depth prediction method can utilize any other depth cue, which can provide 3D line orientation at the edges of objects, and it should perform better if there are other cues available.

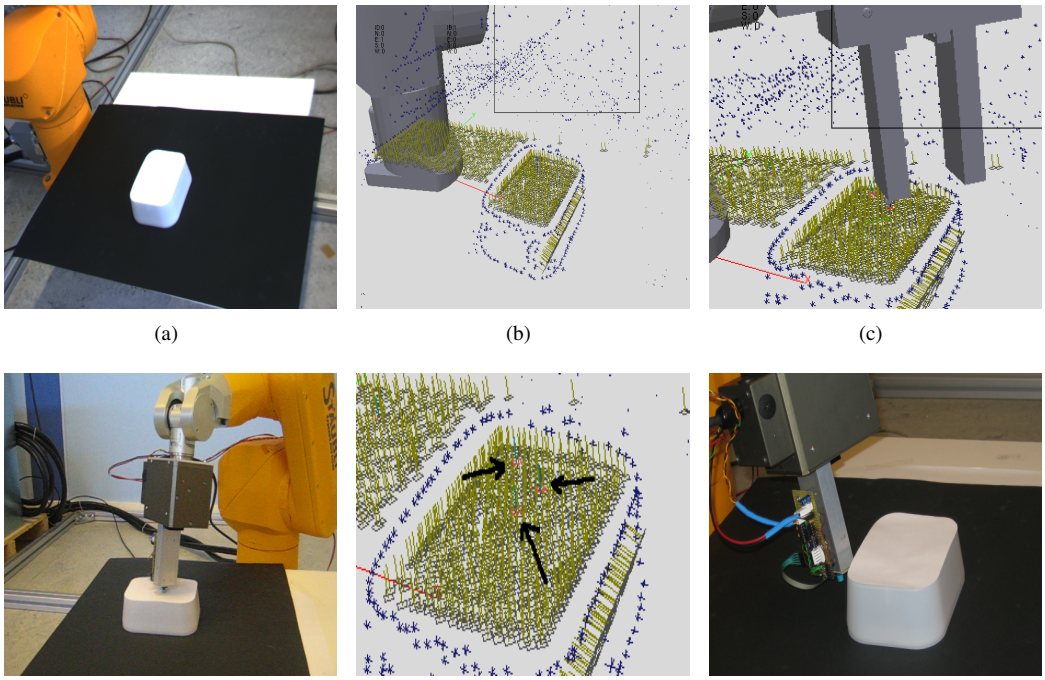
A missing part of the proposed method is handling of occlusions: Occluding edge features should cast votes only on the occluding image areas. Otherwise, as shown in figure 7.14, the occluding edge pulls the predictions in the occluded surface towards the edge. Occlusions can be detected using the motion or dense stereo matching; however, these methods are limited to textured surfaces, and since we are interested in weakly-textured scenes, they would not be applicable.

We used a different set of images than, for example, the Middleburry database [Middlebury, 2007] because the parameters of the cameras are not provided in this database, and without these parameters, it is not possible to reconstruct 3D information from the disparities at the edges, and therefore, our method is not applicable.

7.6.7 Integration into a multi-sensorial framework

As motivated in chapter 1, depth extraction should be understood as a learning problem. For this purpose, a system should be able to touch, grasp and play with objects in its environment to construct their 3D models. Along these lines, [Kjargaard et al., 2007] used the touch sensors on a robot arm to validate the predictions made by our model. Moreover, the surface normal and the 3D position suggested by the touch sensors are added as separate predictions which can be combined to improve the existing depth predictions. Once the current depth prediction method is extended to carry two best hypotheses instead of taking the best one, the robot arm can interact with the vision system to make a decision between the hypothesis. It is likely that babies also use their haptic information to refine their visual perception.

Figure 7.25 shows the process on an example. The depth model predicts three surfaces, one at the top and at the side of the box, and one between the box and the edge of the black surface (figure 7.25(b)). The robot arm can verify the predictions at the top of the box (figure 7.25(c)) and the verifications are added as new predictions (shown in red square in 7.25(e)). In figure 7.25(f), the robot arm disconfirms the predictions made by the occluding edge of the box and the edge of the black surface.



(a) (b) (c)

(d) (e) (f)

Figure 7.25: Surface Verification Experiment (taken from [Kjargaard et al., 2007]). **(a)** Setup of the scene. **(b)** View of the 3 predicted surfaces. **(c)** The robot moving in position to verify the surface on the box. **(d)** The sensor in contact with the surface on the box. **(e)** The 3 detected haptic primitives shown as small red squares. **(f)** The robot moving through the wrongly predicted surface without detecting a contact.

7.7 Conclusion

The current chapter has introduced a voting model that estimates the depth at homogeneous or weakly-textured image patches from the depth of the bounding edge-like structures. The depth at edge-like structures is computed using a feature-based stereo algorithm, and is used to vote for the depth of a mono, which otherwise is not possible to compute easily due to the correspondence problem. The results have been compared with different dense stereo algorithms in order to state that our feature-based algorithm works well for scenes that dense stereo algorithms are not suited. However, our aim is not to claim that our approach is better but rather to suggest that the different approaches are suited for different image contexts and that a combination of them is necessary. A naive combination of the different approaches as a proof of concept is provided to show that such a combination would benefit from both approaches and would be able to work in textured as well as non-textured image areas.

Depth prediction from surrounding edge features should be regarded as one depth cue which utilizes the 3D information at the edges. The utilization of the edges for depth prediction is along the lines of 3D surface interpretation from line drawings of objects [Barrow and Tenenbaum, 1981]. The proposed method can be extended, for example, by making use of depth discontinuities and orientation discontinuities differently since they are indications of different 3D information, as suggested in [Barrow and Tenenbaum, 1981].

One quality of the model is that it can be improved to regard the best two clusters as two different depth hypotheses at each mono, and the model can be modified *not* to make a decision between them until it can pass the hypotheses to a higher-level process which can make the decision. One example for such a high-level process is demonstrated in section 7.6.7.

As motivated in chapter 1, depth prediction can be understood as a feedback mechanism which completes the missing information in early vision. This makes depth prediction a part of an early cognitive vision framework where different cues interact with each other to remove the ambiguities and the missing information in early vision.

We are planning to combine the depth prediction method with dense stereo methods in a *feedback mechanism*. In this mechanism, using high-resolution cameras with a built-in region of interest facility, it is possible to capture image regions at low and high resolutions. At the low resolution, the texture on a surface might be very weak, which favors the utilization of the depth prediction method. At the high resolution, the texture on a surface can become sufficiently distinguishable for stereo matching, which favors dense stereo methods. Based on these observations, we propose to use the depth prediction at the low resolution, and then zoom in to the region of interest to get more signal information (*i.e.*texture detail) from the surface, and verify or refine the original depth predictions using the disparity estimation from dense methods. Such a system can be considered as an attention mechanism where the details are acquired by attending to the regions of interest.

7.8 Acknowledgements

The publications of the author which are relevant for this chapter are [Kalkan et al., 2007b, Kalkan et al., 2007a, Kjargaard et al., 2007, Kraft et al., 2007, Başeski et al., 2007, Kalkan et al., 2008].

Chapter 8

Conclusions

The current chapter concludes the thesis in the following two sections with a summary and an outlook of the contributions.

8.1 Summary

Extraction of different modalities and processing of local image structures are limited, ambiguous and incomplete, as argued in chapter 1. Biological vision systems can cope with such ambiguities and the missing information by:

1. exploiting the redundancy of information in the natural images, which are accessible through the statistical properties of the visual entities,
2. using feedback information from higher visual levels and
3. using lateral feedback information between different visual modalities, for example, in the form of an interpolation process.

Note that these issues overlap; *i.e.*, utilization of one issue may make use of another. This thesis addressed the above mentioned issues in the context of an early cognitive vision system:

- In chapter 3, the extent of the *problem of local processing* (*i.e.*, the aperture problem) in the case of optical flow estimation is investigated using different optic flow estimation algorithms on natural images.

- Junction detection methods are usually biased in positioning junctions [Deriche and Giraudon, 1993, Rohr, 1992]. Moreover, they have the trade-off between the completeness of the detections and the amount of false positives (*i.e.*, spurious detections). In chapter 4, a junction *regularity* measure, called intersection consistency is proposed to improve the positioning of junctions, and the semantic interpretation of junctions is used as a *feedback* mechanism for removing outliers and selecting reliable junction detections.

- In chapter 5, the relation between local image structures and local 3D structure is investigated. The results of this investigation are important for understanding the possible mechanisms underlying *depth interpolation* processes.

- In chapter 6, the investigations in chapter 5 is extended using higher order relations between local 3D structures. The results of this investigation provide insights into *depth interpolation* mechanisms and can be used as priors in a depth prediction model.

- In chapter 7, motivated from the results of chapters 5 and 6, a voting-based depth prediction model that predicts depth at homogeneous image areas is proposed. This model utilizes the sparse local 3D features extracted using a feature based stereo, and its performance is extensively compared against several dense stereo methods. Such a model can be regarded as a lateral feedback between the edge features over long distances that are extracted in early vision to complete the missing information at homogeneous image patches using *depth interpolation*.

  The contribution of chapter 7 is the proposal of a depth cue that exploits the redundancy of information in images. Currently, the depth cue makes use of the 3D information computed using stereo; however, it can work with other depth cues such as structure from motion as long as 3D positions and 3D line orientations are provided.

The thesis utilizes the concept of intrinsic dimensionality in all the chapters for detecting local image structures. Especially in chapter 3 for analyzing the quality of optic flow estimation, and in chapter 5 for investigating the relation between local image structures and local 3D structures, intrinsic dimensionality proves to be a useful tool that can make thorough analysis and make explicit manifestations about properties of local image features, which are otherwise more difficult to observe.

The tools developed in the thesis have become part of an early cognitive vision framework [Pugeault et al., 2006] mainly developed in the European ECOVISION project that otherwise makes use of only edge-like structures. This work has been supported by the European Drivsco [Drivsco, 2007] and PACO-PLUS [PACO-PLUS, 2007] projects.

8.2 Outlook

Depth extraction makes use of monocular or multi-view depth cues in order to recover the third dimension from a set of images. This ill-posed inverse problem is challenging since each depth cue bears ambiguities, or is reliable only for certain types of scenes. Moreover, how the different cues should be integrated or fused together is itself a difficult and an open question because different cues might carry conflicting interpretations of a scene. Because of these reasons, current computer vision algorithms or applications are limited to only certain types of scenes and are not general.

The experiments with babies suggest that depth cues which are not directly based on correspondences evolve rather late in the development of the human visual system. For example, pictorial depth cues are made use of only after approximately 6 months [Kellman and Arterberry, 1998]. This indicates that experience may play an important role in the development of these cues, *i.e.*, that we have to understand depth perception as a statistical learning problem [Knill and Richards, 1996, Purves and Lotto, 2002, Rao et al., 2002], where attention and the utilization of statistical regularities play an important role.

In view of the above mentioned problems, the following need to be tackled in order to build a biologically-motivated fully-functional machine vision system:

1. Which visual abilities and depth cues are humans *equipped* with at birth, and what abilities and cues do we *learn* and in what sequence, if there is an ordering between different depth cues? A more important question is, of course, “how do we do it?”

2. How do we experiment with the world to (1) build representations of objects, to (2) exploit the statistical regularities of natural scenes, and to (3) integrate the information from other senses such as haptic and sound.

The current thesis is relevant to these questions, and any extension in view of these questions are valuable contributions to the field. To name a few:

- Investigation of the *unification* of different examples of feedback mechanisms that allows exchange of information between different visual information at different levels of visual processing.
- Learning of depth cues with as little prior information as possible using statistical regularities in natural scenes.
- Using vision to build a database of object models, and building mechanisms for providing context to the visual processing through attention or other feedback mechanisms.

Appendix A

Algorithmic Details of Intrinsic Dimensionality

This chapter provides the algorithmic details of intrinsic dimensionality which can be used to implement it. There are two different ways to compute *iD*:

1. As proposed in [Felsberg and Krüger, 2003, Krüger and Felsberg, 2003], which computes the origin and line variance explicitly to compute the coordinates of a signal in the *iD* triangle.

2. As proposed in [Felsberg et al., 2007a], which implicitly computes the origin and line variance by mapping the magnitude and orientation of signals to a cone.

The first approach is slower than the second one; therefore, the second method is used in this thesis. For this reason, only the second approach is detailed here; the interested reader is directed to [Felsberg and Krüger, 2003, Krüger and Felsberg, 2003] for the first approach.

In the cone model, the coordinates are constructed from the magnitude m and the orientation θ. Averaging the coordinates locally inside the cone implicitly computes the line variance.

The overall algorithm for an image point $\mathbf{u} = (u_1, u_2)$ is as follows:

1. *Gradient information:* Extract the (complex) gradient data $g = m(\mathbf{u}) \exp(i\theta(\mathbf{u}))$, m being the magnitude and θ the orientation at pixel $\mathbf{u}$.

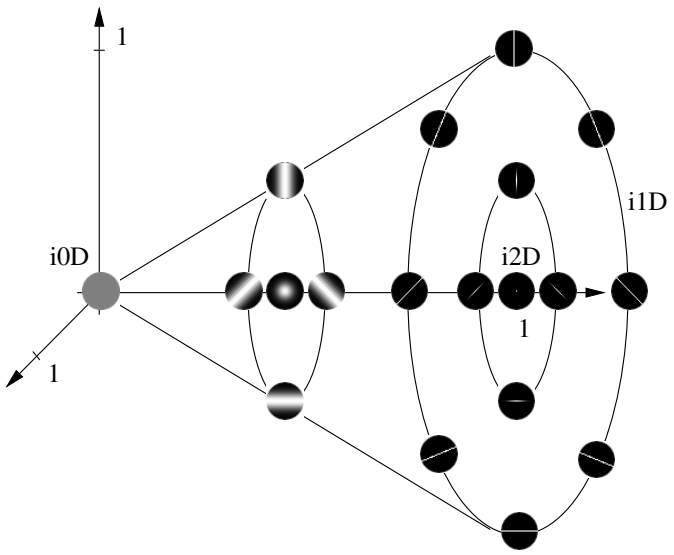


Figure A.1: The cone constructed from the magnitude and the orientation of signals. Taken from [Felsberg et al., 2007a].

2. *Magnitude normalization and double angle representation:* Convert the gradient data to soft-thresholded double angle representation $d(\mathbf{u}) = s_1(m(\mathbf{u}))\exp(i2\theta(\mathbf{u}))$, $s_1(\cdot)$ being the soft threshold function.

3. *Cone representation:* Set the cone coordinates $\mathbf{c}(\mathbf{u}) = (c_1, c_2, c_3) = (|d|, \mathrm{Re}\{d\}, \mathrm{Im}\{d\})$. The cone is exemplified in figure A.1. For different real examples, the cone coordinates of the points in the patch are shown in figure A.2.

4. *Averaging:* Average the cone coordinates locally: $\mathbf{c}'(\mathbf{u}) = \sum_{\mathbf{i}} w_{\mathbf{i}}\mathbf{c}(\mathbf{i})$ where $\mathbf{i}$ runs over the neighborhood of $\mathbf{u}$, and $w_{\mathbf{i}}$ is the two dimensional Gaussian with appropriate σ. In our implementation, *sigma* is set to be $\sqrt{l/4}$, l being the patchsize.

5. *Triangle representation:* $(x^{\triangle}(\mathbf{u}), y^{\triangle}(\mathbf{u})) = (c'_1, \sqrt{(c'_2)^2 + (c'_3)^2})$

6. *Normalization of y values (optional):* Set $(\hat{x}(\mathbf{u}), \hat{y}(\mathbf{u})) = (x^{\triangle}(\mathbf{u}), s_2(x^{\triangle}(\mathbf{u}), y^{\triangle}(\mathbf{u})))$ where s_2 is a monotonic transform to spread the data more uniformly, mainly for the purpose of visualization.

7. *Barycentric coordinates:* Extract barycentric coordinates from $(\hat{x}, \hat{y})$ according to equation (2.1).

The soft-threshold function s_1 and monotonic transform s_2 to spread the y values are as follows:

- The soft-threshold function $s_1 : \mathrm{R}^+ \rightarrow [0,1) : m \mapsto s_1(m)$ maps the unbounded magnitudes to a bounded interval. Basically, we can make use of any activation function used in neural networks (see, *e.g.*, [Bishop, 1995]) such as the logarithmic sigmoid function. However, we must adjust one constant in order to get an appropriate mapping. Our choice is:

$$s_1(m) = \tanh(\alpha m), \tag{A.1}$$

  where α is a parameter controlling the dynamics of m (figure A.3(a)). This parameter can be estimated such that the empirical distribution of $|d|$ follows as good as possible a particular predefined distribution, *e.g.*, a uniform distribution. For the following experiments, we computed α such that the mean $\bar{m}$ of the magnitudes is mapped to the empirically chosen value 0.35: $\alpha = \operatorname{atanh}(0.35)/\bar{m}$.

  An alternative soft-threshold function can be normalization by the maximum magnitude in the image:

$$s_1(m) = \frac{m}{m_{max}}, \tag{A.2}$$

  where m_{max} is the maximum magnitude in the image.

- We apply the mapping s_2 to obtain $\hat{y}$ for ensuring a reasonable spread of representations between i1D and i2D, *i.e.*, we want to ensure that corners are mapped close to the i2D vertex while edge–like structures are mapped close to the i1D vertex. This is mainly for visualization and interpretation purposes and in practice one can omit this mapping. The subsequent illustrations were generated with the mapping $s_2(x', y') = x'(y'/x')^\beta$ with $\beta = 5$.

  The normalization function y^β for $\beta = 5$ is shown in figure A.3(b).

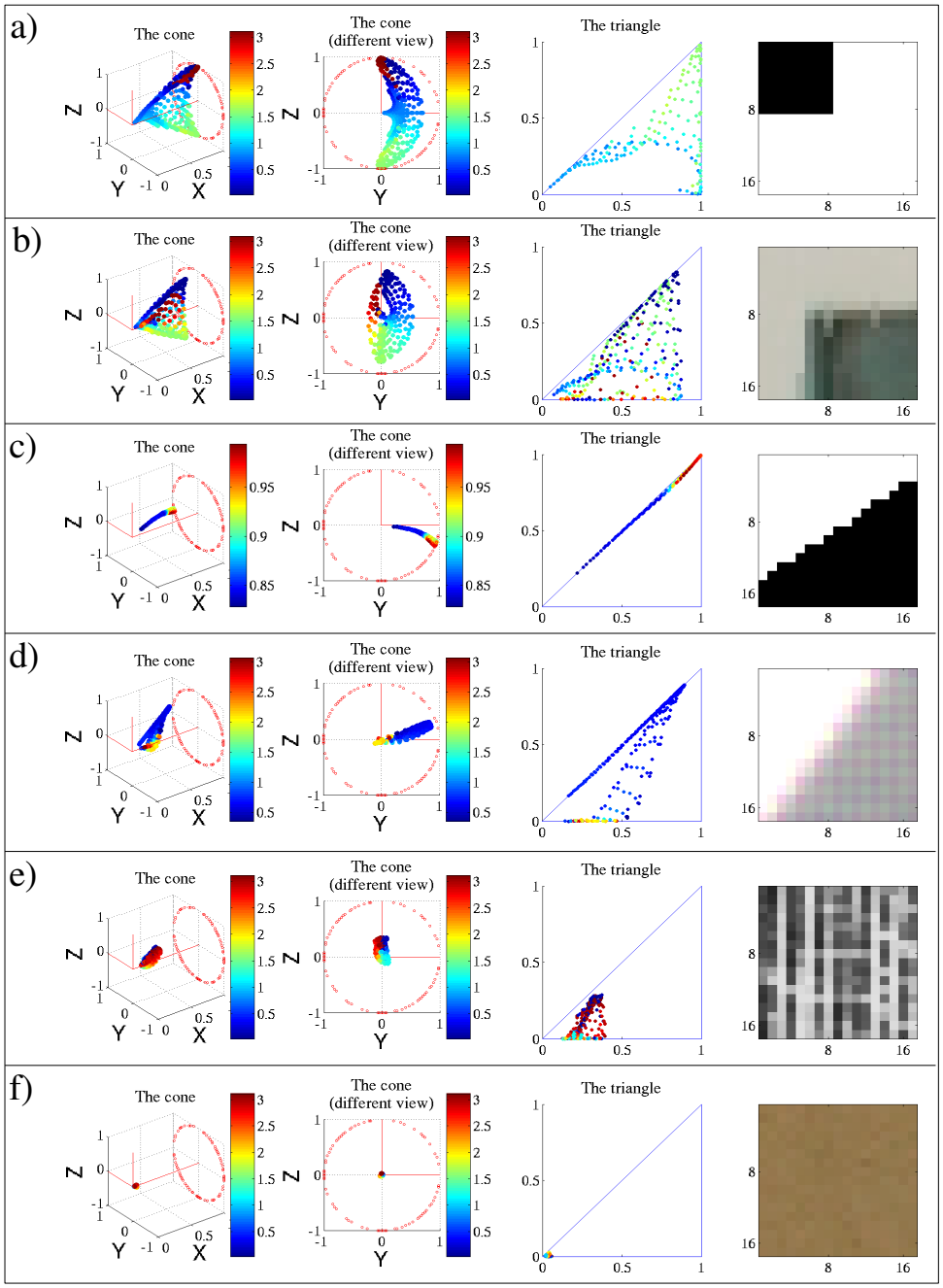
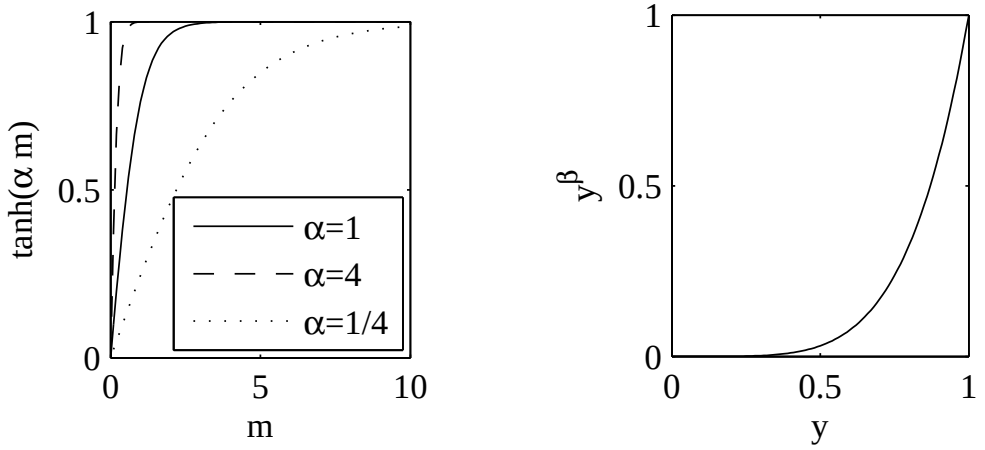


Figure A.2: Illustration of how the points in some image patches taken from real scenes map to the triangle and the cone. The patches are illustrated in the right-most column. The color of the points inside the triangle and the cone encode different orientations, whose values can be accessed using the color-bars.



(a) Soft-threshold function. (b) Spread function.

Figure A.3: (a) Function $s_1(m) = \tanh(\alpha m)$ for soft-thresholding the magnitude with different values of α. (b) Spread function, y^β for $\beta = 5$.

Appendix B

Grouping 2D Primitives

This chapter describes the perceptual grouping relations that are used to group 2D primitives into contours. As the primitives are local contour descriptors, scene contours are expected to be represented by strings of primitives that are locally close to collinear. In the following, we will explain methods for grouping 2D primitives into contours.

In the following, $c(l_{i,j})$ refers to the likelihood for two primitives π_i and π_j to be *linked*: i.e. grouped to describe the same contour.

Position and orientation of primitives are intrinsically related. As primitives represent local edge estimators, their positions are points along the edge, and their orientation can be seen as a tangent at such a point. The estimated likelihood of the contour described by those tangents is based upon the assumption that simpler curves are more likely to describe the scene structures, and highly jagged contours are more likely to be manifestations of erroneous and noisy data.

Therefore, for a pair of primitives π_i and π_j in image $\boldsymbol{I}$, we can formulate the likelihood for these primitives to describe the same contour as a combination of three basic constraints on their relative position and orientation — see [Pugeault et al., 2006].

B.1 Proximity ($c_p[l_{i,j}]$)

A contour is more likely if it is described by a dense population of primitives. Large holes in the primitive description of the contour is an indication that there are two contours which are collinear yet different.

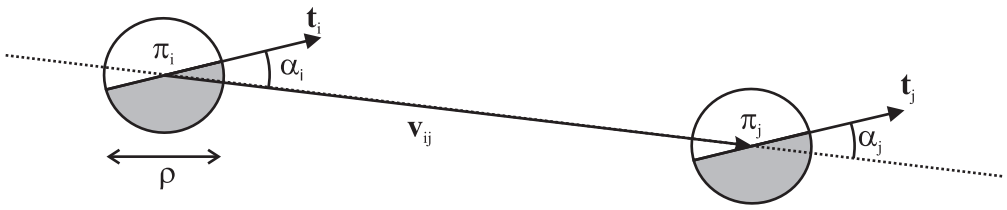


Figure B.1: Illustration of the values used for the collinearity computation. If we consider two primitives $\boldsymbol{\pi}_i$ and $\boldsymbol{\pi}_j$, then the vector between the centres of these two primitives is written $\boldsymbol{v}_{ij}$, and the orientations of the two primitives are designated by the vectors t_i and t_j, respectively. The angle formed by $\boldsymbol{v}_{ij}$ and t_i is written α_i, and between $\boldsymbol{v}_{ij}$ and t_j is written α_j. ρ is the radius of the image patch used to generate the primitive.

The proximity constraint is defined by the following equation:

$$c_p[l_{i,j}] = 1 - e^{-\max\left(1-\frac{\|\mathbf{v_{i,j}}\|}{\rho\tau},0\right)}, \tag{B.1}$$

where ρ stands for the size of the receptive field of the primitives in pixels; $\rho\tau$ is the size of the neighbourhood considered in pixels; and, $\|\mathbf{v_{i,j}}\|$ is the distance in pixels separating the centres of the two primitives.

B.2 Collinearity ($c_{co}[l_{i,j}]$)

A contour is more likely to be linear, or to form a shallow curve rather than a sharp one. A sharp curve might be an indication of two intersecting or occluding contours.

$$c_{co}[l_{i,j}] = 1 - \left|\sin\left(\frac{|\alpha_i| + |\alpha_j|}{2}\right)\right|, \tag{B.2}$$

where α_i and α_j are the angles between the line joining the two primitives centres and the orientation of, respectively, π_i and π_j.

B.3 Co–circularity ($c_{ci}[l_{i,j}]$)

A contour is more likely to have a continuous, or smoothly changing curvature, rather than a varying one. An unstable curvature is an indicator of a noisy, erroneous or under–sampled contour, all of which are

unreliable.

$$c_{ci}[l_{i,j}] = 1 - \left|\sin\left(\frac{\alpha_i + \alpha_j}{2}\right)\right|, \tag{B.3}$$

B.4 Geometric Constraint ($\mathbf{G}_{i,j}$)

The combination of those three criteria provided above forms the following *geometric* affinity measure:

$$\mathbf{G}_{i,j} = \sqrt[3]{c_e[l_{i,j}] \cdot c_{co}[l_{i,j}] \cdot c_{ci}[l_{i,j}]}, \tag{B.4}$$

where $\mathbf{G}_{i,j}$ is the geometric affinity between two primitives π_i and π_j. This affinity represents the likelihood that two primitives π_i and π_i are part of an actual contour of the scene.

B.5 Multi–modal Constraint ($\mathbf{M}_{i,j}$)

The geometric constraint offers a suitable estimation of the likelihood of the curve described by the pair of primitives. Other modalities of the primitives 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, and thus consistency constraints can also be enforced over those modalities. Effectively, the less difference there is between the modalities of two primitives, the more likely that they are expressions of the same contour. In [Elder and Goldberg, 1998], it is already proposed that the intensity can be used as a cue for perceptual grouping; our definition goes beyond this proposal by using a combination of the phase, colour and optical flow modalities of the primitives to decide if they describe the same contour:

$$\mathbf{M}_{i,j} = w_\omega c_\omega[l_{i,j}] + w_{\boldsymbol{c}} c_{\boldsymbol{c}}[l_{i,j}] + w_f c_f[l_{i,j}], \tag{B.5}$$

where c_ω is the phase criterion, $c_{\boldsymbol{c}}$ the colour criterion and c_f the optical flow criterion. Each of the three w_ω, $w_{\boldsymbol{c}}$ and w_f is the relative scaling for each modality, with $w_\omega + w_{\boldsymbol{c}} + w_f = 1$.

B.6 Primitive Affinity ($A_{i,j}$)

The overall affinity between all primitives in an image is formalized as a matrix $\boldsymbol{A}$, where $\boldsymbol{A}_{i,j}$ holds the affinity between the primitives π_i and π_j. We define this affinity from equations B.4 and B.5, such that (1) two primitives complying poorly with the good continuation rule have an affinity close to zero; and (2) two primitives complying with the good continuation rule yet strongly dissimilar will have only an average affinity. The affinity is formalised as follows:

$$c(l_{i,j}) = \boldsymbol{A}_{i,j} = \sqrt{\mathbf{G}\left(\alpha \mathbf{G}_{i,j} + (1-\alpha)\mathbf{M}_{i,j}\right)}, \tag{B.6}$$

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

B.7 Acknowledgements

This appendix is adapted from [Kalkan et al., 2007a] using the contributions of Nicolas Pugeault.

Appendix C

Computation of an Ellipse and the Definition of Coplanarity

The current chapter provides (1) the details of how the parameters of an ellipse are computed, and (2) the definition of coplanarity between two planar patches in 3D. These details are relevant for chapter 6.

C.1 Parameters of an ellipse

Let us denote the position of two 3D edges π_1^e, π_2^e by $(\mathbf{X}_{2D})_1$ and $(\mathbf{X}_{2D})_2$ respectively. The vectors between the 3D edges and IP (let us call l_1 and l_2) can be defined as:

$$\begin{aligned} l_1 &= ((\mathbf{X}_{2D})_1 - IP), \\ l_2 &= ((\mathbf{X}_{2D})_2 - IP). \end{aligned} \tag{C.1}$$

Having defined l_1 and l_2, the ellipse $E(\pi_1^e, \pi_2^e)$ is as follows:

$$E(\pi_1^e, \pi_2^e) = \begin{cases} f_1 = (\mathbf{X}_{2D})_1, f_2 = (\mathbf{X}_{2D})_1', b = |l_2| & \text{if } |l_1| > |l_2|, \\ f_1 = (\mathbf{X}_{2D})_2, f_2 = (\mathbf{X}_{2D})_2', b = |l_1| & \text{otherwise.} \end{cases} \tag{C.2}$$

where $(\mathbf{X}_{2D})'$ is symmetrical with $\mathbf{X}_{2D}$ around the intersection point and on the line defined by $\mathbf{X}_{2D}$ and IP (as shown in figure 6.4(e)).

C.2 Definition of coplanarity

Let π^s denote either a semi-edge π^{se} or a mono π^m. Two π^s are coplanar iff they are on the same plane. When it comes to measuring coplanarity, two criteria need to be tested:

1. Angular criterion: For two π^s to be coplanar, the angular difference between the orientation of the planes that represent them should be less than a threshold. A situation is illustrated in figure C.1(a) where angular criterion holds but the planes are not coplanar.

2. Distance-based criterion: For two π^s to be coplanar, the distance between the center of the first π^s and the plane defined by the other π^s should be less than a threshold. In figure C.1(b), B and C are at the same distance to the plane P which is the plane defined by the planar patch A. However, C is more distant to the center of A than B, and in this paper, we treat that C is more coplanar to A than B is to A. The reason for this can be clarified with an example: Assume that A, B and C are all parallel, and that the *planar* and the Euclidean distances between A and B are both D units, and between A and C are respectively D and $n \times D$. It is straightforward to see that although B and C have the same planar distances to A, for $n >> 1$, C should have a higher coplanarity measure.

It is sufficient to combine these two criteria as follows:

$$\begin{aligned} cop(\pi^s_1, \pi^s_2) \quad = \quad & \alpha(\mathbf{p}^{\pi^s_1},\ \mathbf{p}^{\pi^s_2}) < T_p \text{ AND} \\ & d(\mathbf{p}^{\pi^s_1},\ \pi^s_2)/d(\pi^s_1,\ \pi^s_2) < T_d, \end{aligned} \tag{C.3}$$

where $\mathbf{p}^{\pi^s}$ is the plane associated to π^s; $\alpha(\mathbf{p}_1, \mathbf{p}_2)$ is the angle between the orientations of $\mathbf{p}_1$ and $\mathbf{p}_2$; and, $d(.,.)$ is the Euclidean distance between two entities.

In our analysis, we have empirically chosen T_p and T_d as 20 degrees and 0.5, respectively. These parameters are determined by testing the coplanarity measure over different samples. T_p is the limit for angular separation between two planar patches. Bigger values would relax the coplanarity measure, and vice versa. T_d restricts the distances between the patches; in analogy to T_p, T_d can be used to relax the

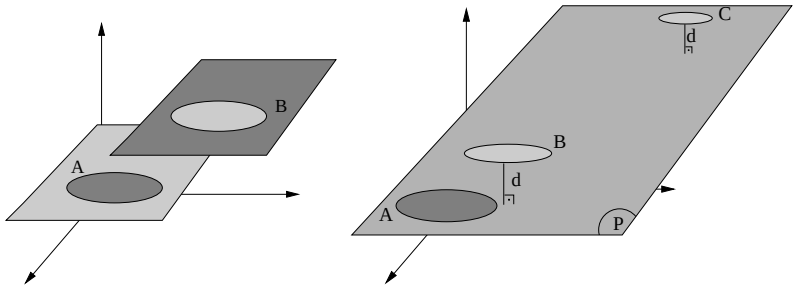


Figure C.1: Criteria for coplanarity of two planes. (a) According to the angular-difference criterion of coplanarity, entities A and B will be measured as coplanar although they are on different planes. In (b), P is the plane defined by entity A. According to the distance-based coplanarity definition, entities B and C have the same measure of coplanarity. However, entity C which is more distant to entity A should have a higher measure of coplanarity than entity B although they have the same distance to plane P (see the text).

coplanarity measure. As shown in figure 6.7 for a stricter coplanarity definition (with T_p and T_d set to 10 degrees and 0.2), different values for these thresholds would quantitatively but not qualitatively change the results presented in chapter 6.

Bibliography

[Aloimonos and Shulman, 1989] Aloimonos, Y. and Shulman, D. (1989). *Integration of Visual Modules — An extension of the Marr Paradigm*. Academic Press, London. 13, 120

[Alvarez et al., 2000] Alvarez, L., Weickert, J., and Sanchez, J. (2000). Reliable estimation of dense optical flow fields with large displacements. *International Journal of Computer Vision*, 39:41–56. 36

[Anderson et al., 2002] Anderson, B. L., Singh, M., and Fleming, R. W. (March 2002). The interpolation of object and surface structure. *Cognitive Psychology*, 44:148–190(43). 11, 16

[Angelucci et al., 2002] Angelucci, A., Levitt, J. B., Walton, E. J. S., Hupe, J.-M., Bullier, J., and Lund, J. S. (2002). Circuits for Local and Global Signal Integration in Primary Visual Cortex. *J. Neurosci.*, 22(19):8633–8646. 15

[Asada and Brady, 1986] Asada, H. and Brady, M. (1986). The curvature primal sketch. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(1):2–14. 49

[Başeski et al., 2007] Başeski, E., Pugeault, N., Kalkan, S., Kraft, D., Wörgötter, F., and Krüger, N. (2007). A scene representation based on multi-modal 2d and 3d features. *3D Representation for Recognition Workshop (in conjunction with ICCV)*. 21, 123

[Baker et al., 1998] Baker, S., Nayar, S. K., and Murase, H. (1998). Parametric feature detection. *Int. Journal of Computer Vision*, 27(1):27–50. March. 48

[Baker et al., 2001] Baker, S., Sim, T., and Kanade, T. (2001). A characterization of inherent stereo ambiguities. In *Int. Conf. on Computer Vision (ICCV)*, volume 1, page 428. 15, 16

[Barron et al., 1994] Barron, J., Fleet, D., and Beauchemin, S. (1994). Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77. 30, 36, 40

[Barrow and Tenenbaum, 1981] Barrow, H. G. and Tenenbaum, J. M. (1981). Interpreting line drawings as three-dimensional surfaces. *Artificial Intelligence*, 17:75–116. 11, 16, 58, 72, 90, 91, 94, 106, 123

[Bayerl and Neumann, 2007] Bayerl, P. and Neumann, H. (2007). Disambiguating visual motion by form–motion interaction — a computational model. *International Journal of Computer Vision*, 72(1):27–45. 12, 15, 29, 31

[Beaudet, 1978] Beaudet, P. (1978). Rotationally invariant image operators. In *Proc.4th Int. Joint Conf. Pattern Recognition*, pages 579–583. Kyoto, Japan. 49

[Bertero et al., 1987] Bertero, M., Poggio, T., and Torre, V. (1987). Ill-posed problems in early vision. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA. 10

[Biederman, 1987] Biederman, I. (1987). Recognition by components: A theory of human image understanding. *Psychological Review*, 94(2). 105

[Bishop, 1995] Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press, New York. 130

[Bolle and Vemuri, 1991] Bolle, R. M. and Vemuri, B. C. (1991). On three-dimensional surface reconstruction methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(1):1–13. 60

[Brown et al., 2003] Brown, M. Z., Burschka, D., and Hager, G. D. (2003). Advances in computational stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(8):993–1008. 108, 109

[Bruce et al., 2003] Bruce, V., Green, P. R., and Georgeson, M. A. (2003). *Visual Perception: Physiology, Psychology and Ecology*. Psychology Press, 4th edition. 14, 93

[Brunswik and Kamiya, 1953] Brunswik, E. and Kamiya, J. (1953). Ecological cue–validity of 'proximity' and of other Gestalt factors. *American Journal of Psychologie*, LXVI:20–32. 11, 18, 73, 86

[Bullier, 2001] Bullier, J. (2001). Integrated model of visual processing. *Brain Research Reviews*, 36:96–107(12). 15

[Calow et al., 2004] Calow, D., Krüger, N., Wörgötter, F., and Lappe, M. (2004). Statistics of optic flow for self-motion through natural scenes. *Proc. Dynamic Perception Workshop*, pages 133–138. 38

[Cavanagh and Mather, 1989] Cavanagh, P. and Mather, G. (1989). Motion: the long and the short of it. *Spatial Vision*, 4:103–129. 30

[Clerc and Mallat, 2002] Clerc, M. and Mallat, S. (2002). The texture gradient equation for recovering shape from texture. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):536–549. 92

[Coello, 1999] Coello, C. A. C. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems*, 1(3):129–156. 47

[Collett, 1985] Collett, T. S. (1985). Extrapolating and Interpolating Surfaces in Depth. *Royal Society of London Proceedings Series B*, 224:43–56. 11, 16

[Coppola et al., 1998] Coppola, D. M., Purves, H. R., McCoy, A. N., and Purves, D. (1998). The distribution of oriented contours in the real world. *PNAS*, pages 4002–4006. 35, 44

[Coxeter, 1969] Coxeter, H. (1969). *Introduction to Geometry (2nd ed.)*. Wiley & Sons. 23

[Deriche and Giraudon, 1990] Deriche, R. and Giraudon, G. (1990). Accurate corner detection: An analytical study. *ICCV 90, Osaka Japan*. 49

[Deriche and Giraudon, 1993] Deriche, R. and Giraudon, G. (1993). A computational approach for corner and vertex detection. *IJCV*, 10(2):101–124. 46, 49, 54, 55, 125

[Dreschler and Nagel, 1982] Dreschler, L. and Nagel, H. H. (1982). Volumetric model and 3d trajectory of a moving car derived from monocular tv frame sequences of a street scene. *Computer Graphics and Image Processing*, 20:199–228. 49

[Drivsco, 2007] Drivsco (2007). Learning to emulate perception-action cycles in a driving school scenario, european project ist-fp6-fet-016276-2, http://www.pspc.dibe.unige.it/drivsco/. Last access: 16.05.2007. 12, 126

[ECOVISION, 2003] ECOVISION (2003). Artificial visual systems based on early-cognitive cortical processing (EU–Project). *http://www.pspc.dibe.unige.it/ecovision/project.html*. 12

[Elder and Goldberg, 1998] Elder, H. and Goldberg, R. (1998). Inferential reliability of contour grouping cues in natural images. *Perception Supplement*, 27. 135

[Elder and Goldberg, 2002] Elder, H. and Goldberg, R. (2002). Ecological statistics of gestalt laws for the perceptual organization of contours. *Journal of Vision*, 2(4):324–353. 18, 73, 86

[Elder et al., 2003] Elder, J. H., Krupnik, A., and Johnston, L. A. (2003). Contour grouping with prior models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(25):1–14. 11, 18, 73, 86

[Esteban and Schmitt, 2004] Esteban, C. H. and Schmitt, F. (2004). Silhouette and stereo fusion for 3d object modeling. *Comput. Vis. Image Underst.*, 96(3):367–392. 95

[Faugeras, 1993] Faugeras, O. (1993). *Three–Dimensional Computer Vision*. MIT Press. 16, 92, 95, 101

[Felsberg et al., 2007a] Felsberg, M., Kalkan, S., and Krüger, N. (2007a). Continuous characterization of image structures of different dimensionality. *Image and Vision Computing (submitted).* 23, 128, 129

[Felsberg et al., 2007b] Felsberg, M., Kalkan, S., and Krüger, N. (2007b). Continuous dimensionality characterization of image structures. *(submitted to)Image and Vision Computing*. 20, 21, 22, 23, 25

[Felsberg and Krüger, 2003] Felsberg, M. and Krüger, N. (2003). A probablistic definition of intrinsic dimensionality for images. *Pattern Recognition, 24th DAGM Symposium*. 22, 23, 26, 128

[Fermueller et al., 2001] Fermueller, C., Shulman, D., and Aloimonos, Y. (2001). The statistics of optical flow. *Computer Vision and Image Understanding*, 82:1–32. 32

[Field, 1994] Field, D. (1994). What is the goal of sensory coding? *Neural Computation*, 6(4):561–601. 18

[Field et al., 1993] 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. 18, 75

[Fleet and Jepson, 1990] Fleet, D. J. and Jepson, A. D. (1990). Computation of component image velocity from local phase information. *International Journal of Computer Vision*, 5:77–104. 36

[Forstner, 1994] Forstner, W. (1994). A framework for low level feature extraction. In *ECCV '94: Proceedings of the third European conference on Computer Vision (Vol. II)*, pages 383–394, Secaucus, NJ, USA. Springer-Verlag New York, Inc. 48, 49, 52

[Gallant et al., 1994] Gallant, J. L., Essen, D. C. V., and Nothdurft, H. C. (1994). *Early Vision and Beyond*, chapter : Two-dimensional and three-dimensional texture processing in visual cortex of the macaque monkey, pages 89–98. MA: MIT Press. 17

[Galuske et al., 2002] Galuske, R. A. W., Schmidt, K. E., Goebel, R., Lomber, S. G., and Payne, B. R. (2002). The role of feedback in shaping neural representations in cat visual cortex. *Proceedings of the National Academy of Science*, 99:17083–17088. 15

[Gautama and Hulle, 2002] Gautama, T. and Hulle, M. M. V. (2002). A phase-based approach to the estimation of the optical flow field using spatial filtering. *IEEE Transactions on Neural Networks*, 13(5):1127–1136. 31, 36, 37

[Geisler et al., 2001] 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. 11, 18

[Grimson, 1982] Grimson, W. E. L. (1982). A Computational Theory of Visual Surface Interpolation. *Royal Society of London Philosophical Transactions Series B*, 298:395–427. 11, 16, 17, 93, 94, 95

[Grimson, 1983] Grimson, W. E. L. (1983). Surface consistency constraints in vision. *Computer Vision, Graphics and Image Processing*, 24(1):28–51. 58, 70, 72

[Grimson, 1984] Grimson, W. E. L. (1984). Binocular shading and visual surface reconstruction. *Computer Vision, Graphics, and Image Processing*, 28(1):19–43. 17, 93

[Grimson, 1993] Grimson, W. E. L. (1993). Why stereo vision is not always about 3d reconstruction. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA. 93

[Guy and Medioni, 1994] Guy, G. and Medioni, G. (1994). Inference of surfaces from sparse 3-d points. In *ARPA94*, pages II:1487–1494. 17, 94, 95

[Guzman, 1968] Guzman, A. (1968). Decomposition of a visual scene into three-dimensional bodies. *AFIPS Fall Joint Conference Proceedings*, 33:291–304. 14

[Hadamard, 1923] Hadamard, J. (1923). *Lectures on the Cauchy Problem in Linear Partial Differential Equations*. Yale, New Haven. 10

[Hahn and Krüger, 2000] Hahn, M. and Krüger, N. (2000). Junction detection and semantic interpretation using Hough lines. *EIS'2000*. 48

[Harris and Stephens, 1988] Harris, C. G. and Stephens, M. J. (1988). A combined corner and edge detector. In *Proc. Fourth Alvey Vision Conference, Manchester*, pages 147–151. 49, 50

[Hartley and Zisserman, 2000] Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision.* Cambridge University Press. 92

[Hoff and Ahuja, 1989] Hoff, W. A. and Ahuja, N. (1989). Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(2):121–136. 94

[Hoover et al., 1996] Hoover, A., Jean-Baptiste, G., Jiang, X., Flynn, P. J., Bunke, H., Goldgof, D. B., Bowyer, K., Eggert, D. W., Fitzgibbon, A., and Fisher, R. B. (1996). An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689. 60

[Horaud and Veillon, 1990] Horaud, R. and Veillon, F. (1990). Finding geometric and relational structures in an image. In *ECCV 90*, pages 374–384, New York, NY, USA. Springer-Verlag New York, Inc. 49

[Howe and Purves, 2002] Howe, C. Q. and Purves, D. (2002). Range image statistics can explain the anomalous perception of length. *PNAS*, 99(20):13184–13188. 19, 59

[Howe and Purves, 2004] Howe, C. Q. and Purves, D. (2004). Size contrast and assimilation explained by the statistics of natural scene geometry. *Journal of Cognitive Neuroscience*, 16(1):90–102. 19, 59

[Huang et al., 2000] Huang, J., Lee, A. B., and Mumford, D. (2000). Statistics of range images. *CVPR*, 1(1):1324–1331. 18, 40, 59, 68

[Hubel and Wiesel, 1969] Hubel, D. and Wiesel, T. (1969). Anatomical demonstration of columns in the monkey striate cortex. *Nature*, 221:747–750. 14, 17, 24

[Johnston and Clifford, 1995] Johnston, A. and Clifford, C. W. G. (1995). A unified account of three apparent motion illusions. *Vision Research*, 35(8):1109–1123. 30

[Jones and Palmer, 1987] 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. 18

[Julesz, 1971] Julesz, B. (1971). *Foundations of Cyclopean Perception*. Univ. of Chicago Press, Chicago, IL. 11, 16

[Kalkan et al., 2004a] Kalkan, S., Calow, D., Felsberg, M., Worgotter, F., Lappe, M., and Krueger, N. (2004a). Optic flow statistics and intrinsic dimensionality. *Proc. of Brain Inspired Cognitive Systems, Scotland, available at http://www.cs.stir.ac.uk/ lss/BICS2004/CD/toc.html*. 20, 21, 40, 45

[Kalkan et al., 2004b] Kalkan, S., Calow, D., Wörgötter, F., Lappe, M., and Krüger, N. (2004b). Local image structures and optic flow estimation. *Proc. Dynamic Perception Workshop*, pages 233–238. 20, 21, 45

[Kalkan et al., 2005] 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. 20, 21, 23, 45, 71

[Kalkan et al., 2007a] Kalkan, S., Pugeault, N., and Krüger, N. (2007a). Perceptual operations and relations between 2d or 3d visual entities. Technical Report 2007-3, Robotics Group, Maersk Institute, University of Southern Denmark. 21, 123, 136

[Kalkan et al., 2006] Kalkan, S., Wörgötter, F., and Krüger, N. (2006). Statistical analysis of local 3d structure in 2d images. *CVPR*, 1:1114–1121. 19, 20, 21, 23, 74, 93, 94

[Kalkan et al., 2007b] Kalkan, S., Wörgötter, F., and Krüger, N. (2007b). Depth prediction at homogeneous image structures. Technical Report 2007-2, Robotics Group, Maersk Institute, University of Southern Denmark. 21, 96, 123

[Kalkan et al., 2007c] Kalkan, S., Wörgötter, F., and Krüger, N. (2007c). First-order and second-order statistical analysis of 3d and 2d structure. *Network: Computation in Neural Systems (in press)*. 20, 21, 74, 87

[Kalkan et al., 2007d] Kalkan, S., Wörgötter, F., and Krüger, N. (2007d). Statistical analysis of second-order relations of 3d structures. *International Conference on Computer Vision Theory and Applications (VISAPP)*. 20, 21, 87

[Kalkan et al., 2007e] Kalkan, S., Wörgötter, F., and Krüger, N. (2007e). Statistical analysis of second-order relations of 3d structures. *Int. Conference on Computer Vision Theory and Applications (VISAPP)*. 89, 99

[Kalkan et al., 2008] Kalkan, S., Wörgötter, F., and Krüger, N. (2008). Depth prediction at homogeneous image structures. *(submitted to) Int. Conf. on Computer Vision Theory and Applications (VISAPP)*. 21, 123

[Kalkan et al., 2007f] Kalkan, S., Yan, S., Pilz, F., and Krüger, N. (2007f). Improving junction detection by semantic interpretation. *International Conference on Computer Vision Theory and Applications (VISAPP)*. 20, 21

[Kang et al., 2001] Kang, K., Tarel, J.-P., Fishman, R., and Cooper, B. D. (2001). A linear dual-space approach to 3D surface reconstruction from occluding contours using algebraic surface. In *International Conference on Computer Vision*, volume 1, pages 198–204. 95

[Kellman and Arterberry, 1998] Kellman, P. and Arterberry, M., editors (1998). *The Cradle of Knowledge*. MIT-Press. 17, 19, 93, 126

[Kjargaard et al., 2007] Kjargaard, M., Bierbaum, A., Kraft, D., Kalkan, S., Krüger, N., Asfour, T., and Dillmann, R. (2007). Using tactile sensors for multisensorial scene explorations. Technical Report 2007-5, Robotics Group, Maersk Institute, University of Southern Denmark. 21, 121, 122, 123

[Knill and Richards, 1996] Knill, D. C. and Richards, W., editors (1996). *Perception as bayesian inference*. Cambridge: Cambridge University Press. 17, 73, 86, 93, 126

[Koenderink and Dorn, 1982] Koenderink, J. and Dorn, A. (1982). The shape of smooth objects and the way contours end. *Perception*, 11:129—173. 14

[Kraft et al., 2007] Kraft, D., Başeski, E., Popovic, M., Krüger, N., Pugeault, N., Kragic, D., Kalkan, S., and Wörgötter, F. (2007). Birth of the object: Detection of objectness and extraction of object shape through object action complexes. *(submitted to) International Journal of Humanoid Robotics*. 21, 123

[Krueger, 1998] Krueger, N. (1998). Collinearity and parallelism are statistically significant second order relations of complex cell responses. *Neural Processing Letters*, 8(2):117–129. 11, 18, 35, 44, 73, 86

[Krüger and Felsberg, 2003] Krüger, N. and Felsberg, M. (2003). A continuous formulation of intrinsic dimension. *Proceedings of the British Machine Vision Conference*. 22, 23, 34, 48, 128

[Krüger and Felsberg, 2004] 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. 24

[Krüger et al., 2004a] Krüger, N., Felsberg, M., and Wörgötter, F. (2004a). Processing multi-modal primitives from image sequences. *Fourth International ICSC Symposium on ENGINEERING OF INTELLIGENT SYSTEMS*. 33

[Krüger et al., 2003] Krüger, N., Lappe, M., and Wörgötter, F. (2003). Biologically motivated multi-modal processing of visual primitives. *Proc. the AISB 2003 Symposium on Biologically inspired Machine Vision, Theory and Application, Wales*, pages 53–59. 16, 78

[Krüger et al., 2004b] Krüger, N., Lappe, M., and Wörgötter, F. (2004b). Biologically motivated multi-modal processing of visual primitives. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, 1(5). 11, 24, 88

[Krüger et al., 2007] Krüger, N., Pugeault, N., and Wörgötter, F. (2007). Multi-modal primitives: Local, condensed, and semantically rich visual descriptors and the formalization of contextual information. *(submitted to) IEEE Pattern Analysis and Machine Intelligence*. 15, 24

[Krüger and Wörgötter, 2002] 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. 18

[Krüger and Wörgötter, 2004] 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. 18, 19, 44, 45

[Krüger and Wörgötter, 2005] Krüger, N. and Wörgötter, F. (2005). Multi-modal primitives as functional models of hyper-columns and their use for contextual integration. *Proc. 1st Int. Symposium on Brain, Vision and Artificial Intelligence, Naples, Italy, Lecture Notes in Computer Science, Springer, LNCS 3704*, pages 157–166. 78

[Lappe et al., 1999] Lappe, M., Bremmer, F., and van den Berg, A. V. (1999). Perception of self-motion from visual flow. *Trends in Cognitive Sciences*, 3:329–336. 38

[Laycock and Day, 2006] Laycock, R. G. and Day, A. M. (2006). Image registration in a coarse three-dimensional virtual environment. *Computer Graphics Forum*, 25(1):69–82. 72

[Lee and Medioni, 1998] Lee, M. S. and Medioni, G. (1998). Inferring segmented surface description from stereo data. In *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 346. 17, 95

[Lee et al., 2002] Lee, M.-S., Medioni, G., and Mordohai, P. (2002). Inference of segmented overlapping surfaces from binocular stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(6):824–837. 17, 94, 95

[Lee et al., 1998] Lee, T. S., Mumford, D., Romero, R., and Lamme, V. A. F. (1998). The role of the primary visual cortex in higher level vision. *Vision Research*, 38:2429–2454. 17

[Liu et al., 2007] Liu, X., Yao, H., and Gao, W. (2007). Shape from silhouette outlines using an adaptive dandelion model. *Computer Vision and Image Understanding*, 105(2):121–130. 95

[Lucas and Kanade, 1981] Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. *Proc. DARPA Image Understanding Workshop*, pages 121–130. 31, 35

[Malik, 1987] Malik, J. (1987). Interpreting line drawings of curved objects. *International Journal of Computer Vision*, 1:73–103. 14

[Marr, 1982] Marr, D. (1982). *Vision: A computational investigation into the human representation and processing of visual information*. Freeman. 12, 14

[Matsumoto et al., 1999] Matsumoto, Y., Fujimura, K., and Kitamura, T. (1999). Shape-from-silhouette/stereo and its application to 3-d digitizer. In *DCGI '99: Proceedings of the 8th International Conference on Discrete Geometry for Computer Imagery*, pages 177–190, London, UK. Springer-Verlag. 95

[Middlebury, 2007] Middlebury (2007). Middlebury stereo database. `http://vision.middlebury.edu/stereo`. 121

[Moravec, 1980] Moravec, H. (1980). Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical Report CMU-RI-TR-3, Carnegie-Mellon University, Robotics Institute. 49

[Mota and Barth, 2000] Mota, C. and Barth, E. (2000). On the uniqueness of curvature features. *Proc. in Artificial Intelligence*, 9:175–178. 30, 45

[Nagel and Enkelmann, 1986] 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. 24, 31, 36

[Nagel and Haag, 1998] Nagel, H.-H. and Haag, M. (1998). Bias-corrected optical flow estimation for road vehicle tracking. *Proc. International Conference on Computer Vision, Bombay, India*, pages 1006–1011. 32

[Nalwa, 1989] Nalwa, V. S. (1989). Line-drawing interpretation: Bilateral symmetry. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(10):1117–1120. 94, 106

[Noble, 1989] Noble, A. (1989). *Descriptions of Image Surfaces*. PhD thesis, Dept. of Engineering Science, Oxford University. p45. 54

[Olshausen and Field, 1996] Olshausen, B. and Field, D. (1996). Natural image statistics and efficient coding. *Network*, 7:333–339. 73, 86

[PACO-PLUS, 2007] PACO-PLUS (2007). Perception, action and cognition through learning of object-action complexes, european project ist-fp6-ip-027657, http://www.paco-plus.org. Last access: 16.05.2007. 12, 126

[Papathomas et al., 1995] Papathomas, T. V., Chubb, C., Gorea, A., and Kowler, E., editors (1995). *Early vision and beyond*. Massachusetts Institute of Technology, Cambridge, MA, USA. 10

[Parida et al., 1998] Parida, L., Geiger, D., and Hummel, R. (1998). Junctions: detection, classification and reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(7):687–698. 46, 48, 49, 52

[Pilz et al., 2007] Pilz, F., Yan, S., Grest, D., Pugeault, N., Kalkan, S., and Krüger, N. (2007). Utilizing semantic interpretation of junctions for 3d-2d pose estimation. *Proc. of International Symposium on Visual Computing (ISVC), California, USA*. 20, 21, 46

[Potetz and Lee, 2003] Potetz, B. and Lee, T. S. (2003). Statistical correlations between two-dimensional images and three-dimensional structures in natural scenes. *Journal of the Optical Society of America*, 20(7):1292–1303. 18, 59

[Princen et al., 1990] Princen, J., Illingworth, J., and Kittler, J. (1990). An optimizing line finder using a Hough transform algorithm. *Computer Vision, Graphics, and Image Processing*, 52:57–77. 30

[Pugeault et al., 2008] Pugeault, N., Kalkan, S., Başeski, E., Wörgötter, F., and Krüger, N. (2008). Reconstruction uncertainity and 3d relations. *(submitted to) Int. Conference on Computer Vision Theory and Appliations (VISAPP)*. 96, 120

[Pugeault and Krüger, 2003] Pugeault, N. and Krüger, N. (2003). Multi–modal matching applied to stereo. *Proceedings of the BMVC 2003*. 17, 88, 89, 95, 96

[Pugeault et al., 2004] Pugeault, N., Krüger, N., and Wörgötter, F. (2004). A non-local stereo similarity based on collinear groups. *Proceedings of the Fourth International ICSC Symposium on Engineering of Intelligent Systems*. 11, 18, 73, 86

[Pugeault et al., 2006] Pugeault, N., Wörgötter, F., , and Krüger, N. (2006). Multi-modal scene reconstruction using perceptual grouping constraints. In *Proceedings of the 5th IEEE Computer Society Workshop on Perceptual Organization in Computer Vision,New York City June 22, 2006 (in conjunction with IEEE CVPR 2006)*. 11, 12, 16, 24, 100, 126, 133

[Purves and Lotto, 2002] Purves, D. and Lotto, B., editors (2002). *Why we see what we do: an empirical theory of vision*. Sunderland, MA: Sinauer Associates. 17, 73, 86, 93, 126

[Ragheb and Hancock, 2002] Ragheb, H. and Hancock, E. (2002). A probabilistic framework for specular shape from shading. *Pattern Recognition*, 36:407–427. 91, 92

[Rao et al., 2002] Rao, R. P. N., Olshausen, B. A., and Lewicki, M. S., editors (2002). *Probabilistic models of the brain*. MA: MIT Press. 17, 73, 86, 93, 126

[Robles-Kelly and Hancock, 2004] Robles-Kelly, A. and Hancock, E. R. (2004). A graph-spectral approach to shape-from-shading. *IEEE Transactions on Image Processing*, 13(7):912–926. 92

[Rohr, 1992] Rohr, K. (1992). Recognizing corners by fitting parametric models. *International Journal of Computer Vision*, 9(3):213–230. 46, 48, 49, 54, 55, 125

[Rosenhahn, 2003] Rosenhahn, B. (2003). *Pose Estimation Revisited (PhD Thesis)*. Institut für Informatik und praktische Mathematik, Chrsitian–Albrechts–Universität Kiel. 44

[Rosenhahn and Sommer, 2002] Rosenhahn, B. and Sommer, G. (2002). Adaptive pose estimation for different corresponding entities. In van Gool, L., editor, *Pattern Recognition, 24th DAGM Symposium*, pages 265–273. Springer Verlag. 45

[Rubin, 2001] Rubin, N. (2001). The role of junctions in surface completion and contour matching. *Perception*, 30:339–366. 14

[Sabatini et al., 2007] Sabatini, S. P., Gastaldi, G., Solari, F., Diaz, J., Ros, E., Pauwels, K., Hulle, K. M. M. V., Pugeault, N., and Krüger, N. (2007). Compact and accurate early vision processing in the harmonic space. *International Conference on Computer Vision Theory and Applications (VISAPP), Barcelona*. 108, 109, 110

[Scharstein and Szeliski, 2001] Scharstein, D. and Szeliski, R. (2001). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Technical Report MSR-TR-2001-81, Microsoft Research, Microsoft Corporation. 108, 109, 110, 115, 116

[Schmid et al., 2000] Schmid, C., Mohr, R., and Bauckhage, C. (2000). Evaluation of interest point detectors. *Int. Journal of Computer Vision*, 37(2):151–172. 49

[Sereno et al., 2002] Sereno, M. E., Trinath, T., Augath, M., and Logothetis, N. K. (2002). Three-dimensional shape representation in monkey cortex. *Neuron*, 33(4):635–652. 17

[Shevelev et al., 2003] Shevelev, I. A., Kamenkovich, V. M., and Sharaev, G. A. (2003). The role of lines and corners of geometric figures in recognition performance. *Acta Neurobiol Exp*, 63(4):361–368. 14

[Shevelev et al., 1998] Shevelev, I. A., Lazareva, N. A., Sharaev, G. A., Novikova, R. V., and Tikhorimov, A. S. (1998). Selective and invariant sensitivity to crosses and corners in cat striate neurons. *Neuroscience*, 84:713–721. 14, 17

[Shevlin, 1998] Shevlin, F. (1998). Analysis of orientation problems using Plücker lines. *International Conference on Pattern Recognition, Bisbane*, 1:65–689. 45

[Shirai, 1987] Shirai, Y. (1987). *Three-dimensional computer vision*. Springer-Verlag New York, Inc. 60, 62

[Simoncelli and Farid, 1996] Simoncelli, E. and Farid, H. (1996). Steerable wedge filters for local orientation analysis. *IEEE Trans Image Proc*, 5(9):1377–1382. 48

[Simoncelli, 2003] Simoncelli, E. P. (2003). Vision and the statistics of the visual environment. *Current Opinion in Neurobiology*, 13(2):144–149. 11

[Simoncelli et al., 1991] Simoncelli, E. P., Adelson, E. H., and Heeger, D. J. (1991). Probability distributions of optical flow. *Proc. IEEE Conference on Computer Vision and Pattern Recognition, Maui, Hawaii*, pages 310–315. 32

[Smith and Brady, 1997] Smith, S. and Brady, J. (1997). SUSAN - a new approach to low level image processing. *Int. Journal of Computer Vision*, 23(1):45–78. 49, 54

[Smith, 1997] Smith, S. M. (1997). Reviews of optic flow, motion segmentation, edge finding and corner finding. Technical Report TR97SMS1, Oxford University. 49

[Spelke, 1993] Spelke, E. (1993). Principles of object perception. *Cognitive Science*, 14:29–56. 19

[Stevens, 1981] Stevens, K. A. (1981). The visual interpretations of surface contours. *Artificial Intelligence*, 17:47–73. 94, 106

[Terzopoulos, 1982] Terzopoulos, D. (1982). Multi-level reconstruction of visual surfaces: Variational principles and finite element representations. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA. 17, 95

[Terzopoulos, 1988] Terzopoulos, D. (1988). The computation of visible-surface representations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(4):417–438. 11, 16, 17, 95

[Torres-Mendez and Dudek, 2006] Torres-Mendez, L. A. and Dudek, G. (2006). Statistics of visual and partial depth data for mobile robot environment modeling. *Mexican International Conference on Artificial Intelligence (MICAI)*. 72

[Treue et al., 1995] Treue, S., Andersen, R. A., Ando, H., and Hildreth, E. C. (1995). Structure-from-motion: perceptual evidence for surface interpolation. *Vision Research*, 35(1):139–48. 11, 16

[Tuceryan and Jain, 1998] Tuceryan, M. and Jain, N. K. (1998). Texture analysis. *The Handbook of Pattern Recognition and Computer Vision (2nd Edition)*, pages 207–248. 14

[Ulupinar and Nevatia, 1991] Ulupinar, F. and Nevatia, R. (1991). Constraints for interpretation of line drawings under perspective projection. *CVGIP: Image Underst.*, 53(1):88–96. 94, 106

[Ulupinar and Nevatia, 1993] Ulupinar, F. and Nevatia, R. (1993). Perception of 3-d surfaces from 2-d contours. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15(1):3–18. 94, 106

[van Diepen and Graef, 1994] van Diepen, P. M. J. and Graef, P. D. (1994). Line-drawing library and software toolbox. Technical Report 165, Laboratory of Experimental Psychology, University of Leuven, Belgium. 91

[Waltz, 1975] Waltz, D. (1975). Understanding line drawings of scenes with shadows. In Winston, I. P., editor, *The Psychology of Computer Vision*, pages 19–91. New York: McGRaw–Hill. 46, 52

[Wegmann and Zetzsche, 1990] Wegmann, B. and Zetzsche, C. (1990). Statistical dependence between orientation filter outputs used in a human-vision-based image code. In *Proc. SPIE Vol. 1360, p. 909-923, Visual Communications and Image Processing '90: Fifth in a Series, Murat Kunt; Ed.*, pages 909–923. 34

[Wörgötter et al., 2004] 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*, 3:293–321. 10, 15

[Yan, 2005] Yan, S. (2005). Model-based corner detector - positioning and semantic interpretation. *Final Project at Engineering College of Copenhagen*. 55

[Yang and Purves, 2003] Yang, Z. and Purves, D. (2003). Image/source statistics of surfaces in natural scenes. *Network: Computation in Neural Systems*, 14:371–390. 18, 59, 68, 86

[Zetzsche and Barth, 1990] Zetzsche, C. and Barth, E. (1990). Fundamental limits of linear filters in the visual processing of two dimensional signals. *Vision Research*, 30(7):1111–1117. 22

[Zetzsche et al., 1991] Zetzsche, C., Barth, E., and Berkmann, J. (1991). Spatio-temporal curvature measures for flow field analysis. *Geometric Methods in Computer Vision*, 1570:337–350. 30

[Zhu, 1999] Zhu, S. C. (1999). Embedding gestalt laws in markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1170–1187. 11, 18, 73, 86

Index

Curriculum Vitae

Name	Sinan Kalkan
Birth date	Sept. 1979
Birth place	Ankara (Turkey)
Nationality	Turkish

1996–1997	English Prep. Course, Middle East Technical University (METU)
1997–2001	B. Sc. in Dept. of Computer Eng., METU
2001–2003	M. Sc. in Dept. of Computer Eng., METU
2001–2003	Research Assistant in Dept. of Computer Eng., METU
2003–2005	Ph. D. Student in Computational Neuroscience, Uni. of Stirling
2005–2007	Ph. D. Student in Informatics Dept., Uni. of Göttingen (continuing the studies started in Stirling)
2005–2007	Research Assistant in *Bernstein Center for Computational Neuroscience*, Uni. of Göttingen