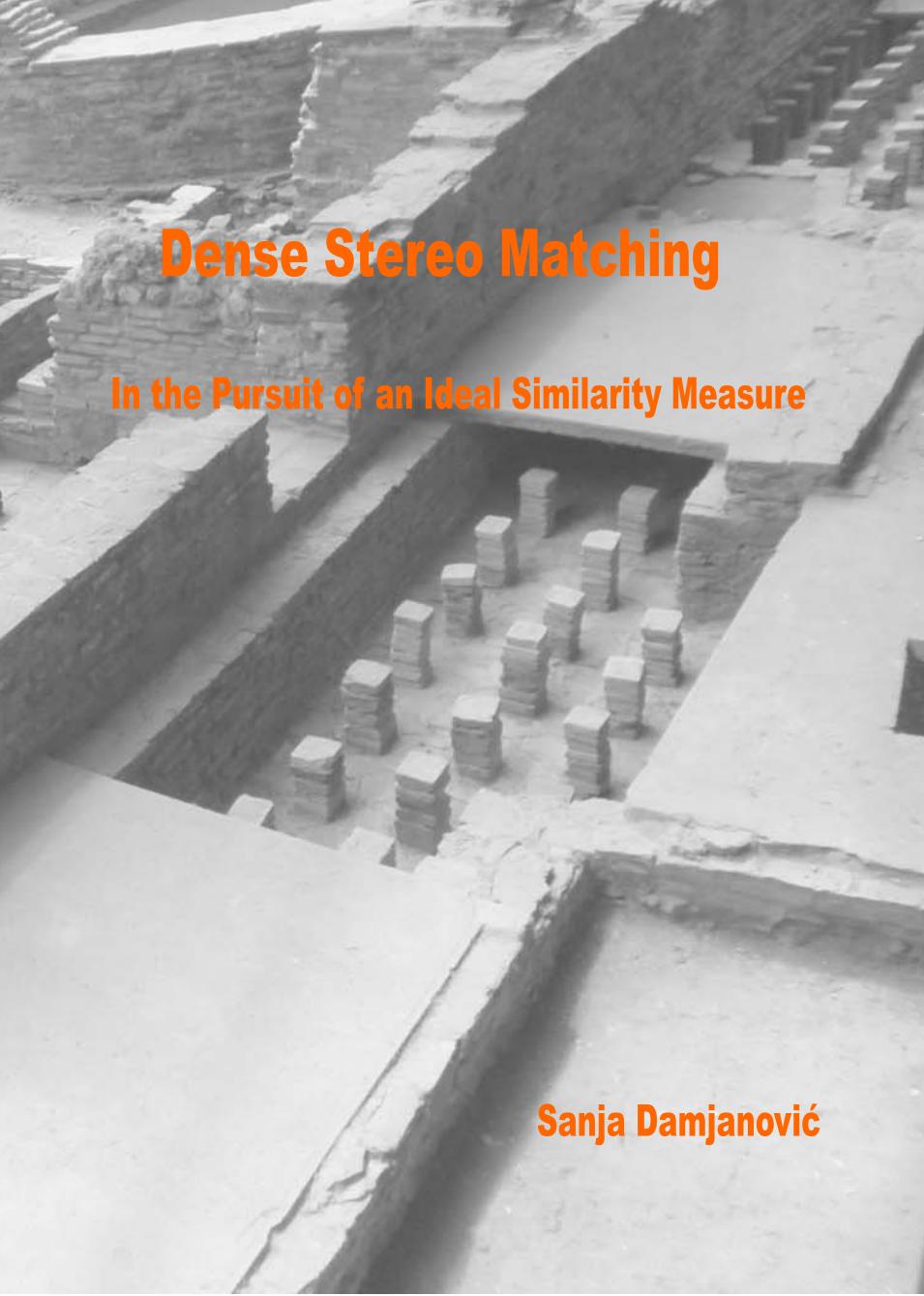


Dense Stereo Matching

In the Pursuit of an Ideal Similarity Measure

Sanja Damjanović

DENSE STEREO MATCHING

IN THE PURSUIT OF AN IDEAL SIMILARITY MEASURE

Sanja Damjanović

De promotiecommissie:

voorzitter en secretaris:	
prof.dr.ir. Mouthaan	Universiteit Twente
promotor:	
prof.dr.ir. C.H. Slump	Universiteit Twente
assistent promotors:	
dr.ir. L.J. Spreeuwers	Universiteit Twente
dr.ir. F. van der Heijden	Universiteit Twente
leden:	
prof.dr. J.C.T. Eijkel	Universiteit Twente
prof.dr. P.H. Hartel	Universiteit Twente
prof.dr.ir. P.H.N. de With	Technische Universiteit Eindhoven
prof.dr. V. Evers	Universiteit Twente
prof.dr.ir. J. Top	Vrije Universiteit Amsterdam

CTIT Ph.D. Thesis Series No. 12-234
Centre for Telematics and Information Technology
P.O. Box 217, 7500 AE
Enschede, The Netherlands.

CTIT

Signals & Systems group,
EEMCS Faculty, University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands.

Printed by Wöhrmann Print Service, Zutphen, The Netherlands.
Typesetting with LaTeX2e.
Image on the cover shows Roman Imperial Palace built in the 3^{rd} century AC in former Sirmium, now Sremska Mitrovica, Serbia.

ISBN 978-90-365-3456-7
ISSN 1381-3617 (CTIT Ph.D.-thesis serie No. 12-234)
DOI 10.3990/1.9789036534567
http://dx.doi.org/10.3990/1.9789036534567

DENSE STEREO MATCHING

IN THE PURSUIT OF AN IDEAL SIMILARITY MEASURE

PROEFSCHRIFT

ter verkrijging van
de graad van doctor aan de Universiteit Twente,
op gezag van de rector magnificus,
prof. dr. H. Brinksma,
volgens besluit van het College voor Promoties
in het openbaar te verdedigen
op donderdag 8 november 2012 om 12.45 uur

door

Sanja Damjanović
geboren op 2 Mei 1976
te Sremska Mitrovica, Servië

Dit proefschrift is goedgekeurd door de promotor:

Prof.dr.ir. C.H. Slump

en de assistent promotors:

dr.ir. L.J. Spreeuwers
dr.ir. F. van der Heijden

This thesis is dedicated to Milan

Contents

1 Introduction

In this chapter we introduce the stereo matching, a common research topic within the computer vision. In addition, we describe the stereo vision system, introduce relevant terminology, and define our research questions. Lastly, we present the outline of the thesis.

1.1 Stereo Vision

The human vision system process visual information effortlessly and can determine how far away objects are, how they are oriented with respect to the viewer, and how they relate to other objects. Computer vision is a field that includes methods for acquiring, processing, analysing, and understanding images, scene reconstruction, event detection, video tracking, object recognition, learning, indexing, motion estimation, and image restoration [1].

Computer vision seeks to model the complex visual world by various mathematical methods including physics-based and probabilistic models. The task of computer vision is difficult one because it tries to solve an inverse problem and seeks to recover some unknowns given insufficient information to fully specify the solution.

One of the aims of computer vision is to describe the world that we see in one or more images and to reconstruct its properties, such as shape, illumination, and color distributions. Stereo vision is a field within computer vision, that deals with an important problem: reconstruction of the three-dimensional coordinates of points in scene given two camera-produced images of known camera geometry and orientation [2].

1.2 Stereo Matching

Binocular stereo is a problem of determining the three-dimensional shape of visible surfaces in a static scene from two images of the same scene taken by two cameras or one camera at two different positions. The central task of binocular stereo is to solve a correspondence problem, i.e. to find pairs of corresponding points in the images. Corresponding points are projections onto images of the same scene point. Stereo matching is a method which aims to solve the correspondence problem [3], [4].

When the camera parameters and geometry are known, the problem can be transformed to a one-dimensional problem. Stereo matching then finds corresponding points along the epipolar lines in both images and their relative displacements. The map of all relative displacements is called a disparity map and with known geometry this can easily be transformed into a depth map.

Undistorted and rectified stereo images serve as the starting point in stereo matching. The geometry of the cameras is thus known, and the images are transformed to correspond to a non-verged stereo system, i.e. a stereo system with cameras with parallel optical axes as shown in Figure 1.1. Cameras are modeled by the *projective pinhole camera model* with an image plain at

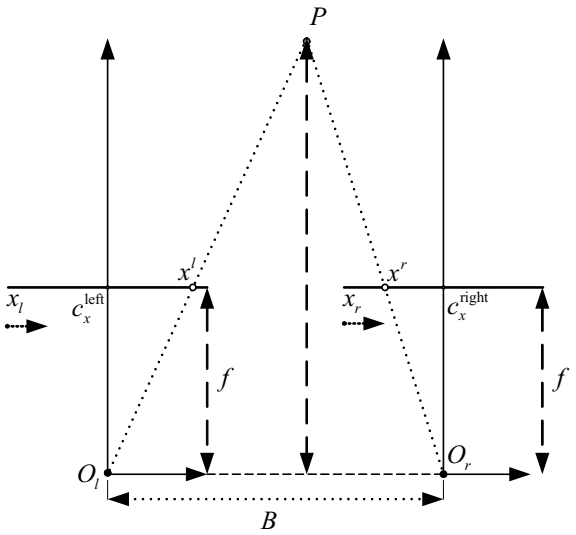


Figure 1.1 Ideal stereo geometry

distance f with respect to a projection center [5]. A crossection of a non-verged stereo camera system is illustrated in Figure 1.1: two cameras with parallel optical axes $\overline{O_l c_x^{left}}$ and $\overline{O_r c_x^{right}}$ at a baseline distance B and with equal focal lengths $f_l = f_r$. Also the principal points c_x^{left} and c_x^{right} have the same pixel coordinates in their respective left and right images.

In such setup the epipolar lines are known, horizontal and aligned. We then assume we can find a point P in the physical world in the left and right images, denoted as p_l and p_r in Figure 1.1. Points p_l and p_r are called corresponding points. In this simplified case, taking x^l and x^r to be the horizontal positions of the points in the left and the right image, p_l and p_r respectively, we can calculate the depth Z of point P if the disparity between image points p_l and p_r is known. Thus, if the disparity as defined by

$$d = x^l - x^r, \tag{1.1}$$

is known, the depth of point P is calculated as

$$Z = \frac{f \cdot B}{x^l - x^r}. \tag{1.2}$$

The first step is to match the points in the two images along the known epipolar lines and to determine their disparities given by equation (1.1), so

that the three-dimensional position of each point can be determined by triangulation given by equation (1.2).

Although the mathematical model and the explanation of stereo vision are simple, stereo matching is often ambiguous for photometric issues, surface structure or geometric ambiguities. The pivoting point of nearly all stereo correspondence algorithms is photometric constancy, i.e. it is assumed that different images of the same scene have the same appearance. But this is not always true. For highly reflective or specular surfaces, the appearances in different images differ significantly. Also, finding corresponding points within uniformly colored regions or surfaces with repetitive texture or structure is problematic. Next, depending on the scene geometry, it can happen that some points in one image do not have corresponding points in the other image due to occlusion or due to the limited field of view.

The starting point in stereo correspondence involves many assumptions and constraints. Although stereo has been a scientific topic of interest since more than half a century, not all questions have been answered and not all problems solved.

1.3 Terminology

The aim of stereo matching is to find, in a reference stereo image, the corresponding point for each pixel in a non-reference stereo image. We introduce the terminology of the stereo correspondence problem on the rectified stereo pair *Tsukuba* from the Middlebury benchmark [6].

The first row in Figure 1.2 shows the rectified stereo pair *Tsukuba*. The left image of the stereo pair is considered as the reference while the second row shows the color coded ground truth disparity map. Disparity ranges from 0 to 15. The actual minimum disparity of the scene is 5; this is coded by light blue. The background of the scene is furthest from the cameras; it has the minimum disparity. The lamp is the object in the scene closest to the cameras and has the largest disparity 14. The third row in Figure 1.2 shows the nonoccludded, occluded and discontinuity regions in gray, black, and white respectively, for the reference image of the stereo pair. Black, except for the image boundary, represents pixels in the left image that do not have corresponding pixels in the right image because they are not visible in the right image, i.e. they are occluded in that image. White represents regions with disparity or equivalently depth discontinuity. In a discontinuity region, the disparity changes abruptly and significantly, i.e. more than one pixel along the epipolar line. Discontinuity regions are rather challenging for an accurate

correspondence calculation.

To solve the stereo correspondence, a template matching method can be used [1]. The template in stereo matching can be a squared window or a segment. The region around a pixel in the reference image is compared to the potential matching regions in the other, non-reference stereo image. To determine which pixel from the candidate pixels from the disparity range is the corresponding one, it is necessary to have a suitable score for template comparison. This score can be expressed as similarity measure, likelihood and cost.

No matter how good a similarity measure, a likelihood or a cost is, there are still other problems inherent to stereo correspondence. First of all, occlusion can lead to erroneous conclusion that are based on the use of the score alone. Closely related to occlusion are discontinuity regions; these can lead to wrong disparity estimates if not taken into account in template selection. Also, different textures have opposing requirements with respect to the most suitable template shape. For low texture regions, it is desirable to have a large window as a template, whereas for successful matching for high texture regions it is sufficient to use a very small window or segment with only a small number of pixels. Window/segment based matching methods inherently assume that all pixels within the matching window or segment have the same disparity. This is known as the fronto-parallel assumption. However, the fronto-parallel assumption is not always an approximation and can result in an erroneous disparity estimation.

We illustrate the above cases with the example in Figures 1.3 and 1.4. We consider different correspondence scores for four characteristic matching cases. We calculate matching scores: for a pixel in low textured regions without disparity discontinuity, marked by the blue rectangle in Figure 1.3; a pixel in a region with repetitive texture without disparity discontinuity, the red rectangle; a pixel within a region with a discontinuity, the green rectangle; and a pixel in a textured region without discontinuity, the pink rectangle. These matching windows with corresponding matching regions and epipolar lines are shown in different colors in image 1.3. We have used a similarity measure, a likelihood and a cost for stereo correspondence.

An example of a similarity measure is normalized cross-correlation (NCC). Given a rectangular window of size $(2n + 1) \times (2n + 1)$ around the current point (u, v) in left image I_l, the similarity with a rectangular window of the same size around the point with disparity d, with coordinates $(u, v - d)$, in right image I_r is calculated by

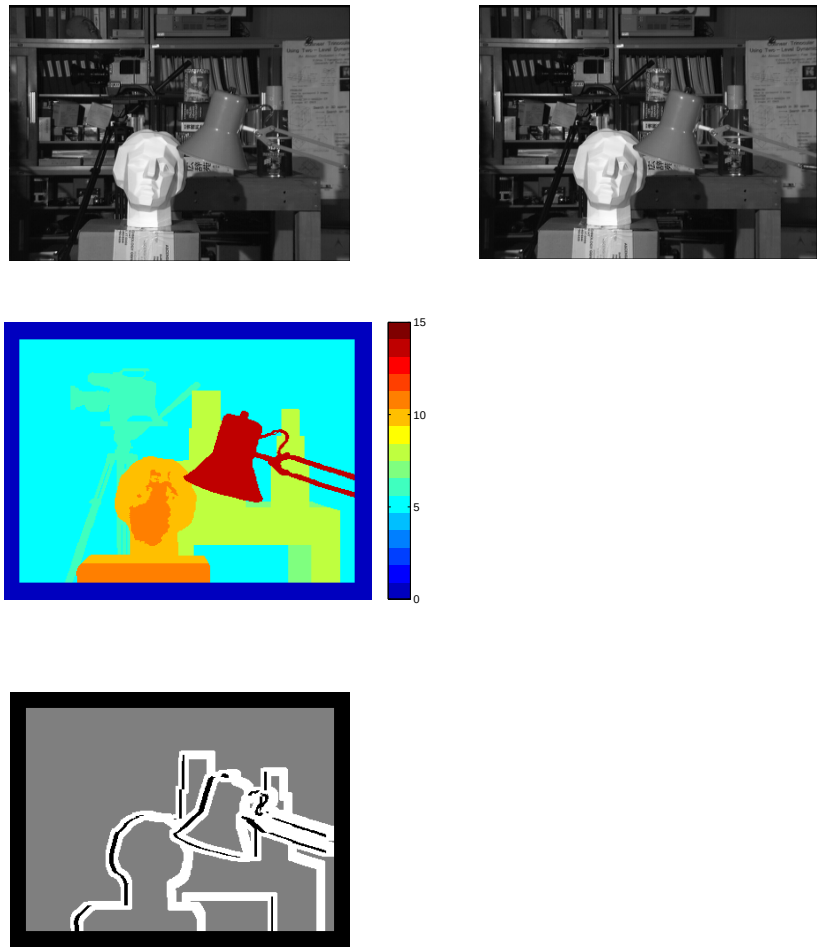


Figure 1.2 Stereo pair, ground truth disparity map, occlusion map

$$\mathrm{S}_{NCC}(u,v,d) = \frac{1}{(2n+1)^2} \cdot \frac{\sum_{i=-n}^{n}\sum_{j=-n}^{n} (I_l(u+i,v+j)-\mu_1)\cdot(I_r(u+i,v-d+j)-\mu_2)}{\sigma_1\cdot\sigma_2}, \quad (1.3)$$

where μ_1 and μ_2 are mean values of left and right windows are

$$\mu_1 = \frac{1}{(2n+1)^2} \cdot \sum_{i=-n}^{n}\sum_{j=-n}^{n} I_l(u+i,v+j) \quad (1.4)$$

and

$$\mu_2 = \frac{1}{(2n+1)^2} \cdot \sum_{i=-n}^{n}\sum_{j=-n}^{n} I_r(u+i,v-d+j), \quad (1.5)$$

and where σ_1 and σ_2 are standard deviations of left and right matching windows are

$$\sigma_1 = \sqrt{\frac{1}{(2n+1)^2} \cdot \sum_{i=-n}^{n}\sum_{j=-n}^{n} \left(I_l(u+i,v+j)-\mu_1\right)^2} \quad (1.6)$$

and

$$\sigma_2 = \sqrt{\frac{1}{(2n+1)^2} \cdot \sum_{i=-n}^{n}\sum_{j=-n}^{n} \left(I_r(u+i,v-d+j)-\mu_2\right)^2}. \quad (1.7)$$

The similarity measure results in a real number, which is the measure of the similarity of the matching windows, and it should have a maximum for the corresponding disparity. Specifically, the NCC always results in a number between -1 and 1, $S_{NCC}(u,v,d) \in [-1,1]$.

The likelihood $L(u,v,d)$ is a real non-negative number that is directly proportional to the similarity of the matching windows. One way to calculate a likelihood is to suitably transform the NCC result, for example as

$$L(u,v,d) \propto \frac{1}{1-S_{NCC}(u,v,d)}. \quad (1.8)$$

This formula transformes the NCC similarity to likelihood because it provides a measure which is non-negative $L(u,v,d) \in [0,\infty)$ and it increases with the window similarity. The similarity of matching windows can be expressed also as a cost. Cost is a kind of similarity measure that is expressed as a real number; it is inversely proportional to the similarity between matching windows. An example of a cost is sum of squared differences of all pixel intensities in matching windows; this can be presented as

$$C(u,v,d) = \sum_{i=-n}^{n}\sum_{j=-n}^{n} \left(I_1(u+i,v+j) - I_2(u+i,v+j-d)\right)^2. \quad (1.9)$$

We show an example of the behaviour of the similarity measure, likelihood, and cost for different characteristic cases in stereo matching in Figure 1.4. Matching is applied to the rectified stereo pair, meaning that the epipolar lines are horizontal and that windows are matched within the disparity range. For the stereo pair in the figure, the disparity range is $d \in [0, 15]$, so there are 16 disparity candidates. We observe characteristic cases: low-textured region matching, periodic structure matching, high-textured region matching, and matching of the occluded pixel as a central window pixel. We furthermore illustrate for those cases the similarity (2.13), likelihood (1.8), and cost (1.9).

Characteristic matching windows also have a specific behaviour, that is mirrored in the matching scores. In the case of the repetitive structure matching similarity and cost have also repetitive behaviour, while likelihood seems to be more suitable for this case with only one pronounced maximum, as illustrated in the red graphs in Figiure 1.4.

All three matching scores estimate an accurate disparity for the case of high textured region without discontinuity, as illustrated in the pink graphs in Figure 1.4.

Matching of the low textured window does not result in pronounced extreme values of any matching score, as illustrated in the blue graphs in Figure 1.4.

Matching of the window with depth discontinuity produces unreliable estimates for all scores, as shown by the green graphs in Figure 1.4 .

1.4 Problem Definition and Research Questions

In this thesis we investigate the problem of dense stereo matching. Correspondence is key problem in dense stereo matching. In dense disparity computation, correspondence needs to be solved for each point in the stereo images. The goal of a stereo matching method is to estimate a reliable disparity map.

To compute reliable dense disparity maps, a stereo algorithm must successfully deal with adverse requirements. Due to unknown differences in gains and offsets of cameras, the corresponding pixels may not have the same intensity. Also, noise can cause differences in appearance. Discontinuities in depth in a scene, such as one object in front of another with respect to camera position, can cause matching result errors if the compared region contains pixels that originate from different objects. The effect of occlusion is that not all scene points are present in both images and that the pixels do not have corresponding pixels in the other image. That results in an incorrect correspondence. If the object surface has a uniform or periodic texture, it will result in the

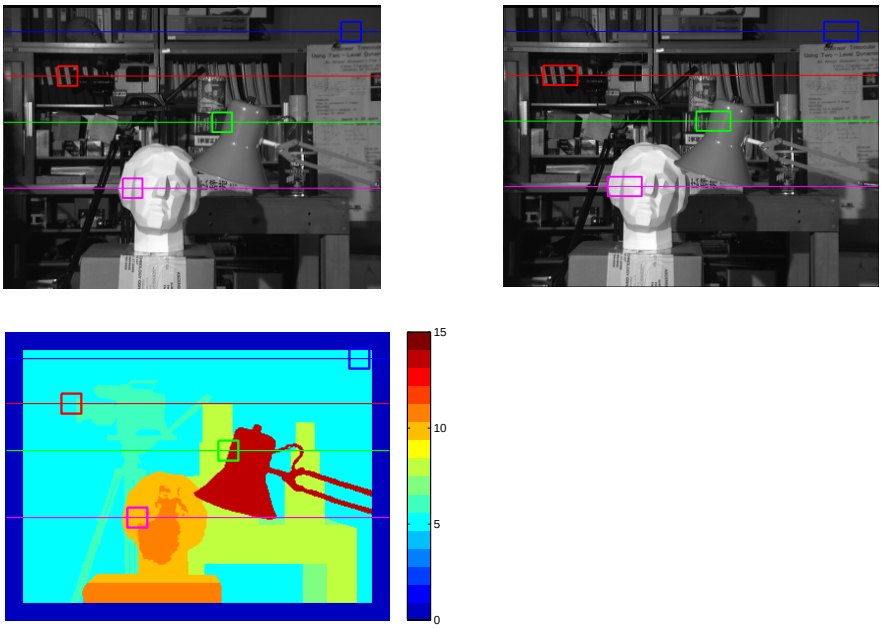


Figure 1.3 Left images: the reference stereo image and ground truth disparity map with matching windows, Right image: matching regions

similarity in a function of disparity that is either flat or has multiple periodic minima.

We begin our research by posing questions. First, we start by comparing rectangular windows and several probabilistic algorithms to investigate the influence of different algorithms on the disparity estimation. We observe the disparity estimation along the epipolar line within the probabilistic framework. As most methods for disparity estimation are rather *ad hoc*, our first research question is: **How can we design a method for disparity estimation that is optimal in a probabilistic sense?**

This first question can be broken down into a number of subquestions:

- How can we define a disparity estimation as a one-dimensional state estimation problem?
- Which probabilistic algorithms can be used to estimate disparity map from stereo images using a one-dimensional hidden Markov model?

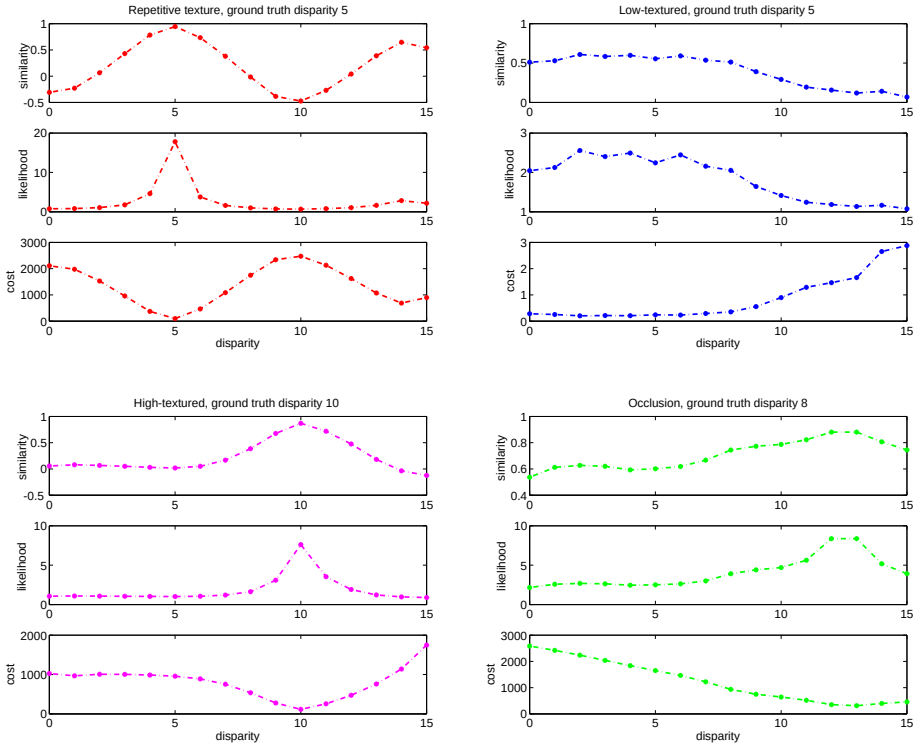


Figure 1.4 Similarity, likelihood and cost for different characteristic cases in matching

- How can particle filter be applied to estimate disparity?
- How do the different state estimation algorithms compare for different state space parameters?

Next, further improvement can be reached by using a more suitable likelihood measure. This leads to our second research question: **How can we define a likelihood measure that is optimal in a probabilistic sense?**

The related subquestion is:

- How can we obtain a likelihood measure that is invariant to unknown texture, gains and offsets?

Finally, we diverge from using the whole squared windows for similarity/cost calculation and examine the mechanism of proper pixel selection for matching within the local stereo matching framework. That leads to our third research question: **How can we define an optimal region for matching?** Related subquestions are:

- How can we suitably select a sparse subset of pixels for matching from the initial matching windows in order to diminish the influence of occlusion and depth discontinuity to the matching, and how do we calculate a matching cost?
- How can we establish a relationship between the fronto-parallel assumption and the local intensity variation for application in stereo matching? How do we select a segment for matching so that the fronto-parallel assumption holds for the segment?
- What kind of intensity transformation on the image pixels makes the image more favourable for local adaptive segmentation?
- Which postprocessing steps deal successfully with inconsistently estimated disparities?

1.5 Thesis Outline

Our research involves the pursuit of an ideal similarity measure, or cost, which will as much as possible diminish the influence of unknown gains, offsets and texture, as well as the ambiguities in stereo correspondence caused by differences in appearance, occlusion, and depth discontinuity.

We start by addressing the correspondence problem, by defining a sound one-dimensional probabilistic framework. Next, we concentrate on the derivation of a suitable likelihood function for the probabilistic matching method. Lastly, we investigate the most suitable segment selection for stereo matching within the local framework.

Following this introduction, we present in Chapter 2 a literature overview of stereo matching approaches and algorithms, and we explain a *de facto* established method of algorithm evaluation. In Chapter 3, we investigate stereo matching as a space-state problem using a one-dimensional hidden Markov model and a particle filter. In Chapter 4, we compare different probabilistic algorithms for disparity estimation.

Chapter 5 introduces a new likelihood function for window-based stereo matching that is invariant to unknown offsets, gains and texture.

In Chapter 6 we observe stereo matching within a local stereo matching framework that uses a sparse subset of pixels for matching from the initial matching windows. In Chapter 7, we perform parameter optimization of the sparse stereo matching algorithm for different stereo pairs with different scene characteristics.

In Chapter 8, we redefine some of the common assumptions used in stereo matching and establish a relationship between the local intensity variation in the image and the fronto-parallel assumption. This new interpretation of the relationship leads us to the adaptive local segmentation and a very accurate local stereo matching algorithm.

In Chapter 9 we draw conclusions, answer the research questions and recommend further research prospects.

2

Stereo Correspondence

In this chapter we introduce the scope and the context of the stereo correspondence problem and present an overview of stereo matching approaches in literature. Stereo matching is the process of finding corresponding points in stereo images. For the rectified stereo image pair, the result of this matching is a relative displacement of the corresponding points along the epipolar lines. The map of displacements for all points in the image is a disparity map. The disparity map is estimated using a local, global or semiglobal algorithm, relying on the similarity measure calculated from the image data and on some of the common matching assumptions. The last step in disparity map estimation is a disparity refinement, which detects erroneously estimated disparities and corrects their values.

2.1 Disparity Map Estimation

Stereo images are two images of the same scene taken from different viewpoints. Dense stereo matching is a correspondence problem that is aimed at finding for each pixel in one image the corresponding pixel in the other image. In dense stereo matching, the disparity for each pixel in the reference image [4] is estimated. We consider stereo matching for known camera geometry that operates on two images and produces a dense disparity map $d(x, y)$. For the rectified stereo image pair, the result of the matching is a real number that represents the relative displacement of the corresponding points along the epipolar lines. A map of all pixel displacements in an image is a disparity map.

To solve and regularize the stereo correspondence problem, it is common to introduce constraints and assumptions. The correspondence between a pixel (x, y) in the reference image and a pixel (x', y') in the matching image is then given by the equation:

$$x' = x + s \cdot d(x, y),\ y' = y, \tag{2.1}$$

where sign s, $s = \pm 1$, is a sign chosen on the basis of the reference image.

Generally, not each pixel has a corresponding pixel due to occlusion. The stereo matching is generally ambiguous as it involves an ill-posed problem due to occlusions and due to specularities caused by non-Lambertian surfaces, or lack of texture, [2]. It is necessary to apply certain assumptions to the matching process in order to obtain a solution. Many assumptions and constraints are introduced to regularize the stereo correspondence [3].

The epipolar constraint is a geometric constraint imposed by the imaging system, which causes the stereo matching to be transformed into a one-dimensional problem. Corresponding points must lie on the corresponding epipolar lines.

The disparity limit constraint regards the maximum disparity range. It can be estimated on the base of the maximum and minimum depth and geometry of a stereo system.

The *constant brightness assumption* (CBA) or *Lambertian assumption* states that corresponding pixels have identical or very similar appearances in the stereo images.

The *smoothness constraint* states that the disparity varies smoothly except at depth discontinuities.

The *fronto-parallel constraint* is an approximation of the smoothness constraint. It assumes that all pixels within the matching region have the same disparity.

The uniqueness constraint is one of the fundamental assumptions. It states that a point in one image should have no more than one corresponding point in the other image, [7]. However, the uniqueness constraint is not fulfilled for highly horizontally slanted surfaces because horizontal slant leads to unequal projections in the two cameras. That requires modification of stereo algorithms for allowing M-to-N pixel or one-to many correspondences, [8, 9]. A simple test for cross-checking is given by

$$|d_l(x, y) + d_r(x', y)| < 1 \tag{2.2}$$

where (x, y) and (x', y) are the correspondence pairs in left and right images with disparities $d_l(x, y)$ and $d_r(x', y)$. The uniqueness constraint can be alleviated for the highly slanted surfaces and be extended to allow for one-to-many mapping scenario as

$$|d_l(x, y) + d_r(x', y)| \leq t \tag{2.3}$$

where $t \geq 1$, [9].

The continuity constraint (CONT) states that the disparity varies smoothly everywhere, except on the small fraction of the area on the boundaries of objects where discontinuity occurs, [7].

The occlusion constraint (OCC) states that a disparity discontinuity in one image corresponds to an occlusion in the other image and vice versa. Discontinuities in depth map usually occur on the intensity edges.

The visibility constraint (VIS) is fulfilled for the points visible in both images, i.e. points that are not occluded. The visibility constraint requires that an occluded pixel has no match in the other image and that a non-occluded pixel has at least one match [10]. The visibility constraint is self-evident because it is derived directly from the definition of occlusion. A pixel in the left image will be visible in both images if there is at least one pixel in the right image that matches it. Unlike the uniqueness constraint, the visibility constraint permits many-to-one matching.

The ordering constraint (ORD) states that the projections of the scene points appear in the same order along the epipolar lines in images [2], i.e. the order of the features along epipolar lines is the same. However, the ordering constraint does not hold if a narrow occluding object is closest to the cameras. This is known as *the double nail illusion* [11], [10].

The limit of the disparity gradient states that the maximum directional derivative of disparity is limited [12].

Constraints are applied locally or globally in the correspondence calculation. We therefore distinguish local and global correspondence algorithms.

2.2 Correspondence Algorithms

2.2.1 Local algorithms

Local algorithms apply constraints to a small number of pixels surrounding a pixel of interest. The starting points are the Lambertian assumption and the disparity limit constraint. The final disparity for the reference pixels is estimated based on the similarity measure or matching cost between local regions around the pixel of interest in the reference image and around a matching pixel in the non-reference image. The final estimated disparity is the disparity with the highest similarity measure or with the lowest matching cost. This method is known as *winner-take-all* (WTA) method.

2.2.2 Global Algorithms

Global correspondence methods exploit nonlocal constraints in order to reduce sensitivity to local regions in the image that fail to match due to occlusion or uniform texture. In global methods, disparity computation is formulated as a global energy minimization process. Two-dimensional energy minimization is generally an NP-hard problem. The optimization techniques also incorporate some regularization steps in order to make the calculation time linear or polynomial. Global methods consist of matching cost computation and disparity optimization.

Energy Minimization

Stereo matching can be interpreted as assigning a label to each pixel in the reference image, where labels represent disparities. Such pixel-labeling problems are represented in terms of energy minimization, where the energy function has two terms: one term penalizes solutions that are inconsistent with the observed data, while the other term enforces spatial coherence (piecewise smoothness). This framework has its interpretation in terms of a maximum *a posteriori* estimation of a Markov random field (MRF) [13], [14], [15].

Every pixel $p \in \mathcal{P}$ must be assigned a label in some finite set $\mathcal{L}$. The aim is to find the labeling f that assigns each pixel $p \in \mathcal{P}$ a label $f_p \in \mathcal{L}$, where f is piecewise both smooth and consistent with the observed data. The labeling f minimizes the energy

$$E(f) = E_{data}(f) + E_{smooth}(f). \tag{2.4}$$

E_{smooth} measures to what extent f is not piecewise smooth, while E_{data} measures the disagreement between f and observed data. E_{smooth} should be

discontinuity preserving. Considering the first-order Markov Random Fields (MRF), the energy terms are

$$E_{data}(f) = \sum_{p \in \mathcal{P}} D_p(f_p) \quad \text{and} \quad E_{smooth}(f) = \sum_{\{p,q\} \in \mathcal{N}} V_{p,q}(f_p, f_q), \tag{2.5}$$

where $\mathcal{N}$ are the edges in the four-connected image grid graph. D_p measures how well label f_p fits pixel p given the observed data; it is also referred to as the data cost. D_p needs to be nonnegative. Interaction penalty $V_{p,q}(f_p, f_q)$ is the cost of assigning labels f_p and f_q to two neighboring pixels; it is also referred to as the discontinuity cost. In general, V must be metric or semimetric in order to optimize it by graph cut algorithm [14]:

$$V(\alpha, \beta) = 0 \;\Leftrightarrow\; \alpha = \beta, \tag{2.6}$$

$$V(\alpha, \beta) = V(\beta, \alpha) \geq 0, \tag{2.7}$$

$$V(\alpha, \beta) \leq V(\alpha, \gamma) + V(\gamma, \beta), \tag{2.8}$$

for any labels $\{\alpha, \beta, \gamma\} \in \mathcal{L}$. If V satisfies only (2.7) and (2.8) it is called a semimetric. The simplest discontinuity preserving model is given by the Potts model

$$V_{p,q}(f_p, f_q) = K \cdot T(f_p \neq f_q) \tag{2.9}$$

where $T(\cdot)$ is 1 if its argument is true and otherwise 0, and K is some constant. This model encourages piecewise constant labeling. The cost can be truncated to make it insensitive to the outliers. The energy expression can be extended to model occlusions [16], segment properties [17], etc. Another class of cost function can be used for smoothing term, e.g. a truncated linear model where the cost increases linearly based on the distance between the labels f_p and f_q as

$$V_{p,q}(f_p, f_q) = \min(s \cdot |f_p - f_q|, d) \tag{2.10}$$

where s is the rate of increase in the cost, and d controls when the cost stops increasing.

This pixel labeling problem is solved by energy function minimization using *graph cuts* (GC), which is a combinatorial optimization technique [14, 18].

Bayesian Methods

Bayesian methods are global methods that model discontinuities and occlusions [19], [20], [21], [22]. Bayesian methods can be classified into two categories: dynamic programming-based or MRF-based.

Belief propagation (BP) is an efficient way to approximately solve inference problems based on passing local messages [23], [24], [15]. Field specific BP algorithms are also known as the forward-backward algorithm, the Viterbi algorithm, iterative decoding algorithms for Gallager codes and turbocodes, the Kalman filter, and the transfer-matrix approach in physics.

BP algorithm can be applied in stereo vision if the problem is defined using pairwise MRFs. In that case a Markov network is an undirected graph with observed and hidden nodes [22]. Nodes $\{y_s\}$ are observed variables, and nodes $\{x_s\}$ are hidden variables i.e. disparity. By denoting $X = \{x_s\}$ and $Y = \{y_s\}$, the posterior $P(X|Y)$ can be factorized as:

$$P(X|Y) \propto \prod_s \psi_s(x_s, y_s) \prod_s \prod_{t \in N(s)} \psi_{st}(x_s, x_t), \tag{2.11}$$

where $\psi_{st}(x_s, x_t)$ is called the compatibility matrix between nodes x_s and x_t, and $\psi_s(x_s, y_t)$ is the local evidence for node x_s. In fact, $\psi_s(x_s, y_s)$ is the observation probability $p(y_s|x_s)$. $N(s)$ represents the 4-connected neighborhood of pixel s. If the number of discrete states of x_s is L, $\psi_{st}(x_s, x_t)$ is an $L \times L$ matrix and $\psi_s(x_s, y_s)$ is a vector with L elements. Its form is identical to the posterior probability for the stereo matching defined within the Baysian framework [22]. Thus, finding the maximum a posteriori (MAP) disparity map is equivalent to finding the MAP of a Markov network meaning that BP algorithm can be applied to efficiently find the disparity map.

Dynamic programming (DP) approaches perform the optimization in one dimension assuming ordering and uniqueness constraints. Each scanline is treated individually. This often leads to a streaking effect [4]. In [21], a set of priors from a simple scene to a complex scene enforces a piecewise-smooth constraint. In [19] only occlusion and ordering constraints are used. One improvement of the DP algorithm is that it proposes a cost calculation that considers whether the matching region is continuous, discontinuity or involves occlusion in either of the images [25]. Tree-based DP performs a two dimensional optimization [26, 27].

2.2.3 Semiglobal Algorithms

The Semiglobal Matching (SGM) method is based on the idea of pixel-wise matching of Mutual Information (MI) and approximating a global two- dimensional smoothness constraint by combining many one-dimensional radial constraints [28, 29]. The pixel cost and the smoothness constraint are expressed by defining the energy that depends on disparity map D, with the addition of the smoothness constraint which penalizes changes of neighboring

disparities. The greater the discontinuity, the more it is penalized. All costs along the eight or sixteen radial paths are added up. The final disparity is determined as in local stereo methods by selecting for each pixel the disparity that corresponds to minimal cost.

SGM yields no streaking artifact. SGM minimizes global two-dimensional energy as a function of disparity map, $E(D)$, by solving a large number of one-dimensional minimization problems. The energy functional is

$$\mathrm{E(D)}=\sum_{\mathbf{p}} \left(C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1 \cdot \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2 \cdot \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1] \right) \quad (2.12)$$

Function $T[\cdot]$ is defined to return 1 if its argument is true and otherwise it returns 0. In energy equation (2.12), the first term calculates the sum of a pixel-wise matching costs $C(\mathbf{p}, D_{\mathbf{p}})$ using, for example BT measure for all pixels $\mathbf{p} = I_l(u, v)$ at their disparities $D_{\mathbf{p}} = D(u, v)$. The second term penalizes small disparity differences of neighboring pixels $\mathbf{q} = I_l(u + i, v + j)$ in neighborhood $N_{\mathbf{p}}$ of point $\mathbf{p}$ with cost P_1. Similarly, the third term penalizes larger disparity steps, i.e. discontinuities with a higher penalty P_2.

SGM calculates energy $E(D)$ along one-dimensional paths from eight directions toward each pixel. The costs of all paths are summed for each pixel and disparity. The disparity is then determined on winner-take-all basis.

2.3 Similarity Measure and Matching Cost

The corresponding pixels in stereo images do not have the same gray intensities or color due to noise, sampling, and the different and unknown gains and offsets of the stereo cameras. This causes the Lambertian assumption to be only approximately satisfied. To make a matching cost and a similarity measure to be more robust to these image imperfections, the cost or similarity is not calculated using only matching pixels but is instead aggregated over the local region around the matching pixels.

The most common similarity measures and cost functions are the normalized crosscorrelation (NCC), the sum of absolute differences (SAD), the sum of squared differences (SSD). We consider the expressions for calculation of the matching score between rectangular window of a size $(2n + 1) \times (2n + 1)$ around the current point (u, v) in left image I_l, and a rectangular window of the same size around the point with disparity d, with coordinates $(u, v - d)$, in the right image I_r.

Normalized crosscorrelation (NCC), also known as zero-mean normalized crosscorrelation (ZNNC), is a similarity measure calculated by formula

$$\mathrm{S}_{NCC}(u,v,d) = \frac{1}{(2n+1)^2} \cdot \frac{\sum_{i=-n}^{n}\sum_{j=-n}^{n} (I_l(u+i,v+j)-\mu_1)\cdot(I_r(u+i,v+j-d)-\mu_2)}{\sigma_1\cdot\sigma_2}, \quad (2.13)$$

where μ_1 and μ_2 are mean values and where σ_1 and σ_2 are standard deviations of the pixels within left and right matching windows.

ZNCC accounts for gain differences and constant offsets of pixel values. The NCC always results in a number between -1 and 1, $S_{NCC}(u,v,d) \in [-1,1]$. It should have a maximum for the corresponding disparity.

Absolute difference (AD) is a pixel-wise cost:

$$C_{AD}(u,v,d) = |I_l(u,v) - I_r(u,v-d))|. \quad (2.14)$$

Sum of absolute differences (SAD) aggregates the AD of the pixels within the matching region :

$$C_{SAD}(u,v,d) = \sum_{i=-n}^{n}\sum_{j=-n}^{n} |I_l(u+i,v+j) - I_r(u+i,v+j-d))|. \quad (2.15)$$

AD and SAD assume the corresponding pixels to be identical. There is also a zero-mean sum of absolute differences (ZSAD). The mean window intensity is subtracted from each intensity inside the window before computing the sum of absolute differences:

$$\mathrm{C}_{ZSAD}(u,v,d) = \sum_{i=-n}^{n}\sum_{j=-n}^{n} |I_l(u+i,v+j) - \mu_1 - (I_r(u+i,v+j-d)-\mu_2)|\,. \quad (2.16)$$

Sum of squared differences (SSD) is a cost measure

$$C_{SSD}(u,v,d) = \sum_{i=-n}^{n}\sum_{j=-n}^{n} \left(I_1(u+i,v+j) - I_2(u+i,v+j-d)\right)^2. \quad (2.17)$$

Common measures can be applied also for colored instead of gray images. For color images, the sum of absolute differences can be defined as the maximum absolute difference of the color channels [30].

Improved common measures The common measures can also be improved by combining them with some certain other custom measures. For example, the SAD measure can be improved by extending it by the gradient measure, [31],

$$C = (1 - w) \cdot C_{SAD}(u, v, d) + w \cdot C_{GRAD}(u, v, d) \tag{2.18}$$

where w represents optimal weighting factor calculated through several iterations and $C_{GRAD}(u, v, d)$ is a gradient based cost.

Birchfield and Tomasi measure (BT) reduces the dissimilarity in high-frequency regions [32], [33]. The BT measure computes the sampling insensitive absolute difference between the extrema of linear interpolations of the corresponding pixels of interest with their neighbors:

$$\begin{aligned}
C_{BT} &= min(A, B), \qquad (2.19)\\
A &= \max(0, I_l(u, v) - I_r^{max}(u, v - d), I_r^{min}(u, v - d) - I_l(u, v))\\
B &= \max(0, I_r(u, v - d) - I_l^{max}(u, v), I_l^{min}(u, v) - I_r(u, v - d))\\
I_{l/r}^{min}(u, v) &= \min(I_{l/r}^{-}(u, v), I_{l/r}(u, v), I_{l/r}^{+}(u, v))\\
I_{l/r}^{max}(u, v) &= \max(I_{l/r}^{-}(u, v), I(u, v), I_{l/r}^{+}(u, v))\\
I_{l/r}^{-}(u, v) &= \frac{I_{l/r}(u, v - 1) + I_{l/r}(u, v)}{2}\\
I_{l/r}^{+}(u, v) &= \frac{I_{l/r}(u, v + 1) + I_{l/r}(u, v)}{2}.
\end{aligned}$$

Filter-based matching measures are mean filter, Laplacian of Gaussian (LoG) filter, or bilateral filter. The filtering results in conjunction with BT, or AD measure can be used in a global pixel-wise matching framework [33].

Mean filter (MF) subtracts from each pixel the mean intensities within a squared window centered at the pixel of interest. Thus, the mean filter performs background subtraction for removing a local intensity offset:

$$I_{MF}(u, v) = I(u, v) - \frac{1}{(2n + 1)^2} \sum_{i=-n}^{n} \sum_{j=-n}^{n} I(u + i, v + j). \tag{2.20}$$

Laplacian of Gaussian (LoG) is a bandpass filter, which performs smoothing, removing noise and an offset in intensities. The filter is often used in local realtime methods [34]. In [33] a LoG filter with a standard deviation of σ pixel

is used, which is applied by convolution with a squared LoG kernel:

$$I_{LoG} = I \otimes K_{LoG},\ K_{LoG} = -\frac{1}{\pi\sigma^4}\left(1 - \frac{u^2+v^2}{2\sigma^2}\right)e^{-\frac{u^2+v^2}{2\sigma^2}}. \tag{2.21}$$

Bilateral filter [35], [36], [33], is smoothing technique that preserves the edge. It sums neighboring values weighted according to proximity and color similarity. Background subtraction is implemented by subtracting from each value the corresponding value of the bilateral filtered image. The parameters of the bilateral filter are the window size $M \times M$, a spatial distance σ_s which defines the amount of smoothing, and a radiometric distance σ_r which prevents smoothing over high-contrast texture differences. This effectively removes a local offset without blurring high-contrast texture differences that may correspond to depth discontinuities. On intensity images, the radiometric distance is computed as the absolute difference of intensities; on color images, the distance in CIELab space is used, as suggested in [35]

$$I_{BilSub}(u,v) = I(u,v) - \frac{\sum_{i=-n}^{n}\sum_{j=-n}^{n} I(u+i,v+j)e^s e^r}{\sum_{i=-n}^{n}\sum_{j=-n}^{n} I(u+i,v+j)}, \tag{2.22}$$

where

$$s = -\frac{(i-j)^2}{2\sigma_s^2},\ \ r = -\frac{(I(u+i,v+j) - I(u,v))^2}{2\sigma_r^2}. \tag{2.23}$$

Mutual information (MI) measure calculates the joint probability distribution P_{I_l,I_r} of corresponding intensities in images I_l and I_r, which is necessary for calculation of the estimate of the joint entropy h_{I_l,I_r} as well as for estimation of image entropies h_l and h_r [37], [28]. The probability distribution P_{I_l,I_r} is calculated on the basis of the histogram of the corresponding intensities, [28]. The starting disparity map for P_{I_l,I_r} calculation can be obtained by correlation. The cost is calculated as negative mutual information $mi_{I_l,I_r}(u,v,d)$

$$C_{MI}(u,v,d) = -mi_{I_l,I_r}(u,v,d). \tag{2.24}$$

This cost measure is well suited for reach-textured regions and is invariant to radiometric differences such as camera gain and bias uncertainties and specularities [37, 38, 33].

Nonparametric matching costs are rank filter, soft rank filter, census filter and ordinal measure [33]. These matching scores are robust against intensity outliers. They use only the local ordering of intensities and are robust to all monotonic mapping radiometric changes. These measures transform image intensities. The transformed images are matched with, for example, the absolute difference.

Rank filter replaces the intensity of a pixel with its rank among all pixels within a certain neighborhood $N_{\mathbf{p}}$, for example within a rectangular window of size $(2n+1) \times (2n+1)$

$$I_{Rank}(u,v) = \sum_{i=-n}^{n} \sum_{j=-n}^{n} T[I(u,v) < I(u+j, v+i)],\ (i,j) \neq (0,0). \quad (2.25)$$

The function $T[\cdot]$ is defined to return 1 if its argument is true and 0 otherwise. The rank filter was proposed to increase the robustness of window-based methods to outliers within the neighborhood, which typically occur near depth discontinuities and leads to blurred object borders [39]. The *Rank filter* is susceptible to noise in textureless areas.

The *soft rank filter* was proposed to reduce the influence of noise in textureless areas by defining a linear, soft transition zone between 0 and 1 for values that are close together:

$$\mathrm{I}_{SoftRank}(u,v) = \sum_{i=-n}^{n} \sum_{j=-n}^{n} \min\left(1, \max\left(0, \frac{I(u,v) - I(u+j, v+i)}{2t} + \frac{1}{2}\right)\right),\ (i,j) \neq (0,0), \quad (2.26)$$

where t is a threshold [33].

The *census filter* defines a bit string where each bit corresponds to a certain pixel in the local neighborhood around a pixel of interest. A bit is set when the corresponding pixel has a lower intensity than the pixel of interest. Thus, census filter not only stores the intensity ordering as rank filter does, but also the spatial structure of the local neighborhood. The transformed images can be matched by computing the Hamming distance between corresponding bit strings [39]. The performance of census is superior to rank [39], but the computational time is longer due to the calculation of the Hamming distance.

The *ordinal measure*, [40], is based on the distance of rank permutations of corresponding matching windows and requires window-based matching. Its potential advantage over rank and census filters is that it avoids dependency on the value of the pixel of interest.

2.4 Matching Primitives

The starting point in local as well as in global stereo correspondence methods is calculation of the matching score using the local neighborhood around the matching pixel. With respect to what kind of local region is taken into account, we distinguish between pixel-based and area-based methods. Global algorithms are usually pixel-based, and data energy term is usually calculated strictly on the basis of the values of the matching pixels. This is acceptable because other terms in the energy functional take into account the neighboring pixels and because the optimization is global. On the other hand, local correspondence algorithms are usually area-based and local pixel areas are used in cost or similarity calculation. Area-based stereo methods match neighboring pixels within generally rectangular window.

Algorithms based on rectangular window matching yield an accurate disparity estimation so long as the majority of the window pixels belongs to the same smooth object surface, with only a slight curvature or inclination relative to the image plain. In all other cases, window-based matching produces an incorrect disparity map: the discontinuities are smoothed, and the disparities of the high-textured surfaces are propagated into low-textured areas [44]. Another restriction of window-based matching is the size of objects whose disparity must be determined. Whether the disparity of a narrow object can be correctly estimated depends mostly on the similarity between the occluded background, visible background, and object [34]. Algorithms which use suitably shaped matching areas for cost aggregation result in a more accurate disparity estimation [73],[76], [66], [77], [68], and [75]. The matching region is then selected using pixels within certain fixed distances in RGB, CEILab color space, and/or Euclidean space.

Rectangular window matching is a common approach in real time applications because of its low computational load and efficient hardware implementation [41], [42], [43]. Inherently, the fronto-parallel disparity regions are assumed. The window matching produces unwanted smoothing and creates the phenomena of fattening and shrinkage of a surface, causing that surface with high intensity variation to extend into neighboring less-textured surfaces across boundaries [44]. A way to remove any fattening effect is to employ the adaptive weight scheme using bilateral filtering [35]. Window-based matching is not suitable for stereo images with surfaces with projective distortion. To reduce the effect of projective distortion, it is necessary to estimate the surface orientation and to take it into account during matching, or to use matching using adaptive windows.

A way to obtain a more accurate disparity estimation around disparity

discontinuities is to apply a shiftable window approach. A shiftable window approach considers multiple square windows centered at different locations and uses the one that yields the smallest average cost [45], [20]. In this approach the size of the window is fixed. Shiftable windows can recover object boundaries quite accurately if both foreground and background regions are textured, and as long as the window fits as a whole within the foreground object. A generalization of the shiftable window method is to employ a variable support strategy on all points detected close to a depth edges, where the final matching cost is obtained by averaging the error function along those displacement positions detected as lying on the same border side [46], [34].

Improved accuracy by window matching is possible by variable support i.e. by allowing the support to have any shape instead of being built upon rectangular windows only, or by assigning adaptive weights to the points belonging to the support window. Area-based algorithms use an alternative approach and vary the size and shape of the window rather than its displacement [47]. This allows the use of bigger areas within low-textured regions for the matching score calculation. Segment-based matching adapts to local characteristics of the image data. One of the first segment-based algorithms is iterative algorithm as given in [48]. Mean shift [49] is the most common algorithm for image segmentation in homogenous color regions [29, 29, 31]. In segment-based matching, it is assumed that disparity inside a segment follows some particular disparity model, for example constant, planar, or quadratic. A drawback of segment-based matching methods is that depth discontinuities may not lie along color boundaries [50], [51].

2.5 Disparity refinement

A disparity map estimated by the correspondence algorithm may contain errors. It can contain areas of incorrect disparity values caused by large low -textured areas. It can also contain isolated disparity errors with significantly different disparity from the neighborhood disparities, so called outliers, caused by isolated pixels or groups of several pixels. Also, there may be disparity errors caused by occlusion. The disparity errors are detected and corrected for in a postprocessing.

The postprocessing step performs a disparity consistency check between disparity maps estimated for both stereo images, eliminates inconsistent disparities, and estimates new values for the eliminated disparities.

2.5.1 Dealing with the Occlusion

Occlusion refers to points in a scene which are visible in one but not in the other image due to scene and camera geometries [3]. Points that are visible in one of two views provided by a binocular imaging system are also termed as binocular half-occluded point [52]. The depth of half-occluded points can not be estimated from the stereo images. Matching methods can be classified into three categories with reference to how they deal with occlusion: methods that detect occlusion, methods that reduce sensitivity to occlusion, and methods that model occlusion geometry [3].

The simplest approach to occlusion regions is to detect them. Occlusion can be observed as the outlier in disparity maps and be eliminated by median filtering. The consistency assumption can also be used for occlusion detection, provided that two disparity maps are calculated. One disparity map is based on the matching of the left image against the right image and the other based on the matching of the right image against the left. Areas with inconsistent disparities are assumed to be occluded. This method is also known as *Left-Right Checking* (LRC) and as left-right cross/consistency checking. The consistency check is based on the occlusion constraint. Both occlusion and mismatches can be distinguished as part of the left/right consistency check [29, 28]. The ordering constraint can also be used to detect disparity outliers, although it is not correct for narrow structures, [22].

A comparison of five different approaches for occlusion detection is presented in [52]. The methods considered are Bimodality (BMD), Match Goodness Jump (MGJ), Left-Right Checking (LRC), Ordering (ORD) and the Occlusion constraint (OCC). *Bimodality* (BMD) occlusion detection is based on the principle that points around occlusion points will match to both the occluded and occluding surface, creating a bimodal distribution in a local histogram of the disparity image. In such regions, the histogram of the disparity should be bimodal. The peak ratio is the ratio of the second highest peak versus the highest peak. The peak ratio test determines whether there is any bimodality. The Match Goodness Jump (MGJ) detects adjacent regions of high/low scores in goodness-of-match. It must be concluded that it does not appear to lead to a simple one-dimensional goodness ranking of the methods. LRC performs well in highly textured scenes, and OCC performs well given a matcher with smoother error characteristics. In scenes with weak texture, MGJ labels occlusions in a reasonable fashion outperforming the other methods in similar situations. For scenarios where three-dimensional border detection is of primary interest, including borders that do not manifest themselves as half-occlusions, BMD performs well, although with a tendency to over

segment the scene. Overall, ORD is the most conservative measure, although it can still produce false positives and is sensitive to the double-nail illusion.

It is desirable to integrate knowledge of occlusion geometry into the search process. This is done within global correspondence methods. In [21], the priors that address a more complicated model of the world, for a series of Bayesian estimators are defined. These are used to define cost functions for dynamic programming.

The use of robust matching measures, such as normalized cross-correlation and nonparametric costs, is one way to reduce the sensitivity of matching to occlusion and to other image differences such as perspective differences and sensor noise. Nonparametric transforms are applied to image intensities before cost calculation [39]. Since these methods rely on relative ordering of intensities rather than on the intensities themselves, they are somewhat robust to outliers. However, the presence of occlusion in a stereo image pair produces disparity discontinuities that are coherent. In other words, while they are outliers to the structure of interest, they are inliers to a different structure.

Another approach to reduce sensitivity to occlusion is to adaptively resize the window and shape in order to optimize the match similarity near occlusion boundaries. In [53], an iterative method for determining window size is proposed. In area based matching algorithms, to alleviate the fronto-parallel assumption, some approaches allow the matching area to lie on the inclined plane, such as in [78] and [79]. The alternative to the idea that properly shaped areas for cost aggregation can result in more accurate matching results is to allocate different weights to pixels in the cost aggregation step. In [54], the pixels closer in the color space and spatially closer to the central pixel are given proportionally more significance, whereas, in [69], the additional assumption of connectivity plays a role during weight assignment.

2.6 Evaluation of Stereo Algorithms

The *de facto* standard for stereo algorithm evaluation, widely accepted within the vision community, is the Middlebury online evaluation benchmark [6]. It evaluates estimated disparity maps by a stereo algorithm of four benchmark stereo image pairs and ranks the results within the online evaluation list. The benchmark stereo pairs are of different size and disparity range, with different scene geometries and versatile texture. The benchmark for stereo algorithms is done on the base of the taxonomy and quantitative evaluation of dense, two-frame stereo algorithms introduced in [4].

The evaluation of the stereo algorithm within the Middleburry framework

is done by examining the error percentage within non-occluded regions, discontinuity regions and occluded regions in estimated disparity maps for all four reference images. Test data and rankings are provided on the Internet [6]. At the moment, the database includes more than 130 ranked algorithms.

3

Stereo Matching Using Hidden Markov Models and Particle Filtering [1]

In this chapter we investigate a new approach to stereo matching using probabilistic techniques and demonstrate that particle filtering is a suitable technique for this application. The potential advantage of particle filtering over other approaches is its flexibility and the ease of incorporating more complex knowledge of the scene into the probabilistic model. We perform the matching using a pair of rectified stereo images, assuming that the scene statistics is described by a first order hidden Markov model (HMM). Stereo matching is treated as state estimation, where the state variable is the disparity. Evolution of the state variable happens along the epipolar line. The transition probabilities allow for continuous and abrupt transitions, i.e. changes in disparity. The likelihood values are derived using the normalized crosscorrelation map (NCC).

This paper presents the first implementation of particle filtering in conjunction with HMM applied to stereo correspondence. We demonstrate that particle filtering with HMM can be successfully applied to stereo matching.

[1] This chapter is based on the paper S. Damjanović, F. van der Heijden and L. J. Spreeuwers, "Stereo Matching Using HMM and Particle Filtering", ProRISC 2008, Veldhoven, The Netherlands

3.1 Introduction

Stereo matching is a correspondence problem whose aim is to find the corresponding points in stereo images. Stereo images are two images of the same scene taken from different viewpoints. The corresponding scene points in images are always at the corresponding epipolar lines. If the geometry of the acquisition system is known, it is possible to rectify the images so that the epipolar lines become horizontal and parallel to the scan lines. The problem of matching between images can be regarded as a problem of optimizing a cost or a similarity function. The global optimization in stereo matching is often performed using dynamic programming [19], [2]. Dynamic programming (DP) is a way of efficiently minimizing functions of a large number of discrete variables.

We approach the stereo matching problem as a state estimation problem using probabilistic based algorithms. We choose a one-dimensional hidden Markov model (HMM) as a prior since HMM is a special state space model in which the state space is discrete [55]. We derive the likelihood from normalized cross-correlation (NCC) coefficients and apply different probabilistic based algorithms for disparity calculation. In addition, we define a performance criterion and compare the performance of different probabilistic based algorithms against the performance of dynamic programming. We compare the probabilistic estimators against a non-probabilistic dynamic programming algorithm. We have chosen dynamic programming as a reference for the performance comparisons because dynamic programming has often been used for stereo matching, but not much in a probabilistic context.

Stochastic approaches such as sampling-based methods, including particle filters and particle filters followed by a smoothing step, can as well be used to find the global optimum. Particle filters are of interest for stereo matching since they can establish more complex models of the scene [56], [57]. Particle filters are a probabilistic-based algorithm based on a state space model. Therefore, particle filters must also be able to handle HMMs. Also, several other algorithms for HMMs are commonly used: thee Viterbi algorithm, the forward algorithm and the forward/backward algorithm [55]. We apply these to stereo matching.

The probabilistic-based algorithms: Viterbi, forward and forward/backward, are a kind of belief propagation algorithms. Belief propagation (BP) is an efficient way to approximately solve inference problems based on passing local messages [24]. There is a connection between non-probabilistic DP and the Viterbi algorithm. DP is a kind of Viterbi algorithm where only transitions between adjacent states are allowed. On the other hand, the Viterbi algo-

rithm allows transitions between any two states of the sate space model. An algorithm in between DP and the Viterbi algorithm has been proposed in the literature. It uses the reduced trellis diagram for state estimation allowing only a finite number of transitions between neighbouring states [58].

In this chapter, we define stereo matching as a one-dimensional state estimation problem with the goal of investigating how particle filters and smoothers fit into this framework. We investigate whether particle filters and smoothers in conjunction with one-dimensional HMM can be applied to stereo matching. We keep the state space model one-dimensional and compare particle filter and smoothers with other probabilistic based estimation algorithms that are commonly used with HMMs and with the reference non-probabilistic DP. We expect the particle filters and smoothers to compare well with other algorithms for this simple state space model. If this holds, then a more complex prior may be used with particle filter to better capture the scene characteristics and produce more accurate estimates [59].

The chapter is organized as follows. In section 3.2 we introduce a probabilistic framework for stereo matching using one-dimensional HMM. In section 3.3, we introduce the probabilistic based algorithms for stereo matching. In section 3.4 we summarise the fundamentals of the reference non-probabilistic DP. Experiments are reported in Section 3.5. In Section 3.6, the conclusion and further directions of work are given.

3.2 Probabilistic Framework for Stereo Matching

We consider the fully calibrated image acquisition setup, and the stereo matching is done along epipolar lines of the rectified images [2]. We approach the disparity calculation along the epipolar lines as state estimation problem [55], where the disparity is a discrete state variable. In state estimation, the state variable of interest changes over time. In the application to disparity calculation, the state variable evolution happens with the increment of the x-coordinate of the epipolar line in the reference image, while the time index of the state variable, present in the classical state estimation approach, is replaced by the x-coordinate value. Hence, the terms 'previous' and 'next' state refer to the disparity values with the smaller or larger value of the x-coordinate than the x-coordinate of the observed, i.e. 'current' disparity denoted by d_x. The length of the state sequence is the number of pixels in the reference epipolar line for which the disparity values are calculated. It is determined by the size of the window used for the likelihood calculation. If the size of the window is denoted by $W_x = 2 \cdot w_x + 1$ and the length of the referent epipolar line by L,

then the x-coordinate takes consecutive values from the array $[w_x+1,\, L-w_x]$.

Application of the probabilistic algorithms for state estimation requires a knowledge of the transition probability and the likelihood function. The transition probability $P_t(d_{x+1}|d_x)$ is the probability that the state at position $x+1$ is d_{x+1} given that the previous state at x is d_x. The discrete state variable d_x takes values from the set $\Omega_d = \{K_{min},\, K_{min}+1,\, K_{min}+2, ...,\, K_{max}\}$, where K_{min} and K_{max} are the minimum and the maximum disparities between images. The number of different states is equal to the number of different possible disparity values, $K = K_{max} - K_{min} + 1$. The initial probability, at the coordinate $x = 0$, is taken as $P_0(d_0) = P_t(d_1|d_0 = 1)$. The transition probabilities are given in the form of KxK matrix whose element in the m^{th} row and n^{th} column is the transition probability $P_t(d_{x+1}|d_x = m)$. The sum of all transition probabilities for a fixed state d_{x+1} is equal to 1, $\sum_{d_{x+1}=1}^{K} P_t(d_{x+1}|d_x) = 1$. The probability that the state variable changes its value in the next time instant is inversely proportional to the absolute difference of the consecutive state values. The state variable has the highest probability of keeping the same value and that probability is the same for all states. The probability decreases linearly with the absolute discrepancy of the consecutive variable values up to $transmax$. The probability P_{jump} that the absolute discrepancy between two state-variable values lies in the range of $[transmax,\ jumpmax]$ is a small constant value e.g. $P_{jump} < 0.15$, while the probability $P_{outlier}$ that the absolute discrepancy between two consecutive values of the state variable is higher than $jumpmax$ is constant and close to zero. Figure 3.1 illustrates the shape of the transition probability $P_t(d_{x+1}|\ d_x = n)$.

Together with the likelihood function, the transition probability forms the HMM. The likelihood function is the probability of the observation given the true state. We derive a heuristic expression based on NCC coefficients.

The number of states of the HMM is equal to the range of disparities i.e. every possible disparity value is represented as a state in the state-space model. We consider the common behaviour of the disparity values along epipolar lines in order to properly choose the HMM. The disparity value stays the same for flat fronto-parallel surfaces of the scene and changes for slanted surfaces, along the epipolar line. As can be seen in figure 3.1, the variable has the highest probability of staying in the same state. The probability sharply decreases with difference $\Delta = |d_{x+1} - d_x|$ up to $\Delta = trans_{max}$. The probability of transitions with a greater difference $\Delta \in [trans_{max},\, jump_{max}]$, 'jumps' in other words, is smaller and constant. The probability of outlier transitions, when the state change is greater than $jump_{max}$, is rather small and can perhaps be neglected. The total probability of the jump transitions and the total proba-

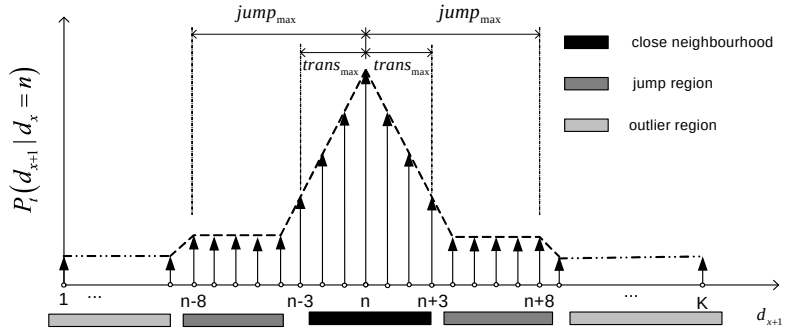


Figure 3.1 HMM: Transition probability P_t

bility of the outlier transitions are denoted as P_{jump} and P_{out}, respectively.

The likelihood function, defined on a position x on an epipolar line in the reference image, and on a position $x + d$ on the corresponding epipolar line in the other image, $p(z_{1,x},\, z_{2,x+d_x} |\, d_x = k)$, $k \in [K_{min},\, K_{max}]$, $x \in [w_x + 1,\, L - w_x]$ represents a similarity measure of the quadratic $W_x \times W_x$ windows of pixels $z_{1,x}$ surrounding the pixel in the reference image at a position x, and the windows $z_{2,x+d_x}$surrounding the pixel at position $x+d_x$ in the other image of the stereo pair. In fact, it is defined as the probability density of having the observed image data in the windows when the true disparity is given. The NCC is another measure of similarity between two pixel-windows in images. It inherently compensates for their different offsets and gains [3, 2]. Using the NCC coefficients $NCC(x,\, d_x = k)$, $k \in [K_{min},\, K_{max}]$, $x \in [w_x + 1,\, L - w_x]$ as an approximation of the likelihood is not suitable. Firstly, the NCC coefficients can be negative, while the likelihood function should always be nonnegative. Secondly, the small ratio between NCC coefficients is not sufficient for proper differentiation between different state values. Empirically we found that the following simple transformation keeps the same order of the values of the coefficients and gives a suitable metric that resembles the desired properties of a likelihood function:

$$p(z_{1,x},\, z_{2,x+d_x} |\, d_x = k) \propto \frac{1}{1 - NCC(x,\, d_x = k)}. \tag{3.1}$$

The expression is defined up to a proportionality constant, but not relevant since, for instance, in the particle filter the resulting posteriors are normalized anyhow.

3.3 Probabilistic Stereo Matching Algorithms

Stereo matching is defined as the estimation of the disparity state variable d_x along the epipolar line, $x = w_x + 1, \ldots, L - w_x$, using HMM. There are two types of state estimations: online and offline. If estimation of the state is done online (in real-time), the estimate is then obtained on the basis of the previous and current measurements, as in the case of the forward algorithm and particle filtering. If estimation is done offline, then the estimate is obtained not only using the measurements from past and present, but also by using the measurements that were made after the time instant of the state being estimated. The offline estimation is done by back propagation through the measurement sequence, same as in the forward/backward algorithm and smoothing. The role of the time index is substituted by the x-coordinate of the pixel position in the image (see Section 3.2). As all pixels values are known, we can apply both online and offline algorithms.

Applying the forward algorithm, the disparity values are calculated using the maximum *a posteriori* (MAP) criterion. The forward/backward algorithm results in disparities along the epipolar lines whose individual disparities have minimal error, while the Viterbi algorithm provides the most likely sequence of disparity values [55]. A detailed description of probabilistic algorithms is given in Chapter 4.

MAP estimation is also done within the particle filtering framework. Estimation of the posterior probabilities or filtering distribution can be realized using the standard filtering recursions via the Chapman-Kolmogorov equation

$$p(d_{x+1}|\, Z(x)) = \int p(d_x|\, Z(x)) \cdot p_t(d_{x+1}|d_x) d(d_x) \tag{3.2}$$

and via Bayes' rule for the update

$$p(d_{x+1}|\, Z(x+1)) = \frac{p(z_{1,\,x+1},\, z_{2,\,x+1+d_{x+1}}|\, d_{x+1}) p(d_{x+1}|\, Z(x))}{p(z_{1,\,x},\, z_{2,\,x+d_x}|\, Z(x))}. \tag{3.3}$$

In (3.3), $Z(x) = \{(z_{1,\,i},\, z_{2,\,i+d_x})_{i \le x,\, d_x = K_{min}, \ldots, K_{max}}\}$ i.e. $Z(x)$ represents pairs of patches in images for which the disparity is calculated. The approximation strategy for the posterior probability density (3.2) is the sequential Monte Carlo method, known as particle filter [55]. Within the particle filter framework, the filtering distribution is approximated by an empirical distribution formed of the point of masses, or particles [60]. So the posterior probability distribution is given by the particle approximation as

$$p(d_x|Z(x)) \simeq \sum_{n=1}^{N_p} w_{i,\,norm}^{(n)} \delta(d_x - d_x^{(n)}) \tag{3.4}$$

where $\delta(\cdot)$ is the Dirac delta function, N_p is the number of particles. The normalized importance weights $w_{i,\,norm}^{(n)}$ are chosen as

$$w_{i,norm}^{(n)} = p(z^{(x)}|\, d_x^{(n)}),\ \sum_{n=1}^{N_p} w_{i,\,norm}^{(n)} = 1, w_{i,\,norm}^{(n)} > 0,\ z^{(x)} = (z_{1,\,x},\ z_{2,\,x+d_x}). \quad (3.5)$$

The number of particles N_p should be sufficiently large in order to represent properly the posterior distribution by means of a set of samples. For the purpose of comparison with the forward algorithm, the state estimation is performed with the MAP criterion.

Smoothing can be performed recursively backward using the smoothing formula

$$p(d_x|Z(x)) = \int p(d_{x+1}|\, Z(x)) \frac{p(d_x|\, Z(x)) p_t(d_{x+1}|\, d_x)}{p(d_{x+1}|\, Z(x))} d(d_{x+1}) \quad (3.6)$$

Smoothing is applied in addition to particle filtering to generate the realizations of the entire smoothing density $p(d_{x=w_x+1,\,...,\,L-w_x}|\, Z(i))$ based on the forward particle filtering results, [60]. The estimated sequence is equal to the sequence of the most probable individual states.

3.4 Dynamic Programming

Dynamic programming computes the minimum-cost path through the matrix of all pairwise matching costs between two corresponding scanlines [4]. In the case of nonexistent match i.e. occlusion, a group of pixels in one image is assigned to a single pixel in the other image.

Figure 3.4 schematically shows how DP works. For each pair of corresponding scanlines, a minimizing path through the matrix of all pairwise matching costs is selected. Lowercase letters a to k symbolize the intensities along each scanline. Uppercase letters represent the selected path through the cost matrix. Matches are indicated by **M**, while partially occluded points, which have a fixed cost, are indicated by **L** or **R**, corresponding to points only visible in the left or right images, respectively. Usually, only a limited disparity range is considered indicated by the non-shaded squares. The disparity range in Figure 3.4 is 0 to 4.

To implement dynamic programming for a scanline, each element in a two-dimensional cost matrix $C(m, n)$ is computed by combining its value with one of its predecessor cost values. Using the representation shown in figure 3.4, the aggregated match costs can be recursively computed as

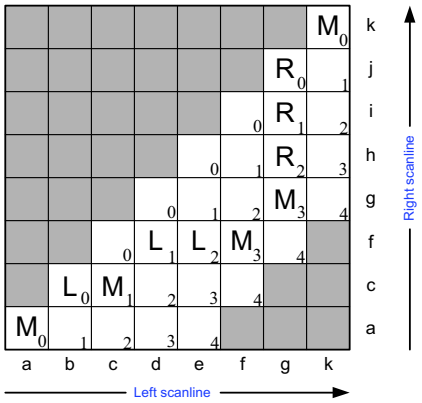


Figure 3.2 Stereo matching using dynamic programming, [4]

$$C(m,n,M) = \min(C(m-1,n-1,M), C(m-1,n,L), D(m-1,n-1,R)) + C_1(m,n)$$
$$C(m,n,L) = \min(C(m-1,n-1,M), C(m-1,n,L)) + O$$
$$C(m,n,R) = \min(C(m,n-1,M), C(m,n-1,R)) + O, \quad (3.7)$$

where O is a occlusion penalty and $C_1(m,n)$ is a matching cost of individual pixels at position m in the left scanline and at position n in the right scan line.

Within our probabilistic framework, the DP algorithm can be interpreted as a Viterbi algorithm whose HMM only allows transitions to the same state and to two immediately neighboring states.

The matching cost of the individual pixels is calculated using windows around pixels and normalized crosscorrelation as

$$C_1(m,n) = 1 - NCC(m, m-n). \quad (3.8)$$

Transformation (3.8) maps normalized crosscorrelation $NCC(m, m-n)$, which is similarity measure with a value between -1 and 1, to cost $C_1(m,n)$ with a value between 2 and 0.

3.5 Experiments

We illustrate stereo matching, using non-probabilistic dynamic programming and probabilistic-based algorithms, on a rectified stereo pair *Bowling2* shown

(a) (b) (c)

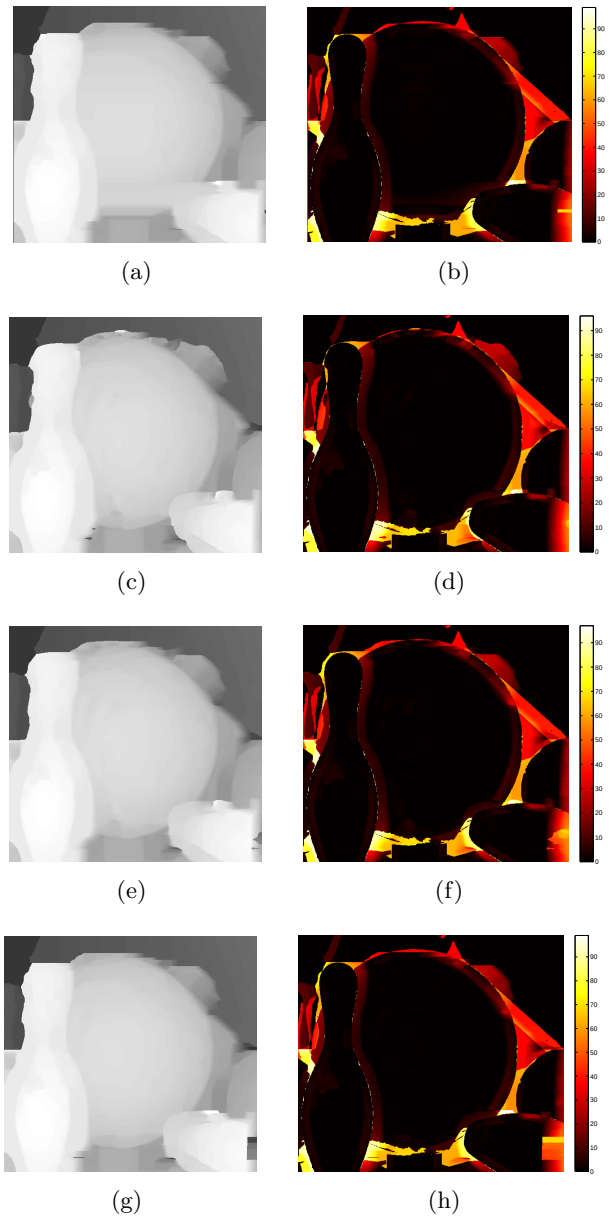
Figure 3.3 Stereo pair *Bowling2* from [6]: (a) left image, (b) right image, and (c) ground truth disparity

in Figure 3.3 (a) and (b), [33, 6]. The purpose of the experiments is to compare the performance of probabilistic based algorithms with a standard non-probabilistic algorithm DP with similar functional structure, to compare the performance of the different optimization criteria of the probabilistic algorithms and to check whether particle based algorithms perform equally as the HMM based algorithms with the same optimization criterion. These experiments are the first, preliminary experiments involving comparison of different algorithms within our one-dimensional probabilistic stereo matching framework.

The disparity values range from 19 to 99, while the occluded pixels are assigned the value 0. So we choose an HMM with $K_{min} = 19$ and $K_{max} = 99$. We select the parameters for the state transition probabilities: $P_{out} = 0$, $P_{jump} = 0.05$, $jump_{max} = 8$ and $trans_{max} = 3$. The size of the windows for the calculation of the NCC coefficients is $W_x = 31$. The number of particles used in the particle filtering is $N_p = 1000$ and in the smoothing step $N_{back} = 100$.

The recovered disparity maps and absolute error maps are very similar. This confirms that the probabilistic techniques can be successfully applied to the stereo matching. The similar have the same causes, namely, the size of the windows used in the calculation of the NCC coefficients and correspondingly likelihood values, limit more precise depth calculation. Smaller windows would yield better results concerning occlusion detection. Also, the perspective distortion is not taken into account. Similarly, the simple prior does not model occlusions.

The smooth parts of the scene (the ball and the pins), are matched very well with small error (dark regions in absolute error maps in figures 3.4 b), d), f) and h), and in figures 3.5 b) and d). We expect that the probabilistic algo-



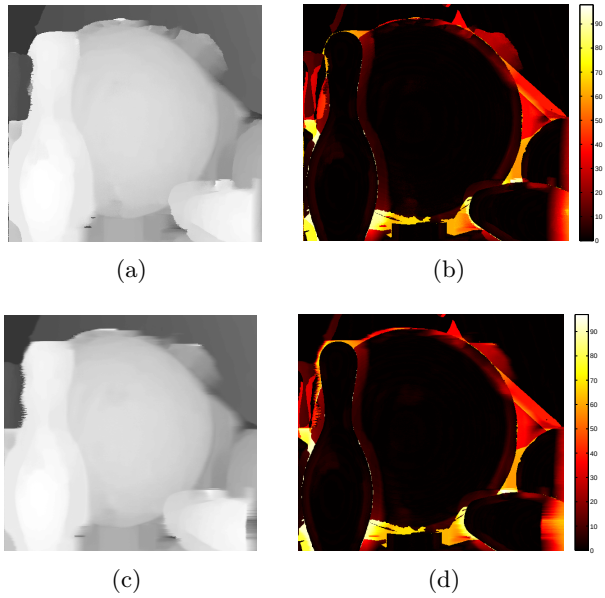
(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 3.4 Estimated disparity maps and their error maps with reference to the ground truth disparity map using dynamic programming (a) and (b), forward algorithm (c) and (d), forward-backward algorithm(e) and (f), and Viterbi algorithm (g) and (h)



(a) (b)

(c) (d)

Figure 3.5 Estimated disparity maps and their error maps with reference to the ground truth disparity map using: (a) and (b) particle filtering, (c) and (d) with smoothing algorithm

rithms can be successfully applied to stereo matching of objects with smooth surfaces and used in e.g. reconstruction of faces.

The quantitative quality comparison is given in Table 3.1. For all cases the percentages of the recovered disparity values which are identical to the ground truth disparities are given in column Q_0, while columns Q_1 and Q_2 show the percentages of the recovered disparities within the range of ± 1 and ± 2 of the ground truth values. The percentage values for the particle filtering result are quite comparable with those of other algorithms. The values along the columns are roughly equal. It would be expected that the algorithms which include the explicit or implicit back propagating step (the forward/backward algorithm, the particle filtering followed by smoothing and the Viterbi algorithm) recovered the disparity map more accurately.

3.6 Conclusion and Further Work

We have demonstrated that HMMs and particle filtering can successfully be applied to stereo matching. In these first experiments, we used the simple

Table 3.1 Quality of disparity maps

Algorithm	Q_0[%]	Q_1[%]	Q_2[%]
dynamic programming	43	67	72
forward alg.	47	68	72
forward/backward alg.	46	68	72
Viterbi alg.	48	69	73
particle filtering (PF)	45	68	72
PF with smoothing	43	66	71

HMM and the likelihood function based on NCC coefficients. The performance of particle filtering is comparable to that of dynamic programming.

Further improvement of the performance of this probabilistic stereo matching approach could be achieved by including more complex knowledge of the scene and a more advanced likelihood function. We expect improvement of the quality of disparity estimation by further exploitation of the flexibility of the particle filtering and by inclusion of scene knowledge by choosing a suitable prior.

4

Comparison of Probabilistic Algorithms Based on Hidden Markov Models for State Estimation [1]

In this chapter, we present an overview of five probabilistic algorithms based on one-dimensional hidden Markov models. We chose HMM as it is suitable for disparity estimation, vary parameters and compare their performance. We compare the forward, forward/backward and Viterbi algorithms, the particle filter and the particle smoother. Next, we apply HMM to stereo matching. We perform stereo matching using a pair of rectified stereo images assuming that the state variable is disparity and that the scene statistics along the epipolar line is described by a first order hidden Markov model. We compare the performance of the five probabilistic algorithms when applied to disparity estimation using the same HMM and then compare this performance to nonprobabilistic dynamic programming.

[1] S. Damjanović, F. van der Heijden and L. J. Spreeuwers, Technical reports: TR s&s 009_08 and TR s&s 010_08, Signals and Systems, University of Twente, 2008

4.1 Introduction

We compare five probabilistic algorithms for state estimation with respect to the number of correctly estimated states of the same discrete sequence generated by a hidden Markov model (HMM) [55], [61]. We compare the forward algorithm, the forward-backward algorithm, the Viterbi algorithm, particle filtering, and particle filtering followed by smoothing.

HMMs describe a state variable evolution over time starting at time instant $i = 0$. The discrete state variable $x(i)$, at time instant i, takes its value from a finite set of states $\Omega = \{\omega_1, ..., \omega_K\}$. The state sequence $x(i)$, $i = 0, 1, ..I$, where $I+1$ is the length of sequence, is hidden and thus not directly observable. The sequence $x(i)$, $i = 0, 1, ..., I$, also satisfies the Markov condition, meaning that the probability of $x(i)$, under the condition of all previous states, is equal to the transition probability

$$P(x(i)|x(0), ..., x(i-1)) = P_t(x(i)|x(i-1)). \tag{4.1}$$

The information about the hidden process $x(i)$, $i = 0, 1, ..., I$, is available through the observation (measurement) sequence $z(i)$, $i = 0, 1, ..., I$, where $z(i)$ takes its value from a finite set $Z = \{\vartheta_1, ..., \vartheta_N\}$. The measurement $z(i)$, $i = 0, 1, ..., I$, of an HMM is memoryless, i.e. the measurement $z(i)$ depends only on $x(i)$ and not on the states at other time instances, and the observation probability is given by

$$P(z(j)|x(0), ..., x(j)) = P_z(z(j)|x(j)). \tag{4.2}$$

The finite-state HMM is defined by the sets Ω and Z, the initial state probability $P_0(x(0))$, the state transition probability $P_t(x(i)|x(i-1))$ and the observation probability $P_z(z(j)|x(j))$. For a discrete finite HMM, the observation probability $P_z(z(j)|x(j))$ is a conditional probability and can be defined by a probability mass function $p_z(z(j)|x(j))$, [62]. $P_z(z(j)|x(j))$ can also be given by a likelihood function.

4.2 Probabilistic Algorithms Based on HMMs for State Estimation

The task of state estimation is to determine the optimal estimate of the state sequence $X(I) = \{x(0), ..., x(I)\}$, based on the measurement sequence $Z(I) = \{z(0), ..., z(I)\}$ of the given HMM. If the state estimation is done online (in real-time), the estimate of the state is done on the base of the previous and current

measurements as in the case of the forward algorithm and the particle filtering. If the estimation is offline, the estimation of the state is done not only using the measurements from previous and current, but also using the measurements which occurred after the time instant of the state being estimated. Offline estimation is done by back propagation through the measurement sequence as in the backward algorithm and in the particle smoother.

4.2.1 Forward Algorithm

The sequence of measurements is processed online to obtain the real-time estimates of the state $x(i)$, using the measurements $Z(i)$ acquired up to and including the time instant i. The solution of the forward algorithm is obtained by maximizing the posterior probability

$$P(x(i)|Z(i)) = \frac{P(Z(i),\, x(i))}{P(Z(i))}, \tag{4.3}$$

which is equivalent to the maximization of the joint probability $P(Z(i),\, x(i))$, because $P(Z(i))$ is constant for a given i. Therefore the maximum a posterior (MAP) estimate is found as

$$\widehat{x}_{\mathrm{MAP}}(i|i) = \arg\max_{k}\{P(Z(i),\, k)\}, \tag{4.4}$$

where the probability $P(Z(i),\, k)$ is calculated by the forward algorithm by means of recursion [55] as

$$\begin{aligned} P(Z(i),\, x(i)) &= \sum_{x(i-1)=1}^{K} P(Z(i),\, x(i),\, x(i-1)) \qquad (4.5)\\ &= \sum_{x(i-1)=1}^{K} P(z(i),\, x(i)|\, Z(i-1), x(i-1))P(Z(i-1),\, x(i-1))\\ &= \sum_{x(i-1)=1}^{K} P(z(i),\, x(i)|\, x(i-1))P(Z(i-1),\, x(i-1))\\ &= P_z(z(i)|\, x(i)) \sum_{x(i-1)=1}^{K} P_t(x(i)|\, x(i-1))P(Z(i-1),\, x(i-1)) \end{aligned}$$

The recursion is initiated with $P(z(0), x(0)) = P_0(x(0))P_z(z(0)|x(0))$. The probability $P(Z(i))$ can be retrieved from $P(Z(i),\, x(i))$ by

$$P(Z(i)) = \sum_{x(i-1)=1}^{K} P(Z(i), x(i)). \tag{4.6}$$

A pseudo-code description of this algorithm is given by algorithm 1. The *forward algorithm* uses the array $F(i,\, x(i)) = \sum_{x(i)=1}^{K} P(Z(i),\, x(i))$ to implement the recursion given by equation (4.6). The computational complexity for i time steps is proportional to $(i+1)K^2$.

Algorithm 1 The Forward Algorithm

Step 1: Initialization $(i = 0)$
$F(0,\, x(0)) = P_0(x(0))P_z(z(0)|x(0))$ for $x(0) = 1, ..., K$

Step 2: Recursion
for $i = 1$ to I **do**
  for $k = 1$ to K **do**

$$F(i,\, x(i)) = P_z(z(i)|\, x(i)) \sum_{x(i-1)=1}^{K} F(i-1, x(x-1))P_t(x(i)|\, x(i-1))$$

$$P(Z(i)) = \sum_{x(i)=1}^{K} F(i,\, x(i))$$

  end for
end for

4.2.2 Backward Algorithm

The offline processing allows for the calculation of the posterior probability $P(x(i)|Z(I))$ and determines the individually most likely states as

$$\hat{x}(i|I) = \arg\max_k \{P(x(i) = \omega_k|Z(I))\}, \tag{4.7}$$

minimizing the error probabilities of the individual states.

The common probabilities $P(x(i),\, Z(i))$ are calculated by the forward algorithm. The *backward algorithm* calculates the probabilities $P(z(i+1),\, ..., z(I)|x(i))$. During each recursion step of the algorithm, the probability $P(z(j), ..., z(I)|x(j-1))$ is recursively derived from $P(z(j+1), ..., z(I)|x(j))$ as

$$P(z(j), ..., z(I)|x(j-1)) = \sum_{x(j)=1}^{K} P_t(x(j)|x(j-1))P_z(z(j)|x(j))P(z(j+1), ..., z(I)|x(j)). \tag{4.8}$$

The backward algorithm starts with $j = I$, and proceeds backwards in time until $j = i+1$. Initialization of the procedure is done by declaring non existing probabilities $P(z(I+1)|x(I))$ equal to 1. The availability of $P(x(i),\, Z(i))$ and

$P(z(i+1),\, ...,\, z(I)|x(i))$ probabilities suffices for the calculation of the posterior probability as

$$\begin{aligned} P(x(i)|\, Z(I)) &= \frac{P(x(i),\, Z(I))}{P(Z(I))} \\ &= \frac{P(z(i+1), ..., z(I)|x(i),\, Z(i))P(x(i),\, Z(i))}{P(Z(I))} \\ &= \frac{P(z(i+1), ..., z(I)|x(i))P(x(i),\, Z(i))}{P(Z(I))} \end{aligned} \tag{4.9}$$

and its maximization by (4.7). A pseudo-code description of this algorithm is given by algorithm 2. The computational complexity of the forward-backward algorithm is of the order of $(I+1)K^2$.

Algorithm 2 The Forward-Backward Algorithm

Step 1: Forward step
Calculate $F(i,\, k),\ i = 0, ..., I,\ k = 1, ...K$ by the forward algorithm as described in Algorithm 1

Step 2: Backward algorithm: Initialization
$B(I,\, k) = 1,\ k = 1, ..., K$

Step 3: Backward algorithm: Recursion
for $i = I - 1$ by -1 to 0 **do**
  for $x(i) = 1$ to K **do**

$$B(i, x(i)) = \sum_{x(i+1)=1}^{K} P_t(x(i)|x(i+1))P_z(z(i+1)|x(i+1))B(i+1, x(i+1))$$

  end for
end for

Step 4: MAP estimation
$\hat{x}_{MAP}(i|I) = \underset{k=1,...,K}{\arg\max}\{B(i,\, k)F(i,\, k)\}$

4.2.3 Viterbi Algorithm

In the process of estimation, the Viterbi algorithm finds the most likely state sequence. The solution maximizes the overall posterior probability:

$$\widehat{x}(0), ..., \widehat{x}(I) = \underset{x(0),...,x(I)}{\arg\max}\ \{P(x(0), ..., x(I))|Z(I)\}\,. \tag{4.10}$$

The computation of this most likely state sequence is performed efficiently by means of a recursion that proceeds forward in time. Taking into account the Markov condition and the memoryless property, the result is obtained by maximization of the common probability over states of the sequence

$$\max_{x(0),\ldots,x(i)} \{P(x(0),...,x(i),x(i+1),Z(i+1))\} = \tag{4.11}$$

$$=\mathrm{P}_z(z(i+1)|x(i+1))\max_{x(i)}\left\{P_t(x(i+1)|x(i))\cdot \max_{x(0),\ldots,x(i-1)}\{P(x(0),...,x(i-1),x(i),Z(i))\}\right\}$$

The value of $x(i)$ that maximizes $P(x(0),...,x(i),x(i+1),Z(i+1))$ is a function of $x(i+1)$:

$$\widehat{x}(i|x(i+1)) = \arg\max_{x(i)}\left\{P_t(x(i+1)|x(i)) \max_{x(0),\ldots,x(i-1)}\{P(x(0),...,x(i-1),x(i),Z(i))\}\right\} \tag{4.12}$$

The Viterbi algorithm uses the recursive equation in (4.11) and the corresponding optimal state dependency expressed in (4.12) to find the optimal path. A pseudo-code description of this algorithm is given by algorithm 3. The computational complexity of the Viterbi algorithm is comparable to that of the forward algorithm and of the order of $(I+1)/K^2$.

4.2.4 Particle Filtering

Estimation of the posterior probabilities or filtering distribution can be achieved using the standard filtering recursions via the Chapman-Kolmogorov equation

$$p(x(i+1)|\,Z(i)) = \int p(x(i)|\,Z(i))p_t(x(i+1)|\,x(i))dx(i) \tag{4.13}$$

and via Bayes rule for the prior update

$$p(x(i+1)|\,Z(i+1)) = \frac{p_z(z(i+1)|\,x(i+1))p(x(i+1)|\,Z(i))}{p(z(i+1)|\,Z(i))}. \tag{4.14}$$

One of the approximation strategies is that of sequential Monte Carlo methods, known as particle filters. Within the particle filter framework, the filtering distribution is approximated with an empirical distribution formed of the point of masses, or particles [60], so the posterior probability distribution is given by the particle approximation as

$$p(x(i)|Z(i)) \simeq \sum_{n=1}^{N_p} w_{i,\,norm}^{(n)}\delta(x(i)-x^{(n)}(i)) \tag{4.15}$$

where $\delta(\cdot)$ is the Dirac delta function, N_p is number of particles and the normalized importance weights $w_{i,\,norm}^{(n)}$ are chosen as

$$w_i^{(n)} = p(z(i)|\,x_i^{(n)}) \quad \text{and} \quad \sum_{n=1}^{N_p} w_{i,\,norm}^{(n)} = 1, \quad w_{i,\,norm}^{(n)} > 0. \tag{4.16}$$

Algorithm 3 The Viterbi Algorithm

Step 1: Initialization ($i = 0$)
for $x(0) = 1, ..., K$ **do**
$Q(0, x(0)) = P_0(x(0))P_z(z(0)|x(0))$
$R(0, x(0)) = 0$
end for

Step 2: Recursion
for $i = 2$ to I **do**
for $x(i) = 1$ to K **do**
$Q(i, x(i)) = \max_{x(i-1)} \{Q(i-1, x(i-1)P_t(x(i)|x(i-1)))\} P_z(z(i)|x(i))$
$R(i, x(i)) = \max_{x(i-1)} \{Q(i-1, x(i-1)P_t(x(i)|x(i-1)))\}$
end for
end for

Step 3: Termination
$P = \max_{x(I)} \{Q(I, x(I))\}$
$\hat{x}(I|I) = \arg\max_{x(I)} \{Q(I,\, x(I))\}$

Step 4: Backtracking
for $i = I - 1$ by -1 to 0 **do**
$\hat{x}(i|I) = R(i + 1, \hat{x}(i + 1, I))$
end for

The number of particles N_p should be sufficiently large in order to properly represent the posterior distribution by means of a set of samples. For the purpose of comparison with the forward algorithm, the state estimation is performed by the MAP criterion. The processing time and number of operation necessary for particle filtering is directly proportional to $I \cdot N_p$. A pseudo-code description of the particle filtering is given by algorithm 4.

4.2.5 Smoothing

Smoothing can be performed recursively backward in time using the smoothing formula

$$p(x(i)|Z(i)) = \int p(x(i+1)|\, Z(i)) \frac{p(x(i)|\, Z(i)) p_t(x(i+1)|\, x(i))}{p(x(i+1)|\, Z(i))} dx(i+1) \qquad (4.17)$$

Smoothing is applied in addition to particle filtering to generate the realizations of the entire smoothing density $p(X(i)|\, Z(i))$ based on the forward particle filtering results.

Algorithm 4 The Particle Filter

Step 1: Initialization ($i = 0$)
Draw N_p samples $x_i^{(n)}$, $n = 1, ..., N_p$, from the prior probability density $p(x(0))$

Step 2: Update using importance sampling
Set the importance weights values: $w_i^{(n)} = p(z(i)|\, x_i^{(n)})$
Calculate the normalized importance weights: $w_{i,norm}^{(n)} = \frac{w_i^{(n)}}{\sum w_i^{(n)}}$

Step 3: Resample by selection
Calculate the cumulative weights $w_{cum}^{(n)} = \sum w_{norm}$
for $n = 1$ to N_p **do**
  Generate a random number r uniformly distributed in $[0, 1]$
  Find the smallest j such that $w_{cum}^{(j)} \geq r^{(n)}$
  Select $\mathrm{x}_{i,\,selected}^{(n)} = \mathrm{x}_i^{(j)}$
end for

Step 4: Predict
Set $i = i + 1$
for $n = 1$ to N_p **do**
  Draw sample from the density $p(\mathrm{x}(i)|\mathrm{x}(i-1) = x_{i,\,selected}^{(n)}$
end for

Step 5: Go to step 2

A pseudo-code description of particle smoothing is given by algorithm 5, [60].

4.3 Experiments and Discussion

We defined the probabilistic framework for stereo matching using HMMs in Section 3.2. The state transition probabilities represent the assumed change of the disparity along the epipolar line. However, the observation probabilities are not known, so we used the likelihood function instead. In order to gain the insight into a performance of different probabilistic algorithms when the observation probabilities are known, we now perform experiments on random generated sequence with known state transition probabilities and known observation probabilities. We chose the transition probabilities in the manner described in Chapter 3.

First, we set up a number of experiments on randomly generated state sequence

Algorithm 5 Smoothing Algorithm

Step 1: Particle filter
Perform a forward sweep of particle filtering described in algorithm 4, generating weighted particles $\left\{x_i^{(n)},\, w_i^{(n)},\, i = 1, ..., I,\, n = 1, ..., N_p\right\}$

Step 2: Initialization $\left\{\widetilde{x}_I^{(m)},\, \widetilde{w}_I^{(m)},\, m = 1, ..., M\right\}$
Generate M random numbers $n_m \in [1, N_p]$ of multinomial distribution given by $w_I^{(n)}$, $n = 1, ..., N_p$ so that $\widetilde{x}_I^{(m)} = x_I^{(n_m)}$ and $\widetilde{w}_I^{(m)} = w_I^{(n_m)}$, set $x_{smooth}(I) = E[\widetilde{x}_I^{(m)}]$

Step 3: Recursion
for $i = I - 1$ by -1 to 0 **do**

Calculate $w_{i|i+1}^{(m,n)} = w_i^{(n)} p_t(\widetilde{x}_{i+1}^{(m)} | x_t^{(n)}),\ for\, \forall n, m$

Chose $\left\{\widetilde{x}_i^{(m)},\, \widetilde{w}_i^{(m)},\, m = 1, ..., M\right\}$ by generating M random numbers $n_m \in [1, N_p]$ of multinomial distribution given by $w_{i|i+1}^{(m,n)}$, $n = 1, ..., N_p$, $m = 1, ..., M$

$x_{smooth}(i) = E[\widetilde{x}_i^{(m)}]$

end for

using HMMs with known state transition probabilities and known observation probabilities. We investigated the influence on the performance of algorithms when observation probabilities are identical or when they differ from transition state probabilities. We performed state estimation using different probabilistic algorithms and compared their performance. We defined the algorithm performance through a success estimation rate, i.e. the ratio between the number of correctly estimated states and the total number of states that had to be estimated. Second, we applied HMM to estimation of a disparity map of a stereo pair and compare the estimated disparity maps with respect to the dynamic programming (DP) result. We have introduced dynamic programming in Section 3.4.

The objective of the experiments was to compare the rate of successful state estimation of the HMM sequence when using different algorithms. The measurement sequences of the length $I = 1000$ were generated from several different HMMs with known state sequence for the quality assessment of the estimation. We specifically chose the state-space model in order to allow for all state transitions, supporting more small changes of the states variable, but allowing for greater changes, but with small

probabilities, as well. We conducted four experiments with number of states $K = 40$ in experiment 1, and with number of states $K = 100$ in experiments 2, 3, 4 and 5.

The transition probabilities are given in the form of KxK matrix whose element in the m^{th} row and n^{th} column is the transition probability $P_t(x(i+1) = n|\, x(i) = m)$. The sum of all transition probabilities for a fixed state $x(i)$ is equal to 1, $\sum_{x(i+1)=1}^{K} P_t(x(i+1)|x(i)) = 1$. The discrete state variable $x(i)$ can take values from the set $\Omega = \{1, 2, ..., K\}$ and the measurements $z(i)$ can take their value from the same set of values i.e. $Z = \Omega$. The initial probability, at time instant $i = 0$, is taken as $P_0(x(0)) = P_t(x(1)|x(0) = 1)$. The probability that the state variable changes its value in the next time instant is inversely proportional to the absolute difference of the consecutive state values. The state variable has the highest probability of keeping the same value and that probability is the same for all states. The probability linearly decreases with absolute discrepancy of the consecutive variable values up to $transmax_t$. The probability $P_{t,jump}$ that the absolute discrepancy between two state-variable values lies in the range of $[transmax_t,\ jumpmax_t]$ is constant and low e.g. in experiment 1, $P_{t,jump} = 0.1$, while the probability $P_{t,outlier}$ that the absolute discrepancy between two consecutive values of the state variable is higher than $jumpmax_t$ is constant and very low e.g. in experiment 1, $P_{t,outlier} = 0.05$. Figure 4.1 illustrates the shape of the transition probability $P_t(x(i+1)|x(i) = 30)$. The generating formula for transition probabilities, and observation probabilities, probabilities is

$$P_t(x_i|x_j) = \begin{cases} \frac{(1-P_{t,outlier}-P_{t,jump})\cdot(1+tm_t-|x_i-x_j|)}{1+tm_t+\sum_{r=1}^{tm_t} 2\cdot r} & : \quad \text{if } 0 \leqslant |x_i - x_j| \leqslant tm_t \\ \frac{P_{t,jump}}{2\cdot(jm_t-tm_t)} & : \quad \text{if } tm_t < |x_i - x_j| \leqslant jm_t \\ \frac{P_{t,outlier}}{K-1-2\cdot jm_t} & : \quad \text{otherwise} \end{cases}, \tag{4.18}$$

where jm_t stands for $jumpmax_t$ and tm_t stands for $transmax_t$.

The likelihood function, or observation probability, is defined by a matrix P_z. In experiment 1, 2, and 3, P_z is taken to be equal to their corresponding matrix P_t. The matrix element $P_z(z(i) = n|\, x(i) = m)$ is the likelihood that the state at time i, $x(i) = m$, yields measurement $z(i) = n$. In experiment 4, the influence of the outliers to the estimation success rate is examined and we vary the probability $P_{z,outlier}$ parameter in the observation matrix. In experiment 5, the influence of the different observation probabilities for the fixed transition probabilities is considered and we vary parameter $transmax_z$. The overview of the parameters of P_t and P_z is given in table 4.1.

In each experiment, we generate a random sequence using a random number generator and a transition probability matrix P_t. Then, we estimate the sequence by five probabilistic algorithms using a observation probability matrix P_z. The probabilistic algorithms used for the state estimation are: the forward algorithm, the forward-backward-algorithm, the Viterbi algorithm, the particle filter and the particle filter with smoother. We also investigate the influence of the number of particles used in the particle filter, N_p, and the number of particles used in particle smoother, M, to the estimation success rate.

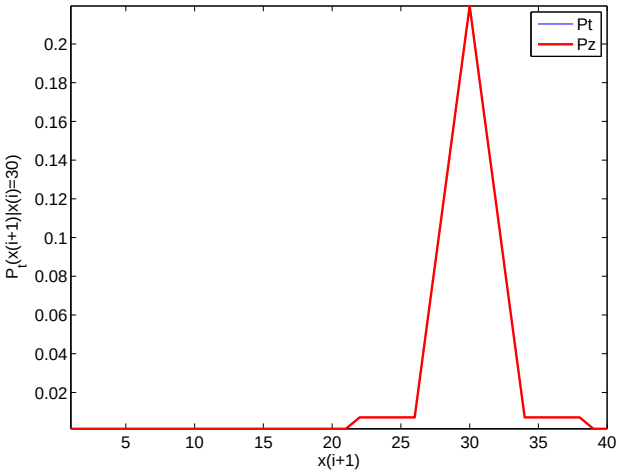


Figure 4.1 $P_t(x(i+1)|x(i) = 30)$ of HMM in experiment 1 with $transmax_t = 4$, $jumpmax_t = 8$, $P_{t,jump} = 0.1$ and $P_{t,outlier} = 0.05$, $P_z = P_t$

		P_t				P_z			
	K	$transmax_t$	$jumpmax_t$	$P_{t,jump}$	$P_{t,outlier}$	$transmax_z$	$jumpmax_z$	$P_{z,jump}$	$P_{z,outlier}$
exp. 1	40	4	8	0.1	0.05	4	8	0.1	0.05
exp. 2	100	4	8	0.1	0.05	4	8	0.1	0.05
exp. 3	100	2	4	0.1	0.05	2	4	0.1	0.05
exp. 4	100	2	4	0.1	$0.01 \cdot k$ $k = 0, 1, 2, ..., 15$	2	4	0.1	$n \cdot P_{t,outlier}$ $n = 1, 2, 4, 6, 8$
exp. 5	100	2	4	0.1	0.05	l $l = 2, 3, ..., 10$	4	0.1	0.05

Table 4.1 HMMs used in experiments

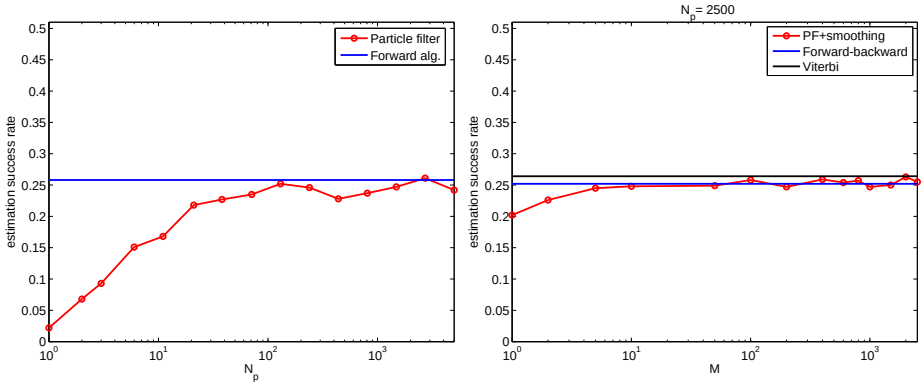


Figure 4.2 Experiment 1: Success rate of estimation
Forward algorithm versus particle filtering; Forward-backward algorithm versus particle filtering with smoothing

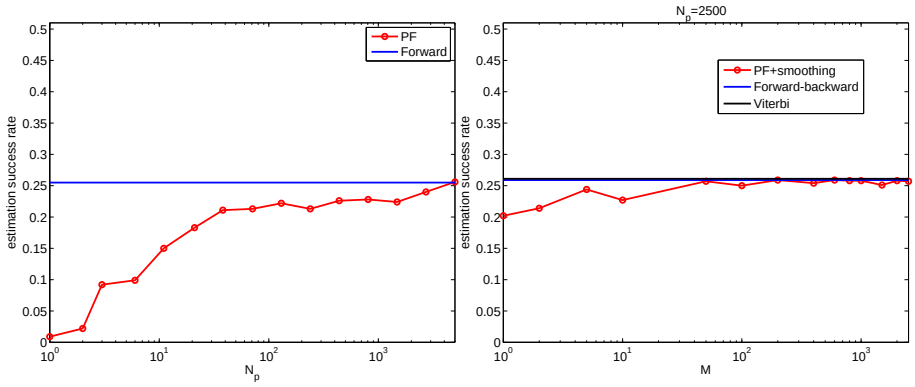


Figure 4.3 Experiment 2: Success rate of estimation
Forward algorithm versus particle filtering; Forward-backward algorithm versus particle filtering with smoothing

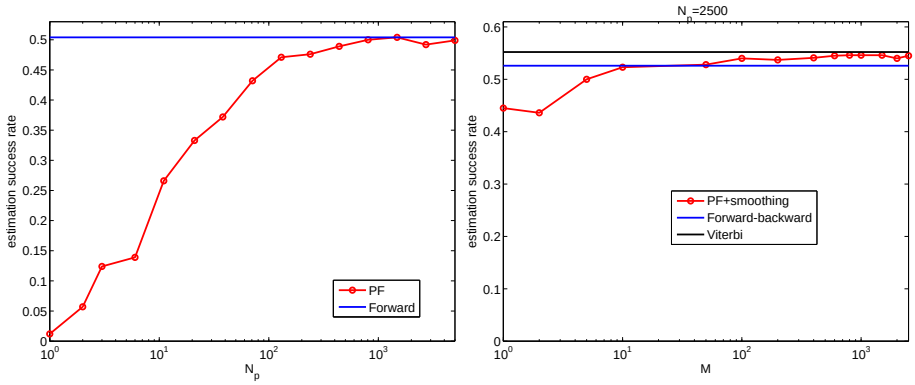


Figure 4.4 Experiment 3: Success rate of estimation
Forward algorithm versus particle filtering; Forward-backward algorithm versus particle filtering with smoothing

In experiment 1, the number of states in the HMM is $K = 40$ and the transition and the observation probability matrix are identical. The results are shown in figure 4.2. The graphs show that the success rate of state estimation by particle filter as the number of used particle is increased asymptotically reaches the rate obtained by the forward algorithm. The smoothing applied after particle filter with $N_p = 2500$, improves the estimation success rate and reaches that of the forward-backward algorithm when the number of particles for smoothing is $M = 10$. Increasing the number of particles involved in smoothing M, leads to further increase of the successful estimation rate.

Experiment 2 is equivalent to experiment 1, with the difference that the number of states in the HMM is $K = 100$. The conclusions are the same as for the estimation success rates. The results are shown in figure 4.3.

In experiment 3, the number of states in the HMM is $K = 100$ as in experiment 2, but we have changed the value of $jumpmax_t$ and $jumpmax_z$ from 8 to 4. The result is a twofold increase of the estimation success rate compared to the rates in experiment 2. The results of experiment 3 are shown in figure 4.4.

In experiment 4, we investigate the influence of the outlier probabilities $P_{t,outlier}$ and $P_{z,outlier}$ on the estimation success rate. The parameters $transmax_{t/z} = 2$ and $jumpmax_{t/z} = 6$ with $P_{t/z,jump} = 0.1$ are kept constant. The number of symbols was $K = 100$. The probability $P_{outlier}$ varies from 0 to 0.15 in 0.01 increments, for the cases when the observation matrix P_z is the same as the transition matrix P_t and when the $P_{z,outlier}$ of the observation matrix P_z is 2, 4, 6 and 8 times greater than the outlier probability of the transition matrix $P_{t,outlier}$. The number of particles used in the particle filter and smoother were $N_p = 100$ and $M = 100$. The results are illustrated in figures 4.5, 4.6, 4.7, 4.8 and 4.9. With the increase of the $P_{outlier}$, the estimation success rate decreases approximately linearly and the decrease is steeper for the higher values of $P_{z,outlier}$. The relative difference between the success rate of different algorithms was preserved, except for the particle filtering with smoothing algorithm, which means room for improvement by choosing a different smoothing procedure.

Experiment 5 illustrates the influence of the measurement probability by increasing the $transmax_z$ value, while $transmax_t$ is held constant. From the figure 4.10, we see that as the difference between the likelihood function and the appropriate transition probability $transmax_z$ parameter increases, the success estimation rate becomes lower.

To get an overview of the estimation success rates for different probabilistic algorithms across experiments, we compared representative cases of different experiments in Table 4.2. It can be seen that the best estimation success rates are obtained in experiment 4 with the Viterbi algorithm and the particle filter followed by smoothing. The results show that the estimation success rate highly depends on the parameters of observation probability.

In the next experiment, we estimate the disparity map of a rectified stereo pair, as shown in Figure 4.11, using the five probabilistic algorithms. The ground truth disparity map is not known, so we compared the results by reference to the disparity map calculated by dynamic programming. The observation probabilities are not

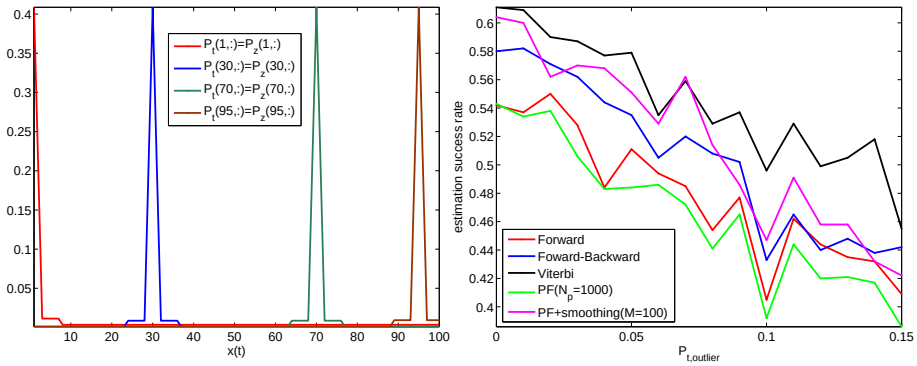


Figure 4.5 Experiment 4: Influence of probability $P_{z,outlier}$, $P_t = P_z$

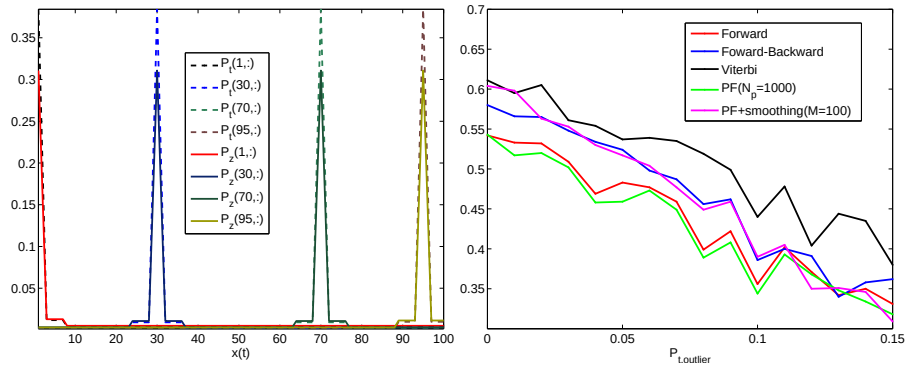


Figure 4.6 Experiment 4: Influence of $P_{z,outlier}$, $P_{z,outlier} = 2 \cdot P_{t,outlier}$

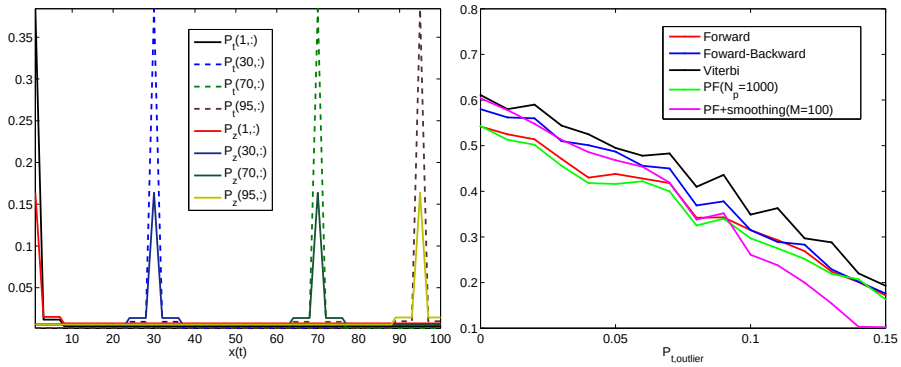


Figure 4.7 Experiment 4: Influence of the $P_{outlier}$, $P_{z,outlier} = 4 \cdot P_{t,outlier}$

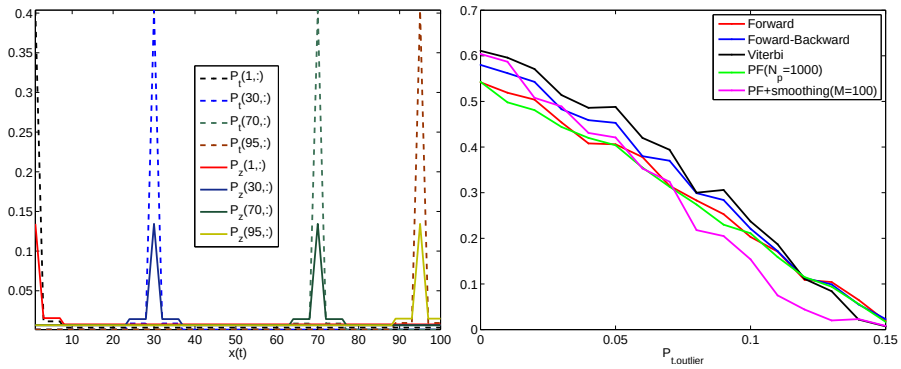


Figure 4.8 Experiment 4: Influence of the $P_{outlier}$, $P_{z,outlier} = 6 \cdot P_{t,outlier}$

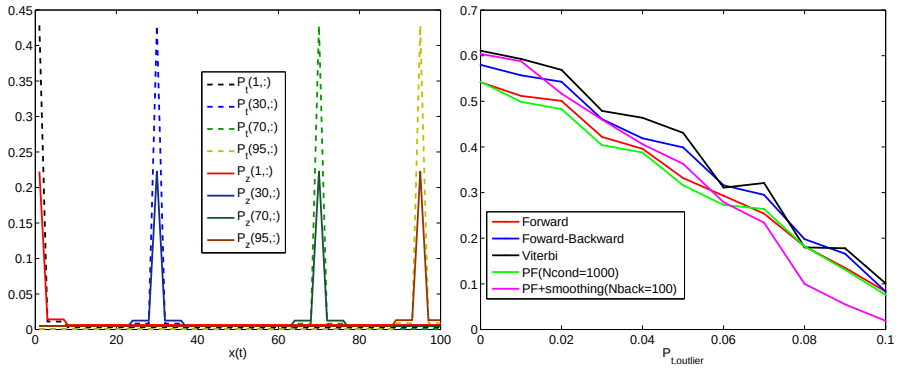


Figure 4.9 Experiment 4: Influence of $P_{outlier}$, $P_{z,outlier} = 8 \cdot P_{t,outlier}$

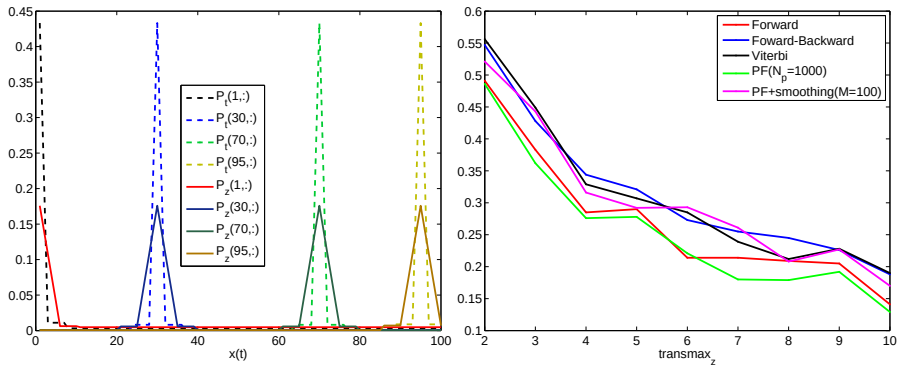


Figure 4.10 Experiment 5: Influence of the $transmax_z$

	Fw	FwBw	Vit	PF	PF+S	k	l	n
exp. 1	0.25	0.25	0.26	0.24	0.25	-	-	-
exp. 2	0.26	0.26	0.26	0.23	0.25	-	-	-
exp. 3	0.51	0.57	0.55	0.50	0.54	-	-	-
exp. 4	0.54	0.58	0.62	0.54	0.61	2	-	2
exp. 5	0.49	0.55	0.56	0.49	0.52	-	2	-

Table 4.2 Experiments comparison: Success estimation rate

known either, so we used the cost and likelihood instead. In DP, the cost function is calculated by using the normalized crosscorrelation of the corresponding windows with the size 31x31 (for a detailed description see Section 3.4). A likelihood function for probabilistic algorithms is also calculated using the normalized crosscorrelation coefficients as in Section 3.4.

The scene in stereo images in figure 4.11 is complex from a disparity estimation point of view: the perspectives in images differ significantly and some scene parts appear in only one image, namely occluded scene parts. However, our probabilistic stereo framework does not model occlusion. The disparity range is 141. For all probabilistic algorithms, an HMM model is used with the number of states $K = 141$ equal to the disparity range and an arbitrarily chosen transition probability matrix Pt as to allow for continuous disparity change and as well as for jumps in disparity. The parameters of P_z are $P_{z,outlier} = 0.05$, $P_{z,jump} = 0.1$, $transmax_z = 4$ and $jumpmax_z = 2$.

In Figure 4.12, we show the estimated disparity maps of stereo images from Figure 4.11 using DP and the five probabilistic algorithm. In Table 4.3, we quantitatively compare the disparity maps as estimated by the probabilistic algorithms to the disparity map as estimated by DP. For all probabilistic algorithms, the percentages of the recovered disparity values which are identical to the DP disparities are given in column Q_0, while columns Q_1 and Q_2 show the percentages of the recovered disparities within the range of ± 1 and ± 2 of the DP disparities. Based on the numbers from the table, we conclude that the Viterbi algorithm gives the results that are the most comparable to the DP result. However, the results of the particle filter and the particle filter followed with smoother are rather low. This can be explained by the discrepancy of the HMM used, as it does not represent the dynamics of the scene accurately. Particle filter is capable of incorporating more complex prior models, but on the other hand is less robust and more prone to errors if the prior is not properly chosen. As other cause of the low particle filter performance could be the choice of the likelihood function, which does not take into account the complexity of the scene and occlusion.

4.4 Concluding remarks

State estimation of sequences of different hidden Markov models has been conducted by the forward algorithm, the forward-backward algorithm, the Viterbi algorithm,

Figure 4.11 Stereo pair: left and right images

Algorithm	Q_0[%]	Q_1[%]	Q_2[%]	Q_5[%]
Forward	55	79	85	90
Forward-Backward	57	85	91	94
Viterbi	73	91	94	95
Particle Filter	33	65	77	86
PF+Ssmoothing	29	56	67	76

Table 4.3 Quality comparison of estimated disparity maps by probabilistic algorithms to disparity map estimated using dynamic programming

particle filtering and the particle filtering followed by smoothing. We compared the success rate of estimation by the forward algorithm against the rate obtained by particle filtering, and the results of forward-backward and Viterbi algorithms against the particle filtering followed by smoothing result. We showed that particle filtering with an increase of the number of particles asymptotically has the same performance as the forward algorithm. As expected, the smoothing improves the correct estimation rate and already when using $M = 10$ particles reaches the quality of the result obtained by the forward-backward algorithm. Also, the influence of the increased outlier probability and the increased $transmax_z$ parameter of the observation function on the success estimation rate has been investigated, and we observed that the estimation rate decreases with an increase of those parameters.

We also demonstrated the disparity estimation using HMM and the five probabilistic algorithms. We compared the results with the DP result as the reference. The disparity map obtained by the Viterbi algorithm proved to be comparable in quality to DP. However, the particle filter and smoother did not achieve estimation quality comparable to the DP result. This can be explained by the scene complexity and the HMM used, which does not completely represent the scene statistics.

There are two possible directions of improvement. One is the use of more complex prior model than HMM, such as two-dimensional Markov random fields, and the other is improvement of the likelihood function.

(a) Dynamic programming (b) Forward algorithm

(c) Forward-backward algorithm (d) Viterbi algorithm

(e) Particle filter (f) Particle smoother

Figure 4.12 Estimated disparity maps

5

A New Likelihood Function for Stereo Matching - How to Achieve Invariance to Unknown Texture, Gains and Offsets? [1]

We introduce a new likelihood function for window-based stereo matching. This likelihood can cope with unknown textures, uncertain gain factors, uncertain offsets, and correlated noise. The method can be fine-tuned to the uncertainty ranges of the gains and offsets, rather than a full, blunt normalization as in NCC (normalized cross correlation). The likelihood is based on a sound probabilistic model. As such it can be directly used within a probabilistic framework. We demonstrate this by embedding the likelihood in a HMM (hidden Markov model) formulation of the 3D reconstruction problem, and applying this to a test scene. We compare the reconstruction results with the results when the similarity measure is the NCC, and we show that our likelihood fits better within the probabilistic frame for stereo matching than NCC.

[1] S. Damjanović, F. van der Heijden, and L. J. Spreeuwers, "A new likelihood function for stereo matching: how to achieve invariance to unknown texture, gains and offsets?", in VISIGRAPP 2009, International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Lisboa, Portugal, pp. 603608, INSTICC Press, February 2009

5.1 Introduction

Stereo correspondence is the process of finding pairs of matching points in two images that are generated by the same physical 3D surface in space, [2]. The classical approach is to consider image windows around two candidate points, and to evaluate a similarity measure (or dissimilarity measure) between the pixels inside these windows. Such an approach is based on the constant brightness assumption (CBA) stating that, apart from noise, the image data in two matching windows are equal. If the noise is white and additive, then the SSD measure (sum of squared differences), or the SAD (sum of absolute differences) is appropriate. Often, the gains and offsets with which the two images are acquired are not equal, and are not precisely known. Therefore, another popular similarity measure is the NCC (normalized cross correlation) which neutralizes these offsets and gains. An alternative is the mutual information, [63], which is even invariant to a bijective mapping between the grey levels of the left and right images.

In a probabilistic approach to stereo correspondence, the similarity measures become likelihood functions being the probability density of the observed data given the ground truth. For the application of stereo correspondence (and related to that motion estimation) several models have been proposed for the development of the likelihood function, but none of them consider the situation of uncertain gains and offsets. In this paper, we introduce a new likelihood function in which the unknown texture, and the uncertainties of gains and offsets are explicitly modelled.

The solution of stereo correspondence is often represented by a disparity map. The disparity is the difference in position between two corresponding points. In the classical approach, the disparity map is estimated point by point on an individual base. Better results are obtained by raising additional constraints in the solution space. For instance, neighbouring disparities should be smooth (except on the edge of an occlusion), unique, and properly ordered. Context-dependent approaches, such as dynamic programming [19] and graph-cut algorithms [64], embed these contextual constraints by raising an optimization criterion that concerns a group of disparities at once, rather than individual disparities. For that purpose, an optimization criterion is defined that expresses both the compliance of a solution with the constraints, and the degree of agreement with the observed image data.

The Bayesian approach has proved to be a sound base to formulate the optimization problem on [19, 21]. Here, the optimization criterion is expressed in terms of probability densities. A crucial role is the likelihood function, i.e. the conditional probability density of the data given the disparities. Suppose that a given point has a disparity x, and that for that particular point and disparity the pixels in the corresponding image windows are given by $\mathbf{z}_1$ and $\mathbf{z}_2$. Then the likelihood function of that point is by definition the pdf $p(\mathbf{z}_1, \mathbf{z}_2|x)$.

The usual expression for this likelihood is again based on the CBA, and assumes Gaussian, additive white noise. Application of this model leads to the following likelihood:

$$p(\mathbf{z}_1, \mathbf{z}_2|x) \propto \exp\left(-\frac{1}{4\sigma_n^2}\left\|\mathbf{z}_1 - \mathbf{z}_2\right\|^2\right) \tag{5.1}$$

Here, $\|\mathbf{z}_1 - \mathbf{z}_2\|^2$ is the SSD. The likelihood function in eq. (5.1) is a monotonically decreasing function of the SSD. It is used by [19] and [21] albeit that both have an additional provision for occluded pixels. However, the function is inappropriate if the gains and offsets are uncertain. Yet, the differences between the grey levels in two corresponding windows is often more affected by differences in gains and offsets than by noise. This paper introduces new expressions which do include these effects. The NCC and the mutual information similarity measures are also invariant to these nuisance factors. However, these measures are parameters derived from the pdfs. But in a true probabilistic approach we really need the pdfs themselves, and not just parameters.

The paper is organized as follows. Section 2 introduces the new likelihood function. Here, a probabilistic model is formulated that explicitly describes the existence of an unknown texture, and uncertain gains and offsets. The final likelihood is obtained by marginalization of these factors. Section 3 analyses the expression that is found for the likelihood. In Section 4, we present some experimental results where the likelihood function is used within a HMM framework. A comparison is made between the newly derived likelihood and the NCC when used in a forward/backward algorithm. Section 5 gives concluding remarks and further directions.

5.2 The Likelihood of two corresponding points

We consider two corresponding points with disparity x. The image data within two windows that surround the two points are represented by $\mathbf{z}_1$ and $\mathbf{z}_2$. The grey levels (or colours) within the windows depend on the texture and radiometric properties of the observed surface patch, but also on the illumination of the surface, and on the properties of the imaging device. We model this by:

$$\mathbf{z}_k = \alpha_k \mathbf{s} + \mathbf{n}_k + \beta_k \mathbf{e} \qquad k = 1, 2 \tag{5.2}$$

Here, $\mathbf{s}$ is the result of mapping the texture on the surface to the two image planes. According to the CBA, this mapping yields identical results in the two images. α_k are the gain factors of the two imaging devices. β_k are the offsets. $\mathbf{e}$ is the all 1 vector. $\mathbf{n}_k$ are noise vectors. We assume Gaussian noise with covariance matrix $\mathbf{C_n}$. Furthermore, we assume that $\mathbf{n}_1$ is not correlated with $\mathbf{n}_2$.

Strictly speaking, the CBA can only hold for fronto-parallel planar surface patches. In all other cases the local geometry of the surface around a point of interest is mapped differently to the two image planes. Thus, the texture on the surface will be observed differently in the images. This problem becomes more distinct as the size of the window increases. The problem can be solved by backmapping the image data within the two windows to the 3D surface before applying the similarity measure, [65]. In the sequel, we will assume that either such a geometric correction has taken place, or that the windows are so small that the aperture problem can be neglected.

In order to get the expression for the likelihood function we marginalize the pdf of $\mathbf{z}_1$ and $\mathbf{z}_2$ with respect to the unknown texture $\mathbf{s}$. Next, we marginalize the resulting expression with respect to the gains α_k. The offsets can be dealt with by regarding

$\mathbf{n}_k + \beta_k \mathbf{e}$ as one additive noise term. Thus, a redefinition of $\mathbf{C_n}$ suffices. This will be looked upon in more detail in Section 5.2.3, but for the moment we can ignore the existence of offsets.

5.2.1 Texture Marginalization

The likelihood function can be obtained by marginalization of the texture:

$$p(\mathbf{z}_1, \mathbf{z}_2|x, \alpha_1, \alpha_2) = \int_{\mathbf{s}} p(\mathbf{z}_1, \mathbf{z}_2|x, \mathbf{s}, \alpha_1, \alpha_2) p(\mathbf{s}|x) d\mathbf{s} \tag{5.3}$$

The pdf $p(\mathbf{s}|x)$ represents the prior pdf of the texture $\mathbf{s}$. For simplicity, we assume a full lack of prior knowledge, thus leading to a prior pdf which is constant within the allowable range of $\mathbf{z}_1$ and $\mathbf{z}_2$. This justifies the following simplification:

$$p(\mathbf{z}_1, \mathbf{z}_2|x) = K \int_{\mathbf{s}} p(\mathbf{z}_1, \mathbf{z}_2|x, \mathbf{s}, \alpha_1, \alpha_2) d\mathbf{s} \tag{5.4}$$

K is a normalization constant that depends on the width of $p(\mathbf{s})$. Any width will do as long as $p(\mathbf{s})$ covers the range of interest of $\mathbf{z}_1$ and $\mathbf{z}_2$. Therefore, K is undetermined. This is not really a limitation since K does not depend on x, $\mathbf{z}_1$ or $\mathbf{z}_2$.

With $\mathbf{s}$ fixed, $\mathbf{z_1}$ and $\mathbf{z_2}$ are two uncorrelated, normal distributed random vectors with mean $\mathbf{s}$, and covariance matrix $\mathbf{C_n}$. Therefore $p(\mathbf{z}_1, \mathbf{z}_2|x, \alpha_1, \alpha_2) = G(\mathbf{z}_1 - \alpha_1\mathbf{s})G(\mathbf{z}_2 - \alpha_2\mathbf{s})$, where $G(\cdot)$ is a Gaussian distribution with zero mean and covariance matrix $\mathbf{C_n}$. This expression can be further simplified by the introduction of two auxiliary variables: $\mathbf{h} \equiv \frac{\mathbf{z}_1}{\alpha_1} - \mathbf{s}$ and $\mathbf{y} \equiv \frac{\mathbf{z}_1}{\alpha_1} - \frac{\mathbf{z}_2}{\alpha_2}$ so that $\mathbf{h} - \mathbf{y} = \frac{\mathbf{z_2}}{\alpha_2} - \mathbf{s}$. The likelihood function can be obtained by substitution:

$$p(\mathbf{z_1}, \mathbf{z_2}|x, \alpha_1, \alpha_2) = K \int_{\mathbf{h}} G(\alpha_1\mathbf{h}) G(\alpha_2(\mathbf{h} - \mathbf{y})) d\mathbf{h}$$

and by rewriting this in the Gaussian form:

$$p(\mathbf{z}_1, \mathbf{z}_2|x, \alpha_1, \alpha_2) \propto \frac{1}{\sqrt{\alpha_1^2 + \alpha_2^2}} \exp\left(- \frac{(\alpha_2\mathbf{z}_1 - \alpha_1\mathbf{z}_2)^{\mathrm{T}} {\mathbf{C_n}}^{-1} (\alpha_2\mathbf{z}_1 - \alpha_1\mathbf{z}_2)}{2(\alpha_1^2 + \alpha_2^2)} \right) \tag{5.5}$$

Note that for $\alpha_1 = \alpha_2 = 1$ and $\mathbf{C_n} = \sigma_n^2\mathbf{I}$ the likelihood simplifies to eq. (5.1). The resulting likelihood function is the same as in [19, 21] although the models on which the expression is based differ.

5.2.2 Marginalization of the Gains

In order to neutralize the unknown gains we marginalize over α_1 and α_2:

$$p(\mathbf{z}_1, \mathbf{z}_2|x) = \int_{\alpha_1} \int_{\alpha_2} p(\mathbf{z}_1, \mathbf{z}_2|x, \alpha_1, \alpha_2) p(\alpha_1) p(\alpha_2) d\alpha_2 d\alpha_1 \tag{5.6}$$

The prior pdfs $p(\alpha_k)$ should reflect the prior knowledge about the gains α_k. Usually, the gain factors do not deviate too much from 1. For that reason, we chose for $p(\alpha_k)$ a normal distribution, centred around 1, and with standard deviations σ_α. In order to make the analytical integration of eq. (5.5) possible, we approximate the term $1/(\alpha_1^2 + \alpha_2^2)$ by its value at $\alpha_k = 1$, that is $\frac{1}{2}$. This approximation is rough, but not too rough. For $\alpha_k < 1$, the factor $1/(\alpha_1^2 + \alpha_2^2)$ is underestimated, but for $\alpha_k > 1$ it is overestimated. Since the integration takes place on both side of $\alpha_k = 1$, the error is partly compensated for.

Under the assumption $\alpha_k \sim N(1, \sigma_\alpha)$, the approximation leads to the following result:

$$p(\mathbf{z}_1, \mathbf{z}_2|x) \propto \frac{\exp\left(-\frac{\sigma_\alpha^2(\rho_{11}\rho_{22}-\rho_{12}^2)+\rho_{11}+\rho_{22}-2\rho_{12}}{\sigma_\alpha^4(\rho_{11}\rho_{22}-\rho_{12}^2)+2\sigma_\alpha^2(\rho_{11}+\rho_{22})+4}\right)}{\sqrt{\sigma_\alpha^4(\rho_{11}\rho_{22} - \rho_{12}^2) + 2\sigma_\alpha^2(\rho_{11} + \rho_{22}) + 4}} \tag{5.7}$$

where:

$$\rho_{k\ell} = \mathbf{z}_k^T \mathbf{C}_\mathbf{n}^{-1} \mathbf{z}_\ell \quad \text{with :} \quad k, \ell = 1, 2 \tag{5.8}$$

In the limiting case, as $\sigma_\alpha \to 0$, we have

$$p(\mathbf{z}_1, \mathbf{z}_2|x) \propto \exp\left(-\tfrac{1}{4}\left(\rho_{11} + \rho_{22} - 2\rho_{12}\right)\right) \tag{5.9}$$

which coincides with eq. (5.1). Intuitively, this is correct since the uncertainties about α_1 and α_2 is zero then. In the other limiting case, as $\sigma_\alpha \to \infty$, the likelihood becomes:

$$p(\mathbf{z}_1, \mathbf{z}_2|x) \propto \frac{1}{\sqrt{\rho_{11}\rho_{22} - \rho_{12}^2}} \tag{5.10}$$

We will analyse these expressions further in Section 3.

5.2.3 Neutralizing the Unknown Offsets

We assume that the offsets β_k have a normal distribution with zero mean, and standard deviation σ_β. The vectors $\beta_k \mathbf{e}$ have a covariance matrix $\sigma_\beta^2 \mathbf{e}\mathbf{e}^T$. Since the random vectors are additive, we may absorb them in the noise vectors $\mathbf{n}_k$. Effectively this implies that the covariance matrix $\mathbf{C}_\mathbf{n}$ now becomes $\mathbf{C}_\mathbf{n} + \sigma_\beta^2 \mathbf{e}\mathbf{e}^T$. Consequently, the variable $\rho_{k\ell}$ in eq. (5.8) should be redefined by $\rho_{k\ell} = \mathbf{z}_k^T (\mathbf{C}_\mathbf{n} + \sigma_\beta^2 \mathbf{e}\mathbf{e}^T)^{-1} \mathbf{z}_\ell$ This can be rewritten in:

$$\rho_{k\ell} = \mathbf{z}_k^T \left(\sum_{n=1}^{N} \mathbf{v}_n \lambda_n^{-1} \mathbf{v}_n^T \right) \mathbf{z}_\ell \tag{5.11}$$

λ_n are the eigenvalues of the covariance matrix. $\mathbf{v}_n$ are the corresponding eigenvectors. Suppose that $N\sigma_\beta^2$ is large relative to all other eigenvalues of $\mathbf{C}_\mathbf{n}$ (N is the dimension of $\mathbf{z}_k$). In case of white noise, the equivalent assumption is $N\sigma_\beta^2 \gg \sigma_n^2$). Then one of the eigenvalues of $\mathbf{C}_\mathbf{n} + \sigma_\beta^2 \mathbf{e}\mathbf{e}^\mathbf{T}$ is close to $N\sigma_\beta^2$, while all other eigenvalues are considerably smaller. The eigenvector that corresponds to σ_β^2 is close to $\mathbf{e}$. The contribution of this particular eigenvalue/eigenvector to $\rho_{k\ell}$ in eq. (5.11) is about:

$$\frac{\mathbf{z}_k^T \mathbf{e}\mathbf{e}^T \mathbf{z}_\ell}{N\sigma_\beta^2}. \tag{5.12}$$

The limit case, $\sigma_\beta \to \infty$, represents the situation of full lack of prior knowledge of the offsets. In this circumstance, the approximations above become exact. Thus, the full contribution in eq. (5.12) becomes zero.

There is no need to embed $\sigma_\beta^2 \mathbf{e}\mathbf{e}^\mathbf{T}$ in $\mathbf{C_n}$. The factor $\mathbf{z}_k^T\mathbf{e}$ is the projection of $\mathbf{z}_k$ on $\mathbf{e}$. We just need to remove this projection from $\mathbf{z}_k$ beforehand, and then its contribution is zero anyhow. This can be obtained by subtracting the average of the elements of the vector. Thus, if $\bar{z}_k$ is the average of the elements of the vector $\mathbf{z}_k$, then:

$$\rho_{k\ell} = (\mathbf{z}_k - \bar{z}_k\mathbf{e})^T\mathbf{C}_\mathbf{n}^{-1}(\mathbf{z}_\ell - \bar{z}_\ell\mathbf{e}) \tag{5.13}$$

Note that this approach to cope with unknown offsets is equivalent to the normalization of the mean, just as in the NCC procedure.

5.3 Likelihood Analysis

In this section, we examine the behaviour of the proposed likelihood in different circumstances. For simplicity, we consider only the white noise case, $\mathbf{C_n} = \sigma_n^2\mathbf{I}$. First we examine the behaviour of the likelihood function under the null hypothesis with varying α_1. Other parameters are kept constant. Substitution of eq. (5.2) in eq. (5.8) yields:

$$\begin{aligned} \rho_{kk} &= (\alpha_k^2\mathbf{s}^T\mathbf{s} + 2\alpha_k\mathbf{s}^T\mathbf{n}_k + \mathbf{n}_k^T\mathbf{n}_k)/\sigma_n^2 \\ \rho_{12} &= (\alpha_1\alpha_2\mathbf{s}^T\mathbf{s} + \alpha_1\mathbf{s}^T\mathbf{n}_1 + \alpha_2\mathbf{s}^T\mathbf{n}_2 + \mathbf{n}_1^T\mathbf{n}_2)/\sigma_n^2 \end{aligned} \tag{5.14}$$

We regard $\mathbf{s}$ as a nonrandom signal. The energy σ_s is defined as $\sigma_s^2 \equiv \mathbf{s}^T\mathbf{s}/N$. We examine the behaviour by replacing the inner products in eq. (5.14) by their root mean squares. That is:

$$\begin{aligned} \mathbf{s}^T\mathbf{n}_k &\sim \sqrt{\mathrm{E}\left[\left(\mathbf{s}^T\mathbf{n}_k\right)^2\right]} = \sqrt{N}\sigma_s\sigma_n \\ \mathbf{n}_k^T\mathbf{n}_k &\sim \sqrt{\mathrm{E}\left[\left(\mathbf{n}_k^T\mathbf{n}_k\right)^2\right]} = \sqrt{N^2+2N}\sigma_n^2 \\ \mathbf{n}_1^T\mathbf{n}_2 &\sim \sqrt{\mathrm{E}\left[\left(\mathbf{n}_1^T\mathbf{n}_2\right)^2\right]} = \sqrt{N}\sigma_n^2 \end{aligned} \tag{5.15}$$

Figure 5.1 shows the likelihood function $p(\mathbf{z}_1, \mathbf{z}_2|x)$ for $\sigma_\alpha = \infty$, conform eq. (5.10), and for $\sigma_\alpha = 0.1$, conform eq. (5.7) for varying α_1. Of course, a substitution by RMSs is not exact, but nevertheless, the resulting figure gives a good impression of the behaviour. As expected, if σ_α is very large, the likelihood function covers a wide range of α_1. If σ_α is small, then the function is narrowly peaked around $\alpha_1 = 1$.

In order to check whether the new likelihood function is able to distinguish between similar textures and dissimilar textures, we also examined the ratio of the likelihood function under these two different cases. For that purpose, we also considered the alternative model:

$$\mathbf{z}_k = \alpha_k\mathbf{s}_k + \mathbf{n}_k + \beta_k\mathbf{e} \qquad k = 1, 2 \tag{5.16}$$

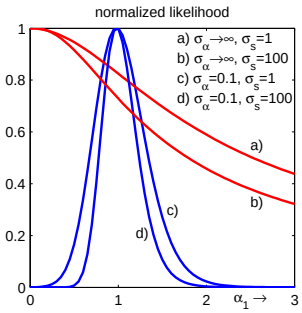


Figure 5.1 The likelihood function with varying α_1. Other parameters are: $\alpha_2 = 1$, $\sigma_n = 1$, $N = 225$

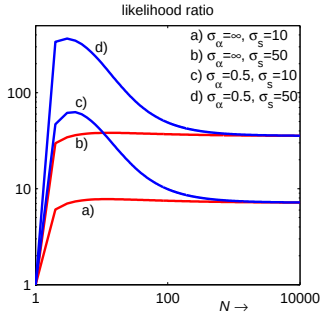


Figure 5.2 The likelihood ratio with varying N. Other parameters are: $\alpha_1 = 1$, $\alpha_2 = 1$, $\sigma_n = 1$

In this situation, $\mathbf{s}_1$ and $\mathbf{s}_2$ are two different textures, but with the same signal energy σ_s. If we model $\mathbf{s}_1$ and $\mathbf{s}_2$ as realizations from two independent random signals, then $\mathrm{E}[(\mathbf{s}_1^T\mathbf{s}_2)^2]^{1/2} = \sigma_s^2\sqrt{N}$. Thus, if the textures are dissimilar, the RMS of the factor $\mathbf{s}^T\mathbf{s}$ in ρ_{12} in eq. (5.14) should be replaced accordingly. The ratio between the likelihoods in the two cases is:

$$\Lambda(\mathbf{z}_1, \mathbf{z}_2) \equiv \frac{p(\mathbf{z}_1, \mathbf{z}_2|x, \text{similar textures})}{p(\mathbf{z}_1, \mathbf{z}_2|x, \text{dissimilar textures})} \tag{5.17}$$

Figure 5.2 shows this ratio for varying N. We see that the ratio's with $\sigma_\alpha = 0.5$ are always larger than the one with $\sigma_\alpha = \infty$, but for large N the ratio's with $\sigma_\alpha = 0.5$ approaches the other one and becomes constant on the long run. The reason for this typical behaviour is that in the factor $\rho_{11}\rho_{22} - \rho_{12}^2$ the contribution of the signal $\alpha_1\alpha_2\mathbf{s}^T\mathbf{s}$ is cancelled out, while the contribution of the noise, i.e. $\mathbf{n}_k^T\mathbf{n}_k$, is proportional to N, and thus keeps growing as N increases.

5.4 Experiments

A preliminary experiment is conducted to demonstrate the abilities of our newly proposed likelihood. For that purpose, two rectified stereo images were selected.

(a) (b) (c)

Figure 5.3 Stereo pair *Bowling1* from [6] : (a) left image, (b) right image, and (c) ground truth disparity

(a) (b) (c)

Figure 5.4 Reconstructed depth maps.: (a) the new likelihood function, (b) NCC based likelihood, and (c) SSD based likelihood

See Figure 5.3. In order to embed the likelihood function within a probabilistic framework, we treat the stereo correspondence along the epipolar line as a state estimation problem using a HMM (Hidden Markov Model). The reconstruction is done using the FwBw (forward-backward) algorithm [55]. The Viterbi algorithm is also applicable, but in our experiments, FwBw outperformed Viterbi. We calculated the disparity map using the new likelihood function as the observation probability, and compared this map with a map obtained from the same HMM, but with an other likelihood function plugged in.

5.4.1 The Hidden Markov Model

Each row of the left image is considered as a HMM. Thus, the running variable is the row index i. The state variable of the HMM is taken to be the disparity x_i. The set of allowable states is: $x_i \in \Omega = \{K_{min}, ..., K_{max}\}$. K_{min} and K_{max} are the minimum and the maximum disparities between the two images. The number of different states is $K = K_{max} - K_{min} + 1$. The transition probabilities between consecutive states are

given by the transition probability $P_t(x_{i+1} = n|\, x_i = m)$.

The disparity x_i along an image row is a piecewise continuous function of i. Sudden jumps are caused by occlusions and boundaries between adjacent objects of different depth in the scene, but for the remaining part the depth tends to be smooth. We can model this prior knowledge by selecting $P_t(x_{i+1} = n|\, x_i = m)$ such that the next state x_{i+1} is likely to be close to the current state x_i. The variable has the highest probability to stay in the same state. The probability should decrease as the absolute difference $\Delta \equiv |x_{i+1} - x_i|$ increases. However, the probability should also allow the large jumps that are caused by occlusions and object boundaries.

In our experiments, P_t consists of two modes. Large jumps are modelled with an overall probability $P_{outlier}$ uniformly distributed over the range $x_i - J_{max}, \cdots, x_i + J_{max}$. In this mode, each state within this range is reached with a probability $P_{outlier}/(2J_{max} + 1)$. Inliers are modelled with an overall probability of $1 - P_{outlier}$. Here, the transition probability linearly decreases with Δ up to where Δ is larger than a threshold T_{max}. We chose $P_{outlier} = 0.05$, $J_{max} = 8$, and $T_{max} = 3$. Note, however, that the choice of P_t could be refined by, for instance, using the uniqueness constraint on the disparities [2].

5.4.2 Reconstruction

The selected rectified stereo pair is shown in Figure 5.3. These images are taken from [33]. The scene, 'bowling1', is chosen because our intention is to apply the algorithm for the reconstruction of textureless and smooth surfaces so that later the application can be extended to the 3D reconstruction of faces. The minimum and maximum disparities of these images are $K_{min} = 374$ and $K_{max} = 446$, which means that the state-space model has $K = 73$ states.

The reconstruction is done by applying the forward-backward algorithm to an HMM with the transition probability described above and with the observation probability given by eq. (5.7). For the calculation of the likelihood expression, we consider that the noise variance is $\sigma_n^2 = 0.05$, the gain variances $\sigma_{\alpha 1}^2 = \sigma_{\bar{\alpha} 1}^2 = 0.25$. We performed the calculations on the pixels within 31x31 windows. Thus, $N = 961$.

The reconstruction is also performed using the NCC as similarity measure. Since this measure is not a probability density, it possibly should undergo a rescaling to make it more suitable for a substitute of the observation probability. After some experimentation, we found that the following mapping of the NCC

$$\left(\tfrac{1}{2}\left(1 + NCC\right)\right)^{\gamma} \tag{5.18}$$

is a suitable choice. The best reconstruction was obtained with $\gamma = 6$. We applied this expression within a HMM with the transition probability described above. The windows that were used are also 31x31.

5.4.3 Results

The reconstructed disparity maps are shown in Figure 5.4. A comparison with the ground truth (Figure 5.3) shows that the reconstruction based on the new likelihood

function is more accurate and more robust than the one based on the NCC measure. The new likelihood expression is better able to deal with, especially, the steplike transitions due to occlusion. The NCC-based result is oversmoothed, and cannot locate this transitions accurately. Note that the large error on the right-hand side of the disparity maps are caused by missing data in the left image.

5.5 Conclusion

We have found an expression for a likelihood function that can cope with unknown textures, uncertain gain factors and uncertain offsets. In contrast to the classical approaches this likelihood is not based on some arbitrary selected heuristics, but on a sound probabilistic model. As such it can be used within a probabilistic framework. The likelihood can be fine-tuned by setting a limited range of allowable gain factors rather than just any gain factor.

Using the model we were able to show that coping with unknown offsets can safely be done by normalizing the means of the data, as done in other approaches such as the normalized correlation coefficient. Unknown gain factors and unknown textures are dealt with in a way that differs a lot from other approaches. Yet, the computational complexity of the proposed metric is quite comparable with, for instance, the computational load of the NCC.

We demonstrated stereo reconstruction within the probabilistic framework by the forward-backward algorithm with a suitably chosen HMM and showed that it is a resourceful approach. We showed that the newly proposed likelihood is more suitable for stereo reconstruction within the probabilistic framework than the NCC. The reconstruction using the new likelihood deals better with occlusion, while the NCC tends to oversmooth the area with greater abrupt change in depth.

6

Sparse Window Local Stereo Matching[1]

We propose a new local algorithm for dense stereo matching of gray images. This algorithm is a hybrid of the pixel based and the window based matching approach and uses a subset of pixels from the large window for matching. Our algorithm does not suffer from the common pitfalls of the window based matching because it successfully recovers disparities of the thin objects and preserves disparity discontinuities. The only criterion for pixel selection is the intensity difference with the central pixel. The subset contains only pixels which lay within a fixed threshold from the central gray value. As a consequence of the fixed threshold, a low-textured windows will use a larger percentage of pixels for matching, while textured windows can use just a few. In this way, this approach also reduces the memory consumption. The cost is calculated as the sum of squared differences normalized to the number of the used pixels. The algorithm performance is demonstrated on the test images from the Middlebury stereo evaluation framework.

[1]S. Damjanovic, F. van der Heijden, and L. J. Spreeuwers, "Sparse window local stereo matching", VISIGRAPP 2011, pp. 689693, 2011

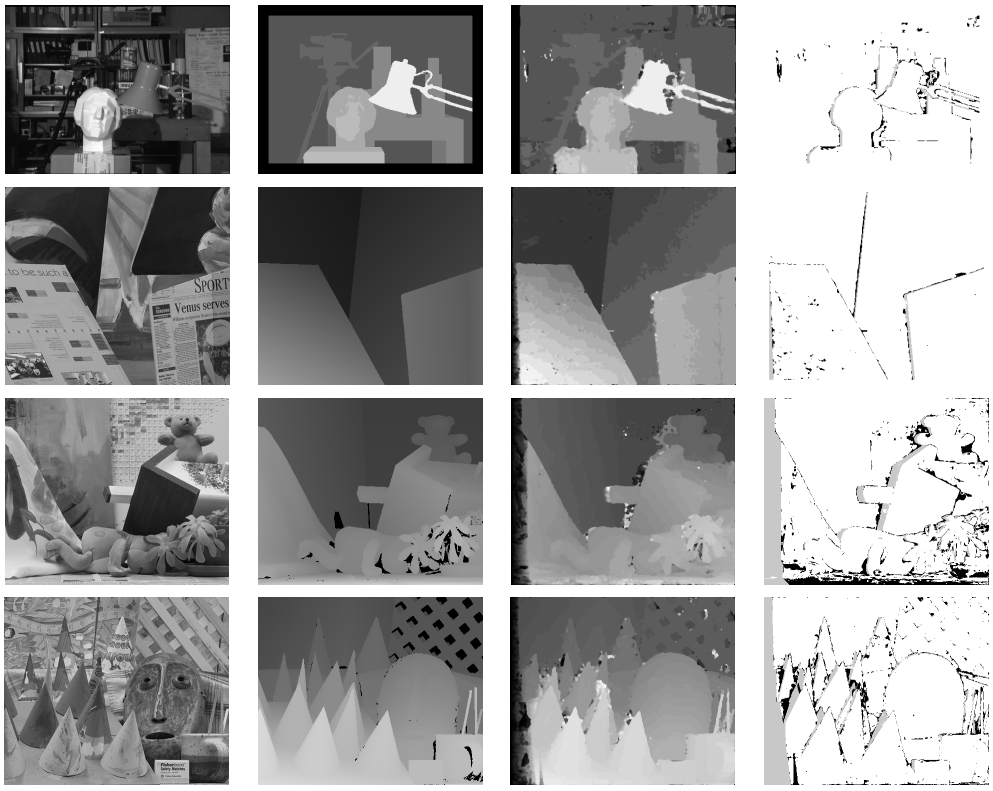


Figure 6.1 Disparity results for the stereo pairs (1st row: Tsukuba, 2nd row: Venus, 3rd row: Teddy, 4th row: Cones) from the Middlebury database. From left to the right columns show: The left image, Ground truth, Result computed by the sparse window matching technique, Disparity errors larger than 1 pixel. The nonoccluded regions errors with ranking (January 2011) are respectively: *Tsukuba* 2.82% (65), *Venus* 1.20% (67), *Teddy* 9.16% (68), *Cones* 5.91% (75)

6.1 Introduction

Stereo matching has been a popular topic of research for almost four decades, ever since one of the first papers appeared in 1979 [7]. A *de facto* evaluation framework for objective comparison of different stereo algorithms has been established [4]. Stereo algorithms can be classified into two categories: local and global. Although the global algorithms are more sophisticated and achieve high accuracy, the local algorithms

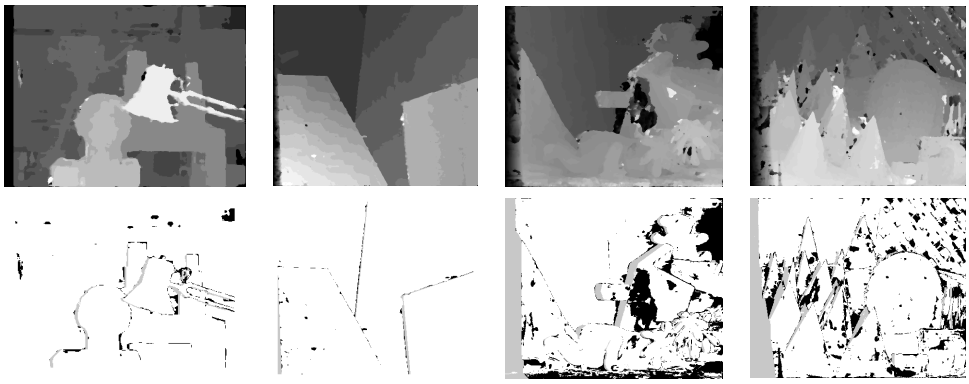


Figure 6.2 Disparity maps calculated by sparse window technique without the offset compensation [the upper row] and their bad pixels maps [the lower row]. The nonoccluded regions errors with ranking (January 2011) are respectively: *Tsukuba* 2.53% (61), *Venus* 0.63% (47), *Teddy* 17.5% (99), *Cones* 13.8% (101)

are more present in the practical computer vision applications because of its low computational load and efficient hardware implementation [42], [66], [67].

In local stereo matching, the cost is aggregated over a support window, which is most often rectangular. It is inherently assumed that all pixels within the matching window have the same disparity; whereas, the fronto-parallel assumption is not valid for e.g. curved surfaces due to perspective distortion and occlusion. Therefore, the window-based matching produces different artifacts in the final disparity map: the discontinuities are smoothed and the disparity of the texture richer surfaces are propagated into the lower textured areas [44]. Another limitation is the dimension of the objects whose disparity can be successfully recovered; the object's height and width in the image should be at least half the size of the window dimensions in order to be detected in the window matching. The idea that properly shaped support areas for cost aggregation can result in better matching result exists in the literature [68], [66], [69].

The ideal window for matching would be only one pixel; however, the one-pixel window does not provide sufficiently discriminatory cost for the local stereo matching. We introduce the hybrid support: a set of properly chosen pixels within the rectangular window in order to combine the support of many pixels for cost aggregation as in the window-based matching but not to be limited by the window dimension like in the pixel-based matching. We use "sparse window" in cost aggregation and the sum of squared differences normalized to the number of pixels (nSSD) for cost aggregation and the *winner takes all* (WTA) in the disparity selection step.

The pixel selection by thresholding is also present in the literature [68], which

presents the area-based matching technique. The point-based matching within the global framework is considered in experiments [70] by explicit modeling the mutual relationships among neighboring points. In both of these approaches [68], [70], RGB images were used; whereas, we use gray valued images.

6.2 Sparse Window Matching

6.2.1 Algorithm Framework

We have a pair of gray valued rectified stereo images I_L and I_R with disparity range D. We recover the disparity map which corresponds to the reference image I_L. In the matching process, we observe the rectangular $W_x \times W_x$, $W_x = 2 \cdot w_x + 1$, windows and select some pixels from the left and right matching windows as a suitable. The pixel from the left matching window declared as suitable is selected for the cost aggregation step only if the pixel at the same position from the right window is also declared as suitable. From the N_p selected pixels in each window, we form two $N_p \times 1$ vectors $\mathbf{z_l}$ and $\mathbf{z_r}$. SSD normalized to the number of pixels N_p is used for the cost calculation. Winner-Takes-All (WTA) method is applied to reliable disparity candidates. In the postprocessing step, we use the common median filter.

6.2.2 Pixel Selection

The *continuity constraint* states that disparity varies smoothly everywhere, except on the small fraction of the area on the boundaries of object where discontinuity occurs [7]. Window based matching methods consider the approximation of the continuity constraint and assume that all the pixels in the window have the same disparity. This approximation is too rough in many cases, e.g. for inclined surface, thin objects, round surfaces. We introduce less restrictive assumption and assume that the pixels with close gray values have the same disparity i.e. we do not assume that all window pixel have the identical disparity but only some. The pixels which are close to the central window pixel in the color space should be used in the cost aggregation step.

We declare the pixel at the position (i, j), $i, j = 1, .., W_x$ in the left window as suitable for matching if its gray value $w_l^{i,j}$ differs from the central pixel's gray value $c_l = w_l^{w_x+1, w_x+1}$ for less than the predefined threshold T_L. The suitable pixels in the right window are chosen in the similar manner. Pixel at the position (i, j), $i, j = 1, .., W_x$ in the right window is declared as suitable for matching if its gray value $w_r^{i,j}$ differs from the central pixel's gray value $c_r = w_r^{w_x+1, w_x+1}$ for less than the predefined threshold T_R. The vectors $\mathbf{z_l}$ and $\mathbf{z_r}$ are formed from the pixels at the position at which pixels in both matching windows are declared as suitable. The pseudo-code of the pixel selection step is given in Algorithm 6.

With the fixed window size W_x and fixed thresholds T_L and T_R we expect the low-textured windows to have a high number participating pixels ($N_p \to W_x^2$) and for rich-textured windows sometimes just a few pixels or even one. In these two extreme cases we introduce additional steps in order to prevent errors. In the case

Algorithm 6 Pixel selection

$N_p = 0$
for $i = 1$ to W_x **do**
  for $j = 1$ to W_x **do**
    if $|w_l^{i,j} - c_l| < T_L$ and $|w_r^{i,j} - c_r| < T_R$ **then**
      $N_p = N_p + 1$
      add $w_l^{i,j}$ to vector $\mathbf{z_l}$
      add $w_r^{i,j}$ to vector $\mathbf{z_r}$
    end if
  end for
end for

of low textured window, we erode the selected pixel mask to prevent the pixels from the neighboring textureless areas with the similar intensities influence the cost. In the case of rich-textured windows with only several pixels selected for matching, we perform dilation in order to prevent errors due to e.g. aliasing.

6.2.3 Cost Aggregation

We assume that the constant brightness assumption (CBA) is satisfied in the process of matching. We expect the corresponding pixels to be very close in intensity values, except for the Gaussian noise with the variance σ_n^2. This expectation is justified by the outlier elimination in the process of pixel selection as explained in the previous subsection 6.2.2. We choose the cost based on the sum of squared differences (SSD) [21], [71]. In order to be able to compare the costs with different number of pixels participating in the matching for the same central pixel, we introduce the SSD cost normalized to the number of pixels N_p:

$$C_{nSSD} \propto \frac{1}{N_p} \cdot \frac{\| \mathbf{z_l} - \mathbf{z_r} \|^2}{4 \cdot \sigma_n^2}. \tag{6.1}$$

The proposed cost eq.(8.8) is not invariant to unknown pixel offsets which can cause erroneous matching result. We deal with unknown offsets by subtracting a constant from vectors $\mathbf{z_l}$ and $\mathbf{z_r}$, [72], by subtracting the central pixel values c_l and c_r from vectors $\mathbf{z_l}$ and $\mathbf{z_r}$:

$$\mathbf{z_l} = \mathbf{z_l} - c_l \cdot \mathbf{e} \tag{6.2}$$
$$\mathbf{z_r} = \mathbf{z_r} - c_r \cdot \mathbf{e} \tag{6.3}$$

where $\mathbf{e}$ is all 1 column vector of the length N_p.

6.2.4 Adjusted WTA and Postprocessing

The WTA method is used to select the optimal disparity $d^{r,c}$ for the pixel at the position (r, c) in the left image. The WTA method takes into account the number of pixels that support the decision by choosing among the trustworthy disparity candidates. The trustworthy disparity candidates have at least $N_s = K_p \cdot \max\{N_p^{r,c}\}$ pixels participating in the cost aggregation, where $N_p^{r,c}$ is $D \times 1$ vector with number of the participating pixels in the cost aggregation for each possible disparity value. K_p is the ratio coefficient $0 < K_p \leq 1$. The optimal disparity $d^{r,c}$ is found as:

$$d^{r,c} = \arg\min_{d_i} \{C_{nSSD}^{r,c}(d_i) |\ N_p^{r,c}(d_i) > N_s\}, \tag{6.4}$$

where $r = 1, \ldots, R$ and $c = 1, \ldots, C$, for the image of the dimension $R \times C$ pixels. The postprocessing step performs median $L \times L$ filtering on the disparity map d to eliminate spurious disparities.

6.3 Experiment Results and Discussion

We have used the Middlebury stereo benchmark [4] to evaluate the performance of the sparse window technique. The parameters of the algorithm are fixed for all four stereo pairs: $T_L = 10$, $T_R = 10$, $w_x = 15$, $W_x = 31$, $\sigma_n^2 = 0.5$. In the process of pixel selection, we declare the window as textureless if in more than $w_x + 1$ columns and in more than $w_x + 1$ rows, more than half pixels from the left window are selected for matching. The structuring element in erosion step is square $N_E \times N_E$, $N_E = 5$. Dilation is performed with squared $N_D \times N_D$, $N_D = 3$ structuring element, if there are less than N_{min} columns with less than N_{min} pixels or if there are less than N_{min} rows with less than N_{min} pixels, $N_{min} = 5$. WTA parameter is $K_p = 0.5$. Postprocessing step is $L \times L$ median filtering with $L = 5$. These parameters have been found empirically.

The disparity maps obtained by our algorithm (with offset compensation) for the stereo pairs from the Middlebury database are shown in the third column in Figure 7.1. The leftmost column contains the left images of the four stereo pairs. Images of the *Tsukuba* stereo pair in the first row are followed by *Venus*, *Teddy* and *Cones*. Ground truth (GT) disparity maps are in the second column. The forth column shows the bad disparity maps where the wrong disparities are shown in black. The occlusion regions are gray and the white regions denote correctly calculated disparity values. The quantitative results in the Middlebury stereo evaluation framework are presented in Table 6.1 which shows the ranking of the results together with the error percentages for the nonoccluded region (NONOCC), error for all pixels (ALL), and the error percentage in the discontinuity region (DISC). We consider the ranking of the NONOCC column most important because we do not deal with the occluded and discontinuity regions in our algorithm although the results show that with our hybrid technique edges of the objects are preserved. The disparities of some narrow structures are successfully detected and recovered, although their dimensions are much smaller than the size of the window. Example of the narrow objects are most noticeable in

Tsukuba disparity map (the lamp reconstruction) and in *Cones* disparity map (pens in a cup in the lower right corner). Whereas, the disparities of the large low textured surfaces in stereo pairs *Venus* and *Teddy* are also successfully recovered with the same sparse window technique.

The images in the Middlebury database have different sizes and disparity ranges, as well as different radiometric properties. The stereo pairs *Tsukuba* (384×288 pixels) and *Venus* (434×383) have disparity ranges from 0 to 15 and from 0 to 19. The radiometric properties of the images in these stereo pairs are almost identical, and our algorithm gives even better results without the offset compensation given by eq. (6.2). The error percentages for the nonoccluded regions for these two pairs without the offset compensation are 2.53% and 0.62% respectively, see Figure 6.2. Figure 6.2 shows in the upper row the disparity maps calculated using the sparse window technique without the offset compensation step for all four stereo pairs from the evaluation framework and the lower row of figure 6.2 contains corresponding bad pixel maps with color coding as in the previous figure. The stereo pairs *Teddy* (450×375 pixels) and *Cones* (450×375) have disparity ranges from 0 to 59. The images of these stereo pairs are not radiometrically identical. The sparse window matching without the offset compensation step results in very large errors, see Figure 6.2. The error percentages for the nonoccluded regions for the stereo pairs *Teddy* and *Cones* without the offset compensation are 17.5% and 13.8% respectively.

Table 6.1 Evaluation results based on the online Middlebury stereo benchmark [4]: The errors are given in percentages for the nonoccluded (NO) region, the whole image (ALL) and discontinuity (DISC) areas. The numbers in the brackets indicate the ranking in the Middleburry table on January 27th, 2011.

Images	NONOCC	ALL	DISC
Tsukuba	**2.82 (65)**	4.68 (73)	11.7 (67)
Venus	**1.20 (67)**	2.87 (77)	12.4 (73)
Teddy	**9.16 (68)**	18.4 (85)	22.1 (77)
Cones	**5.91 (75)**	16.2 (88)	15.0 (79)

6.4 Conclusion

We introduced a new sparse window technique for local stereo matching. The algorithm is simple for implementation, as it is based on pixel selection by thresholding, normalized sum of squared differences cost and plain median filtering in the postprocessing step. Our algorithm does not suffer from the common pitfalls of the window-based matching. It does not use color information as many other algorithms and that may improve results in some cases. Yet, the sparse window local stereo matching produces accurate smooth and discontinuity preserving disparity maps. Although, the presented disparity maps are results of only one left to right matching

and without parameter optimization, they score well in the comparison with other algorithms, outperforming even some global algorithms and algorithms with much more sophisticated segmentation and postprocessing techniques.

We demonstrated that the sparse window matching is promising technique. Our algorithm can be further improved by introducing disparity map refinement and occlusion treatment.

7

Sparse Window Stereo Matching with Optimal Parameters [1]

We proposed a new local stereo matching algorithm for dense matching of gray images. The algorithm is based on selection of a set of pixels from the matching windows which participate in the cost calculation and represents a hybrid approach in between the pixel based and the window based local stereo matching approach. The optimal choice of the window size and the threshold value in sparse window matching is important and depends on the stereo pair properties. We chose the optimal parameters for different stereo pairs and demonstrate the algorithm performance on the test images from the Middlebury stereo evaluation framework.

[1] S. Damjanović, F. van der Heijden, and L. J. Spreeuwers, "Sparse window stereo matching", Proceedings of the International Workshop on Computer Vision Applications, 2011

7.1 Introduction

Stereo matching algorithms can be classified into two categories: local and global [4]. In local stereo matching, the cost is aggregated over a support window which is most often rectangular. It is inherently assumed that all pixels within the matching window have the same disparity. This is not true for e.g. curved surfaces due to perspective distortion and occlusion. Also, the dimension of the objects whose disparity can be successfully recovered depends on the window size: the object's height and width in

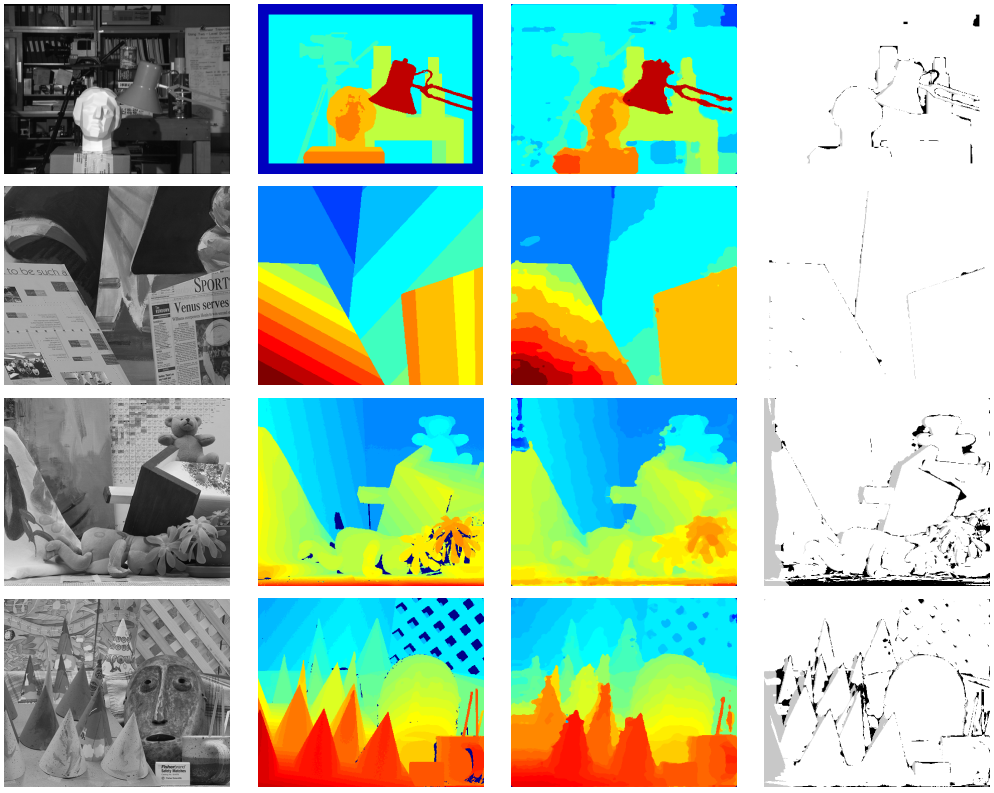


Figure 7.1 Disparity results for the stereo pairs (1st row: Tsukuba, 2nd row: Venus, 3rd row: Teddy, 4th row: Cones) from the Middlebury database [6]. From left to the right columns show: The left image, Ground truth disparity maps, Result computed by the sparse window matching technique with postprocessing, Disparity errors larger than 1 pixel. The nonoccluded regions errors with ranking (March 2011) are respectively: *Tsukuba* 1.88% (47), *Venus* 0.21% (20), *Teddy* 7.31% (44), *Cones* 4.96% (62)

the image should be at least half the size of the window dimensions.

The ideal window for matching would be only one pixel. However, the one-pixel window does not provide sufficiently discriminatory cost for the local stereo matching. In order to combine the support of many pixels for cost aggregation as in the window-based matching but not to be limited by the window dimension like in the pixel-based matching, we introduce the hybrid support: a set of properly chosen pixels within the rectangular window i.e. "sparse window" [73].

We improve the matching results from [73] by choosing the optimal parameters in sparse window matching for different stereo pairs. Stereo pairs from the evaluation framework [6] have different properties. The stereo pairs differ in sizes, disparity range and level of details, see table 7.1. We improve the postprocessing step from [73] by introducing the disparity consistency check and by filling in the missing disparities.

7.2 Sparse window matching

We consider a pair of gray valued, rectified stereo images I_L and I_R with disparity range D. We recover the disparity map which corresponds to the reference image I_L. In the matching process, we observe the rectangular $W \times W$, $W = 2 \cdot w + 1$, windows and select some pixels from the left and right matching windows as a suitable for matching if and only if the pixels lay within the fixed threshold T from the central pixels. The pixel from the left matching window declared as suitable is selected for the cost aggregation step only if the pixel at the same position from the right window is also declared as suitable for matching. From the N_p selected pixels in each window, we form two $N_p \times 1$ vectors. The sum of squared differences normalized to N_p is used for the cost calculation. The adjusted Winner-Takes-All (WTA) method is applied to trustworthy disparity candidates [73].

7.2.1 Parameter selection

We vary the values of the parameters: the window size w and the threshold T. We calculated the disparity error rate for different w and T and chose those which give the smallest errors w.r.t. the ground truth disparity maps, Table 7.1.

7.2.2 Postprocessing

We calculate disparity maps corresponding to the both images of the stereo pair using the optimal parameters. As the first postprocessing step, we apply 5×5 median filter to the both disparity maps. Next, we perform consistency check with the tolerance 1 for the disparity map corresponding to the left image. The inconsistent disparities are filled in by one of the four closest consistent neighbor disparities along vertical or horizontal direction. We chose the disparity of the neighbor pixel with the smallest intensity difference with the pixel with the inconsistent disparity. Finally, we apply 7×7 median filter to the final disparity map.

Table 7.1 Stereo image properties and optimal parameters

Stereo pair	Size	Disparity range	w^{opt}	T^{opt}
Tsukuba	384x288	0 to 15	15	12
Venus	434x383	0 to 19	18	14
Teddy	450x375	0 to 59	12	16
Cones	450x375	0 to 59	15	12

Table 7.2 Error percentages with the Middlebury ranks [6] (March 2011)

Stereo pair	Nonoccluded	Discontinuities	All
Tsukuba	1.88 (47)	3.10 (53)	8.96 (51)
Venus	0.21 (20)	0.71 (33)	2.84 (28)
Teddy	7.31 (44)	14.6 (61)	19.9 (64)
Cones	4.96 (62)	11.9 (62)	13.1 (69)

7.3 Results and Conclusion

Figure 7 shows the resulting disparity maps obtained by our algorithm for the stereo pairs from the Middlebury database. The quantitative results within the Middlebury stereo evaluation framework are presented in Table 7.2. For the stereo pairs *Teddy* and *Cones* we applied the central point subtraction step to compensate for the radiometric differences [72], [73].

The results show that with our hybrid technique edges of the objects are preserved. The disparities of some narrow structures are successfully detected and recovered, although their dimensions are much smaller than the size of the matching window. Such example of the narrow objects are most noticeable in *Tsukuba* disparity map (the lamp reconstruction) and in *Cones* disparity map (pens in a cup in the lower right corner). On the other hand, the disparities of the large low textured surfaces in stereo pairs *Venus* and *Teddy* are also successfully recovered with the same sparse window technique.

In comparison to our previous result in [73], the parameter optimization and the new postprocessing significantly reduced the error rates.

8

Local Stereo Matching Using Adaptive Local Segmentation [1]

We propose a new dense local stereo matching framework for gray-level images based on an adaptive local segmentation using a dynamic threshold. We define a new validity domain of the fronto-parallel assumption based on the local intensity variations in the 4-neighborhood of the matching pixel. The preprocessing step smoothes low textured areas and sharpens texture edges, whereas the postprocessing step detects and recovers occluded and unreliable disparities. The algorithm achieves high stereo reconstruction quality in regions with uniform intensities as well as in textured regions. The algorithm is robust against local radiometrical differences; and successfully recovers disparities around the objects edges, disparities of thin objects, and the disparities of the occluded region. Moreover, our algorithm intrinsically prevents errors caused by occlusion to propagate into nonoccluded regions. It has only a small number of parameters. The performance of our algorithm is evaluated on the Middlebury test bed stereo images. It ranks highly on the evaluation list outperforming many local and global stereo algorithms using color images. Among the local algorithms relying on the fronto-parallel assumption, our algorithm is the best ranked algorithm. We also demonstrate that our algorithm is working well on practical examples as for disparity estimation of a tomato seedling and a 3D reconstruction of a face.

[1] S. Damjanović, F. van der Heijden and L.J. Spreeuwers, "Local Stereo Matching Using Adaptive Local Segmentation", ISRN Machine Vision, 2012

8.1 Introduction

Stereo matching has been a popular topic in computer vision for more than three decades, ever since one of the first papers appeared in 1979 [7]. Stereo images are two images of the same scene taken from different viewpoints. Dense stereo matching is a correspondence problem with the aim to find for each pixel in one image the corresponding pixel in the other image. A map of all pixel displacements in an image is a disparity map. To solve the stereo correspondence problem, it is common to introduce constraints and assumptions, which regularize the stereo correspondence problem.

The most common constraints and assumptions for stereo matching are the epipolar constraint, the constant brightness or the Lambertian assumption, the uniqueness constraint, the smoothness constraint, the visibility constraint and the ordering constraint, [3], [2], [4]. Stereo correspondence algorithms belong to one of two major groups, local or global, depending on whether the constraints are applied to a small local region or propagated throughout the whole image. Local stereo methods estimate the correspondence using a local support region or a window [74] [75]. Local algorithms generally rely on an approximation of the smoothness constraint assuming that all pixels within the matching region have the same disparity. This approximation of the smoothness constraint is known as the fronto-parallel assumption. However, the fronto-parallel assumption is not valid for highly curved surfaces or around disparity discontinuities. Global stereo methods consider stereo matching as a labeling problem where the pixels of the reference image are nodes and the estimated disparities are labels. An energy functional embeds the matching assumptions by its data, smoothness and occlusion terms and propagates them along the scan-line or through the whole image. The labeling problem is solved by energy functional minimization, using dynamic programming, graph cuts or belief propagation [21], [14], [22]. A recent review of both local and global stereo vision algorithms can be found in [67].

Algorithms based on rectangular window matching give an accurate disparity estimation provided the majority of the window pixels belongs to the same, smooth object surface with only a slight curvature or inclination relative to the image plain. In all other cases, window-based matching produces an incorrect disparity map: the discontinuities are smoothed and the disparities of the high-textured surfaces are propagated into low-textured areas [44]. Another restriction of window-based matching is the size of objects of which the disparity is to be determined. Whether the disparity of a narrow object can be correctly estimated depends mostly on the similarity between the occluded background, visible background and object [34]. Algorithms which use suitably shaped matching areas for cost aggregation result in a more accurate disparity estimation, [73], [76], [66], [77], [68], and [75]. The matching region is selected using pixels within certain fixed distances in RGB, CEILab color space and/or Euclidean space.

To alleviate the fronto-parallel assumption, some approaches allow the matching area to lie on the inclined plane, such as in [78] and [79]. The alternative to the idea that properly shaped areas for cost aggregation can result in more accurate matching results is to allocate different weights to pixels in the cost aggregation step.

In [54] the pixels closer in the color space and spatially closer to the central pixel are given proportionally more significance, whereas, in [69], the additional assumption of connectivity plays a role during weight assignment.

Our stereo algorithm belongs to the group of local stereo algorithms. Within the stereo framework, we rely on some standard and some modified matching constraints and assumptions. We use the epipolar constraint to convert the stereo correspondence into an one-dimensional problem. However, we modify the interpretation of the fronto-parallel assumption and the Lambertian constraint. A novel interpretation of the fronto parallel assumption is based on local intensity variations. By adaptive local segmentation in both matching windows, we constrain the fronto-parallel assumption only to the intersection of the central matching segments of the initial rectangular window. This mechanism prevents the propagation of the matching errors caused by occlusion and enables an accurate disparity estimation for narrow objects. The algorithm estimates correctly disparities of both textured as well as textureless surfaces, disparities around depth discontinuities, disparities of the small as well as large objects independently of the initial window size. We apply the Lambertian constraint to local intensity differences and not to the original gray values of the pixels in the segment. In the postprocessing step, we apply the occlusion constraint without imposing the ordering constraint, which enables successful disparity estimation for narrow objects. Also, our stereo algorithm is suitable for a fast real-time implementation, because it is local algorithm for gray-valued images which uses a local segmentation and only a small subset of window pixels for cost calculation.

Our main contribution is the introduction of the relationship between the fronto-parallel assumption and the local intensity variation and its applications to the stereo matching. In addition, we introduce a preprocessing step that smoothes low textured areas and sharpens texture edges producing the image more favorable for a proper local adaptive segmentation.

The paper is organized as follows: in Section 8.2, we explain our stereo matching framework: the preprocessing step, the adaptive local segmentation, the matching region selection, the stereo matching, and the postprocessing step; in Section 8.3, we show and discuss the results of our algorithm on different stereo images; in Section 8.4 we draw conclusions.

8.2 Stereo Algorithm

Our algorithm consists of three steps: a preprocessing step, a matching step and a postprocessing step. The flow chart of the algorithm is shown in Figure 8.1. Input to the algorithm is a pair of rectified stereo images I_l and I_r, where one of them, for instance I_l, is considered as the reference image. For each pixel in the reference image we perform matching along the epipolar line for each integer-valued disparity within the disparity range. Firstly, the input images are preprocessed, as explained in subsection 8.2.1. The preprocessing step is applied to each image individually. Next, we calculate the local intensity variations maps for the preprocessed images and used them to determine the dynamic threshold for adaptive local segmentation, elaborated

in subsection 8.2.2. Further, the stereo matching comprises a final region selection from segments, a matching cost calculation for all disparities from the disparity range and disparity estimation by a modification of the winer-take-all estimation method, see subsection 8.2.3. The result of the matching are two disparity maps, D_{LR} and D_{RL}, corresponding to the left and right images of the stereo pair. Finally, postprocessing step calculates the final disparity map corresponding to the reference image as described in subsection 8.2.4.

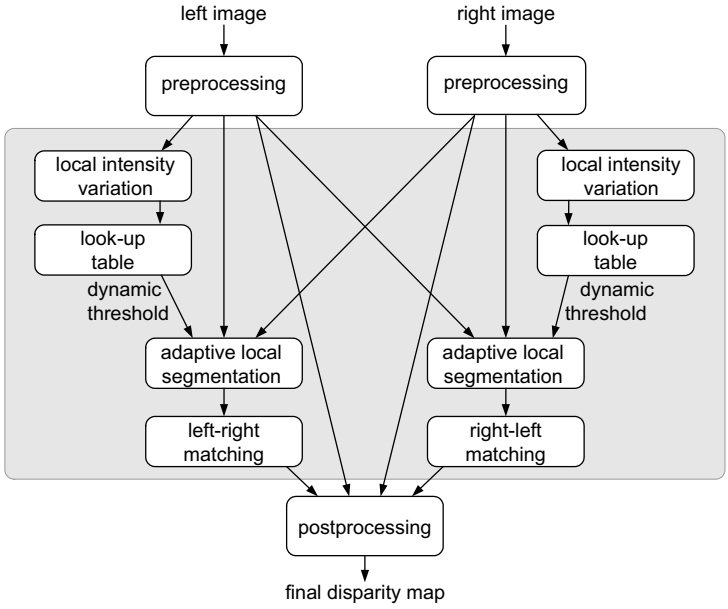
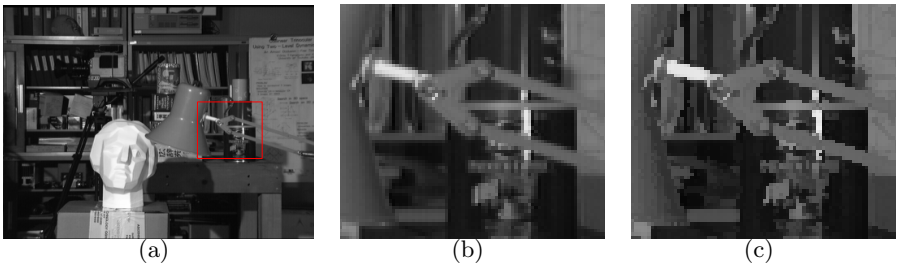


Figure 8.1 Flow chart of the local stereo matching algorithm using adaptive local segmentation

8.2.1 Preprocessing

We apply a nonlinear intensity transformation to the input images in order to make them more suitable for adaptive local segmentation. The presence of the Gaussian noise and the sampling errors in image can produce erroneous segments for matching. The noise is dominant in the low textured and uniform regions, while the sampling errors are pronounced in the high textured image regions. The sampling effects can be tackled by choosing a cost measure insensitive to sampling as in [32], or by interpolating the cost function as in [80]. We handle these problems differently and within the



(a) (b) (c)

Figure 8.2 Illustration of the preprocessing step for one image from *Tsukuba* stereo pair: (a) Original image, (b) Detail of the original image, (c) Detail of the original image after the preprocessing step is applied

preprocessing step. The applied transformation suppresses the noise in low textured regions while simultaneously suppressing the sampling effects in the high textured regions.

The transformation is based on the interpolated subpixel samples by bi-cubic transform in the 4-neighborhood and by consistently replacing the central pixel value by maximum or by minimum value of the set, depending on the relation between the mean and the median of the set. We form a set of samples of the observed pixel at the position (x, y), and the intensities in horizontally and vertically interpolates image at the sub-pixel level at δ_i:

$$\delta_i = -\frac{7}{8} + i \cdot \frac{1}{8}, \quad i \in \{0,\, 1, \cdots,\, 14\}. \tag{8.1}$$

$$v = \{I(x - \delta_i, y),\ I(x, y - \delta_i) |\ \forall i \in \{0,\, 1, \cdots,\, 14\}\ \}. \tag{8.2}$$

The intensity transformation is performed by replacing the intensity $I(x, y)$ with the new intensity as

$$I(x,y) = \begin{cases} \max\{v\} & : \quad if\ \text{median}\{v\} > \text{mean}\{v\} \\ \min\{v\} & : \quad otherwise \end{cases} \tag{8.3}$$

All intensity values are corrected in the same manner. If the pixel intensity differs significantly from its four neighbors, as in the high textured regions, it will be replaced by the maximum value in the interpolated subpixel set v, resulting in the sharpening effect. On the other hand, in low textured regions the intensity change is small and replacing the initial intensity value systematically with the minimum value of the interpolated subpixel set v, produces the favorable denoising effect. These positive effects originate from the image resampling done by bi-cubic interpolation, because the

bi-cubic interpolation exhibits overshoots at locations with large differences between adjacent pixels, see chapter 4.4 in [81] and chapter 6.6 in [82]. These favorable effects are lacking if the interpolation method is linear.

We illustrate the effect of the preprocessing step for an image from a stereo pair from the Middlebury evaluation database in figure 8.2. Therefore, the preprocessing step modifies regions with high intensity variations and results in the sharper image. Further, in section 8.3, we show the influence of this step to overall algorithm score.

8.2.2 Adaptive Local Segmentation

Adaptive local segmentation establishes a new relationship between the local intensity variation and the fronto-parallel assumption applied to stereo matching. Adaptive local segmentation selects a central subset of pixels from a large rectangular window for which we assume that the fronto parallel assumption holds for the segment. The segment contains the central window pixel and pixels, spatially connected to the central pixel, whose intensities lie within the dynamic threshold from the intensity of the central window. Starting from the segment, we form a final region selection for matching, see subsection 8.2.3.

The idea behind the adaptive local segmentation is to prevent that the matching region contains the pixels with significantly different disparities prior to actually estimating disparity. We accomplish this aim by conveniently choosing threshold for segmentation based on the local texture. If local texture is uniform with local intensity variations caused only by the Gaussian noise, we opt for a small threshold value. In this way, because the intensity variations are small, the segment will comprise the whole uniform region. We assume that these pixels originate from the smooth surface of one object and therefore that the fronto-parallel assumption holds for the segment. On the other hand, if the window is textured i.e. intensity variations are significantly larger than the noise level, it is not possible to distinguish based only on the pixel intensities and prior to matching, whether the pixels originate from one textured object or from several different objects at different distances from the camera. In this case, relying on the high texture for an accurate matching result, it is good to select small segment in order to assure that the segment contain pixels from only one object and does not contain depth discontinuity. Due to the high local intensity variations, this is achieved by large threshold.

We introduce local intensity variation measure in order to determine the level of local texture and subsequently the dynamic threshold. We define the local intensity variation measure as a sharpness of local edges in the 4-neighborhood of the central window pixel. The sharper local edges are, the larger the local intensity variation. We calculate the local intensity variation using the maximum of the first derivatives in the horizontal and the vertical directions at the half-pixel interpolated image by benefiting again from overshooting effect of the bi-cubic interpolation.

The horizontal central difference for a pixel at the position (x, y) in image I is calculated as

$$H = \mid I(x - \frac{1}{2}, y) - I(x + \frac{1}{2}, y) \mid, \tag{8.4}$$

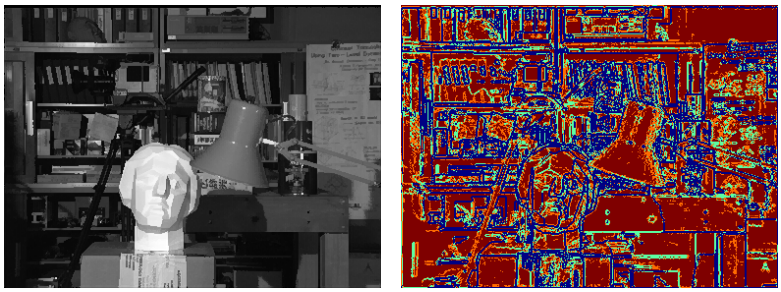


Figure 8.3 Left image from *Tsukuba* stereo pair with a color-coded local intensity variations levels: the lowest local intensity variation is in red, and in the ascending order follow orange, green, the highest local intensity variations are in blue.

where $I(x-\frac{1}{2},y)$ and $I(x+\frac{1}{2},y)$ are horizontal half-pixel shifts of image I to the left and to the right. The vertical central difference for a pixel at the position (x,y) in image I is calculated as

$$V = \mid I(x,y-\frac{1}{2}) - I(x,y+\frac{1}{2}) \mid, \tag{8.5}$$

where $I(x,y-\frac{1}{2})$ and $I(x,y+\frac{1}{2})$ are vertical half-pixel shifts of image I. We define the intensity variation measure as

$$M_t(x,y) = max(V,H). \tag{8.6}$$

We divide local intensity variations into four ranges based on the preselected constant T and define a dynamic threshold for each range by a look-up table :

$$T_d(x,y) = \begin{cases} \frac{T}{2} & : \quad M_t(x,y) \in [0,\frac{T}{4}) \\ \frac{3\cdot T}{4} & : \quad M_t(x,y) \in [\frac{T}{4},\frac{T}{2}) \\ T & : \quad M_t(x,y) \in [\frac{T}{2},T) \\ 2\cdot T & : \quad M_t(x,y) \in [T,\infty) \end{cases} \tag{8.7}$$

Figure 8.3 shows a color-coded dynamic threshold map, or equivalently local intensity variation ranges, for the left image from *Tsukuba* stereo pair from the Middlebury stereo evaluation set, [6].

The dynamic threshold $T_d(x,y)$ defined by equation (8.7) for the reference pixel in the reference image, is also used for the adaptive local segmentation in the non-reference image for all potentially corresponding pixels from the disparity range.

The adaptive local segmentation pseudocode for the reference pixel $I_l(x,y)$ in the left image is given by algorithm 7. The segmentation is performed for reference and non-reference windows independently using the same threshold $T_d(x,y)$. Thus, in the

Algorithm 7 Adaptive local segmentation for reference pixel $I_l(x, y)$

Step 1: Dynamic thresholding
for $i = 1$ to W **do**
  for $j = 1$ to W **do**
    if $|p_{l/r}^{i,j} - c_{l/r}| < T_d(x, y)$ **then**
      set $\mathbf{B}_{l/r}^{i,j}$ to 1
    end if
  end for
end for

Step 2: Dilation
Dilate $\mathbf{B}_{l/r}$ with 3×3 squared structured element

Step 3: Imposing connectivity
for $i = 1$ to W **do**
  for $j = 1$ to W **do**
    if $\mathbf{B}_{l/r}^{i,j} = 1$ and not connected to $\mathbf{B}_{l/r}^{w+1,w+1}$ **then**
      set $\mathbf{B}_{l/r}^{i,j}$ to 0
    end if
  end for
end for

$W \times W$ window, where $W = 2 \cdot w + 1$, around the pixel at the position (x, y) in the reference image, we declare that the pixel at (i, j) position, where $i, j = 1, .., W$ in the reference window, belongs to the segment if its gray value $p_l^{i,j}$ differs from the central pixel's gray value $c_l = p_l^{w+1,w+1}$ for less than the dynamic threshold $T_d(x, y)$. The segment pixels in the non-reference window are chosen in similar way using the same threshold $T_d(x, y)$. Next, the central $8-$connected components in the dilated masks are selected. The final segments are defined by the binary $W \times W$ maps, B_l and B_r, with ones if the the pixels belong to the segment. Dilation is performed by 3×3 squared structured element to include additional neighbor pixels into segments and to merge isolated but close selected pixels.

8.2.3 Stereo Correspondence

The matching region is defined by the overlap of the adaptive local segments in the reference and non-reference windows. Thus, the matching region is defined by binary map B, which has ones if and only if both binary maps, B_l and B_r, have ones at the same positions, as given in algorithm 8.

We assume the corresponding pixels have similar intensities and that the differ-

Algorithm 8 The final binary map calculation

for $i = 1$ to W **do**
  for $j = 1$ to W **do**
    if $\mathbf{B}_l^{i,j} \wedge \mathbf{B}_r^{i,j}$ **then**
      set $\mathbf{B}^{i,j}$ to 1
    end if
  end for
end for

ences exist only due to the Gaussian noise with the variance σ_n^2. One-dimensional vectors, $\mathbf{z}_l$ and $\mathbf{z}_r$, are formed from the pixels from the left and right matching window at positions of ones within the binary map $\mathbf{B}$. Besides the noise, differences between vectors can occur due to different offsets and due to occlusion. To make the matching vectors insensitive to local different offsets, we subtract the central pixel values c_l and c_r from vectors $\mathbf{z_l}$ and $\mathbf{z_r}$, given by algorithm 9. In this way, the intensity information is transformed from the absolute intensities to the differences of intensities with respect to the central window pixels. Further, we impose the Lambertian assumption on the pixels after the central pixel subtraction and not on the original pixel intensities. To prevent the occlusion influence in matching, we eliminate the occlusion outliers by keeping only the coordinates of vectors which differ for less than threshold T as given by algorithm 10.

Algorithm 9 Offset neutralization

N_p' is the length of the vectors $\mathbf{z_l}$ and $\mathbf{z_r}$
c_l and c_r are the central intensities in the left and in the right window
for $i = 1$ to N_p' **do**
  $\mathbf{z}_l(i) = \mathbf{z}_l(i) - c_l$
  $\mathbf{z}_r(i) = \mathbf{z}_r(i) - c_r$
end for

We calculate the matching cost using the sum of squared differences (SSD) [21], [71]. To compare the costs with different length of vectors $\mathbf{z}_l$ and $\mathbf{z}_r$ for different disparities, we introduce the normalized SSD:

$$C_{nSSD}(d) \propto \frac{1}{N_p} \cdot \frac{\| \mathbf{z}_l - \mathbf{z}_r \|^2}{4 \cdot \sigma_n^2}, \tag{8.8}$$

where N_p is the length of vectors $\mathbf{z}_l$ and $\mathbf{z}_r$ for disparity d.

The *winner-take-all* (WTA) method selects the disparity with the minimal cost for the observed reference pixel. In our algorithm, besides the cost, the number of pixels participating in the cost calculation is also an indication of a correspondence.

Algorithm 10 Elimination of the outliers

N'_p is the length of the initial vectors $\mathbf{z}_l$ and $\mathbf{z}_r$
$k = 0$
for $i = 1$ to N'_p **do**
  if $|\mathbf{z}_l(i) - \mathbf{z}_r(i)| \geq T$ **then**
    Remove $\mathbf{z}_l(i)$ and $\mathbf{z}_r(i)$
  end if
end for
N_p is the length of the final vectors $\mathbf{z}_l$ and $\mathbf{z}_r$

This ordinal measure cannot be used directly in the disparity estimation, because it is not always a reliable indication of the correspondence as in the case of occlusion. If the number of pixels used in the cost calculation is very low, it may be due to occlusion. However, a reliable match has a substantial ordinal support.

We combine the cost and the number of participating pixels in the disparity estimation and introduce a hybrid WTA: we consider only disparities supported by a sufficient number of pixels as potential candidates for a disparity estimate. Thus, the final disparity estimate is chosen from a subset of the all possible disparities from the disparity range. We term these disparity candidates as the reliable disparity candidates [73], [83].

The reliable disparity candidates have at least $N_s = K_p \cdot \max\{N_p^{x,y}\}$ supporting pixels, where $N_p^{x,y}$ is a set containing the number of pixels participating in the cost aggregation step for each possible disparity value from the disparity range $[D_{min}, D_{max}]$. K_p is the ratio coefficient $0 < K_p \leq 1$. The estimated disparity $d(x, y)$ is:

$$d(x,y) = \underset{d_i \in \{D_{min},\ldots,D_{max}\}}{\arg\min} \{C_{nSSD}^{x,y}(d_i) |\ N_p^{x,y}(d_i) > N_s\}, \tag{8.9}$$

where $x = 1, \ldots, R$ and $y = 1, \ldots, C$, for image of the dimension $R \times C$ pixels and d_i belongs to the set of all possible disparities from the disparity range $[D_{min}, D_{max}]$.

The final result of the hybrid WTA is the disparity map D

$$D = \{d(x,y) | \forall x \in [1,R] \wedge \forall y \in [1,C]\}. \tag{8.10}$$

We calculate two disparity maps, one disparity map, D_{LR}, with the left image I_l as the reference, and the other, D_{RL}, as the right image I_r as the reference.

8.2.4 Postprocessing

In the postprocessing, we detect the disparity errors and correct them. There are some areas of incorrect disparity values caused by low textured areas larger than the initial window. There are some isolated disparity errors with significantly different

disparity from the neighborhood disparities, so called outliers, caused by isolated pixels or groups of several pixels if the adaptive local segmentation did not result in sufficiently large segment due to high local intensity variation. Also, there are disparity errors caused by occlusion. Although the matching procedure is the same for both occluded and nonoccluded pixels, our stereo matching algorithm does not propagate error caused by occlusions because the boundaries of objects are taken into account by both the adaptive local segmentation and the final matching region selection. However, occluded pixels do not have corresponding pixels and the estimated disparities for the occluded pixels are incorrect.

The post-processing consists of several steps including median filtering of the initial disparity maps, disparity refinement of the individual disparity maps, consistency check and propagation of the reliable disparities.

First, we apply $L \times L$ *median filter* to both disparity maps, D_{LR} and D_{RL}, and eliminate disparity outliers. Second, we refine the filtered disparity maps individually to correct low textures areas with erroneous disparities, in an iterative procedure. The refinement step propagates disparities by histogram voting to the regions with close intensities defined by a look-up table given in equation (8.11) across the whole image as illustrated in propagation scheme in figure 8.4. Some similar notions to this approach appear separately in the literature, [75] and [28], and we were inspired by them. In [28], the cost aggregation is done along the 16 radial directions in disparity space, while in [75], histogram voting is used within the segment for disparity refinement. We refine our disparity maps by histogram voting of accumulating disparities along 8 radial directions across the whole disparity map with constraint of the maximum allowed intensity difference with the pixel being refined. The maximum intensity difference is defined by a dynamic threshold T_p with the same logic behind as in local intensity variation measure in section 8.2.2, with the difference that here we distinguish three ranges of intensity differences. Thus, the histogram is formed using disparities of the pixels with close intensities along 8 radial directions, see figure 8.4 and table 8.1. The pixels are close in intensities and their disparities are taken into account in histogram forming, if they lie within the threshold T_p from the intensity of the pixel at the observed position (x, y). The threshold $T_p(x, y)$ is selected based on a look-up table:

$$T_p(x,y) = \begin{cases} \frac{T}{2} & : \quad M_t(x,y) \in [0, \frac{T}{2}) \\ \frac{3T}{4} & : \quad M_t(x,y) \in [\frac{T}{2}, \frac{3 \cdot T}{4}) \\ T & : \quad M_t(x,y) \in [\frac{3 \cdot T}{4}, \infty) \end{cases} \tag{8.11}$$

The histogram H with a number of bins equal to the number of disparities within the disparity range, is formed by counting the disparities along 8 radial directions for the pixels whose intensity is within threshold $T_p(x, y)$:

$$H(d(x_{tmp}, y_{tmp})) = H(d(x_{tmp}, y_{tmp})) + 1, \text{ if } |I(x_{tmp}, y_{tmp}) - I(x,y)| < T_p(x,y), \tag{8.12}$$

where x_{tmp} and y_{tmp} are given by table 8.1.

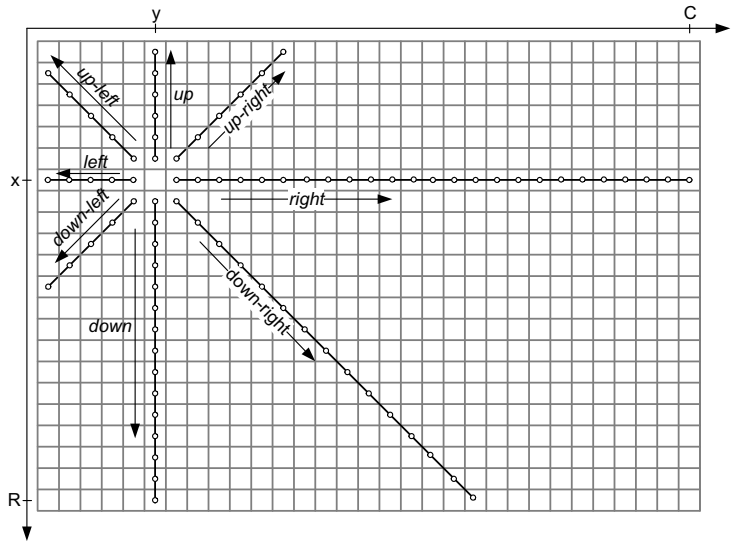


Figure 8.4 Propagation scheme

Table 8.1 x_{tmp} and y_{tmp} values for histogram calculation in equation (8.12)

	direction	x_{tmp}	y_{tmp}	condition
1	up	$x - i_u$	y	$i_u = \{\ 1 \quad \text{to} \quad x-1 \mid\ x-1>0\ \}$
2	up-right	$x - i_{ur}$	$y + i_{ur}$	$i_{ur} = \{\ 1 \quad \text{to} \quad \min(x-1, C-y) \mid\ \min(x-1, C-y)>0\ \}$
3	right	x	$y + i_r$	$i_r = \{\ 1 \quad \text{to} \quad C-y \mid\ C-y>0\ \}$
4	down-right	$x + i_{dr}$	$y + i_{dr}$	$i_{dr} = \{\ 1 \quad \text{to} \quad \min(R-x, C-y) \mid\ \min(R-x, C-y)>0\ \}$
5	down	$x + i_d$	y	$i_d = \{\ 1 \quad \text{to} \quad R-x \mid\ R-x>0\ \}$
6	down-left	$x + i_{dl}$	$y - i_{dl}$	$i_{dl} = \{\ 1 \quad \text{to} \quad \min(R-x, y-1) \mid\ \min(R-x, y-1)>0\ \}$
7	left	x	$y - i_l$	$i_l = \{\ 1 \quad \text{to} \quad y-1 \mid\ y-1>0\ \}$
8	up-left	$x - i_{ul}$	$y - i_{ul}$	$i_{ul} = \{\ 1 \quad \text{to} \quad \min(x-1, y-1) \mid\ \min(x-1, y-1)>0\ \}$

We calculate disparity d_h as a disparity of the normalized histogram maximum:

$$h(i) = \frac{H(i)}{\sum_i H(i)},\ i = D_{min}\ to\ D_{max} \tag{8.13}$$

$$d_h = \arg\max_i h(i),\ i = D_{min}\ to\ D_{max} \tag{8.14}$$

The initial disparity $d(x, y)$ is replaced by the new value d_h if it is significantly supported i.e. if the normalized histogram value $h(d_h)$ is greater than α, otherwise it is left unchanged:

$$d(x,y) = \begin{cases} d_h & : \text{ if } |d_h - d(x,y)| > 1 \ \wedge \ h(d_h) > \alpha \\ d(x,y) & : \text{ otherwise} \end{cases} \tag{8.15}$$

where α, $0 \leq \alpha < 1$, is a significance threshold. The steps given by equations (8.12), (8.13), (8.14) and (8.15), are repeated iteratively until there are no more updates to disparities in the map.

Next, we detect *occluded disparities* by the consistency check between two disparity maps:

$$|D_{RL}(x, y - D_{LR}(x, y)) - D_{LR}(x, y)| \leq 1. \tag{8.16}$$

If the condition in (8.16) is not satisfied for disparity $D_{LR}(x, y)$, we declare it as inconsistent and eliminate it from the disparity map. The missing disparities are filled in by an iterative refinement procedure similar to the previously applied procedure for the disparity propagation by histogram voting. In the iterative step to fill in the inconsistent disparities, we use the threshold look-up table (8.11) as in the disparity refinement step. We calculate the histogram h of the consistent disparities with close intensities along 8 radial directions as given by (8.12) and (8.13). The missing disparity is filled in with the disparity d_h with the largest support in the histogram, provided that the histogram is not empty. The remaining unfilled inconsistent disparities, we fill in by the disparity of the nearest neighbor with known disparities with the smallest intensity differences. As a last step in the postprocessing, we apply $L \times L$ *median filter* to obtain the final disparity map.

8.3 Experiments and Discussion

We have used the Middlebury stereo benchmark [4] to evaluate the performance of our stereo matching algorithm. The parameters of the algorithm are fixed for all four stereo pairs as required by the benchmark. There are five free parameters in our algorithm. The threshold value is set to $T = 12$. The half-window size is $w = 15$, and the window size is $W \times W$ where $W = 31$. The noise variance σ_n^2 is a small and constant scaling factor in equation (8.8). The ratio coefficient in hybrid WTA is $K_p = 0.5$. In the post-processing step, the median filter parameter is $L = 5$ and the significance threshold in histogram voting is $\alpha = 0.45$.

Figure 8.5 shows results for all four stereo pairs from the Middlebury stereo evaluation database: *Tsukuba*, *Venus*, *Teddy* and *Cones*. The leftmost column contains the

left images of the four stereo pairs. The ground truth (GT) disparity maps are shown in the second column, the estimated disparity maps are shown in the third column and the error maps are shown in the forth column. In the error maps, the white regions denote correctly calculated disparity values which do not differ for more than 1 from the ground truth. If the estimated disparity differs for more than 1 from the ground truth value, it is marked as an error. The errors are shown in black and gray, where black represents the errors in the nonoccluded regions and gray represents errors in the occluded regions. The quantitative results in the Middlebury stereo evaluation framework are presented in Table 8.2.

The results show that our stereo algorithm preserves disparity edges. It estimates successfully the disparities of thin objects, and successfully deals with subtle radiometrical differences between images of the same stereo pair. Occlusion errors are not propagated and occluded disparities are successfully filled in the post-processing step. A narrow object is best visible in the *Tsukuba* disparity map (the lamp construction) and in *Cones* disparity map (pens in a cup in the lower right corner). Our algorithm correctly estimates disparities of both textureless and textured surfaces e.g. the example of large uniform surfaces in stereo pairs *Venus* and *Teddy* are successfully recovered.

The images in the Middlebury database have different sizes, different disparity ranges, and different radiometric properties. The stereo pairs *Tsukuba*, 384×288 pixels, and *Venus*, 434×383 pixels, have disparity ranges from 0 to 15 and from 0 to 19. The radiometric properties of the images in these stereo pairs are almost identical, and the offset compensation given by algorithm 9 is not significant for these two example pairs, as we demonstrated in [73]. As required by the Middlebury evaluation framework, we apply the offset compensation to all four stereo pairs. The stereo pairs *Teddy*, 450×375 pixels, and *Cones*, 450×375 pixels, have disparity ranges from 0 to 59. The images of these stereo pairs are not radiometrically identical and the offset compensation successfully deals with these radiometrical differences [73].

Table 8.2 Evaluation results based on the online Middlebury stereo benchmark [4]: The errors are given in percentages for the nonoccluded (NONOCC) region, the whole image (ALL) and discontinuity (DISC) areas. The numbers within brackets indicate the ranking in the Middlebury table.

Images	NONOCC	ALL	DISC
Tsukuba	**1.33 (37)**	1.82 (32)	7.19 (46)
Venus	**0.32 (39)**	0.79 (46)	4.5 (58)
Teddy	**5.32 (17)**	11.9 (40)	14.5 (19)
Cones	**2.73 (14)**	9.69 (53)	7.91 (21)

The error percentages together with ranking in the Middlebery evaluation online list are given in Table 8.2. The numbers show error percentages for non-occluded regions (NONOCC), discontinuity regions (DISC) and the whole (ALL) disparity map.

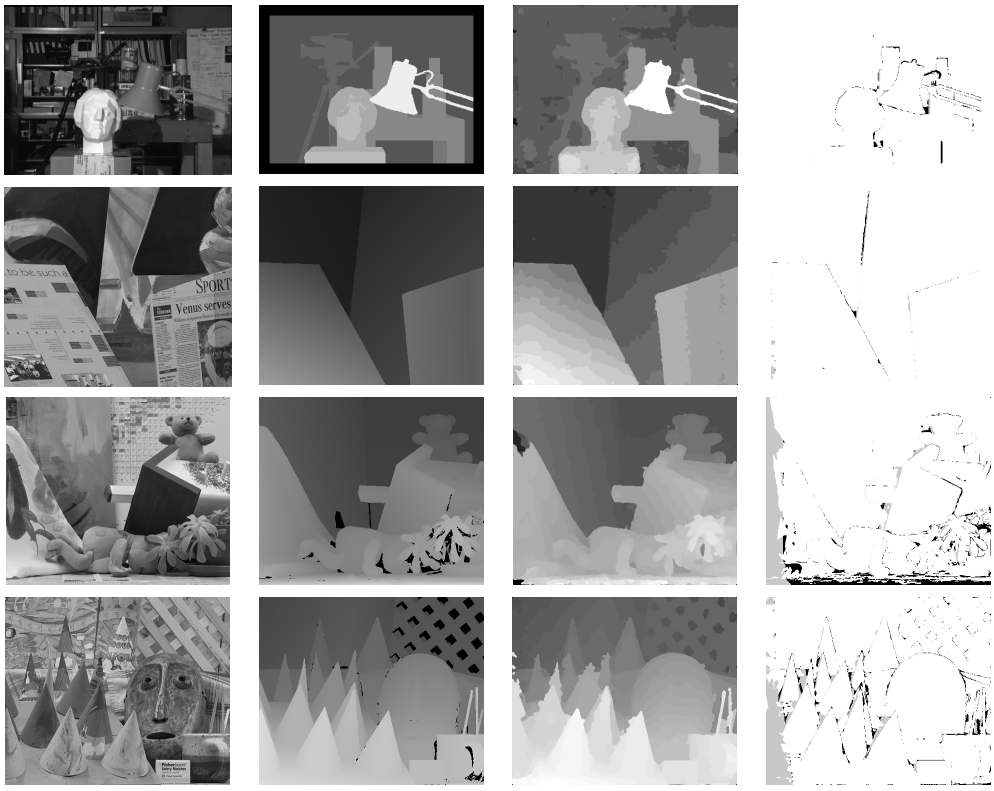


Figure 8.5 Disparity results for the stereo pairs (1st row: *Tsukuba*, 2nd row: *Venus*, 3rd row: *Teddy*, 4th row: *Cones*) from the Middlebury testbed database. The columns show, from left to the right : The left image, Ground truth, Result computed by our stereo algorithm, Disparity error map larger than 1 pixel. The nonoccluded regions errors with ranking are respectively: *Tsukuba* 1.33% (37), *Venus* 0.32% (39), *Teddy* 5.32% (17), *Cones* 2.73% (14)

The overall ranking of our algorithm in the Middlebery evaluation table of stereo algorithms is the 28^{th} place out of 123 evaluated algorithms. Thus, our stereo algorithm outperforms many local as well as global algorithms. Among the algorithms ranked in the Middlebury stereo evaluation, there are only two local algorithms ranked higher than our algorithm but both of them do not impose the fronto-parallel assumption strictly: a local matching method using image geodesic supported weights *GeoSup* [74], and a matching approach with slanted support windows *PatchMatch* [84]. Both of these algorithms use colored images, while our algorithm works with intensity images and achieves comparable results. Although these approaches have better general ranking in the Middlebury stereo evaluation list, our approach with matching based on fronto-parallel regions outperforms the *PatchMatch* algorithm for *Tsukuba* stereo pair, and the *GeoSup* algorithm for *Tsukuba*, *Teddy* and *Cones* stereo pairs. Thus, our approach with region selection by threshold produces more accurate disparity maps for cluttered scenes than *GeoSup* algorithm with region selection using geodesic support weights.

To investigate the contribution of the preprocessing and the postprocessing steps to the overall result, we show in table 8.3 the results we obtained on the benchmark stereo pairs with or without the preprocessing and the postprocessing steps in the algorithm. We show the results if neither, only one, and both steps are applied. If our postprocessing step was omitted, the $L \times L$ median filter was applied. From the results in table 8.3, we conclude that both steps, if individually applied, improve the qualities of the final disparity maps. If we apply both steps, the accuracy of the disparity maps is the highest. Furthermore, the improvement contribution of the preprocessing step is greater than the postprocessing step only for *Venus* stereo pair. This is because the sampling effects were most pronounced in *Venus* scene. In addition, we show in figure 8.6 the disparity maps for *Tsukuba* stereo pair for all four combinations: if the preprocessing and the postprocessing steps are included or not in the algorithm. We conclude that the preprocessing step plays a significant role in accurate disparity estimation of textureless areas, while the postprocessing step especially helps in an accurate estimation of disparity discontinuities.

To illustrate the subtle features of our algorithm not captured in the standard test bed images, we apply our stereo algorithm, while retaining the parameter values, on some other images from the Middlebury site in Figure 8.7. For two other stereo pairs, *Art* and *Dolls*, we show the left images of two stereo pairs in the leftmost column. The ground truth (GT) disparity maps are in the second column. The third column shows our estimation of the disparity maps. The fourth column shows the error maps with regard to the ground truth. The algorithm successfully recovers the disparities of very narrow structures as in *Art* disparity map. The disparity of the cluttered scene is successfully estimated, as in *Dolls* disparity map.

Next, we demonstrate that the presented local stereo algorithm works well on practical problems. Examples of disparity map estimation and 3D reconstruction of a face are shown for stereo pair *Sanja* in figure 8.8. The disparity map estimation of a plant in stereo pair *Tomato seedling* is shown in figure 8.9. The parameters of the algorithm are kept the same as in the previous examples. Thus, our algorithm successfully estimates the disparity of the smooth low textured objects and is suitable

		Tsukuba			*Venus*			*Teddy*			*Cones*		
preP	postP	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
-	-	3.6	5.41	10.04	2.76	4.38	13.18	8.11	17.42	19.73	4.77	15.04	12.33
+	-	2.74	4.50	10.11	0.62	1.63	7.95	7.52	16.82	19.41	3.98	14.37	11.27
-	+	2.45	3.05	7.31	1.53	2.11	5.75	6.11	12.49	15.20	3.20	9.30	9.14
+	+	1.33	1.82	7.19	0.32	0.79	4.5	5.32	11.90	14.50	2.73	9.69	7.91

Table 8.3 Comparison of results with (+) or without(-) preprocessing (preP) and postprocessing (postP) steps

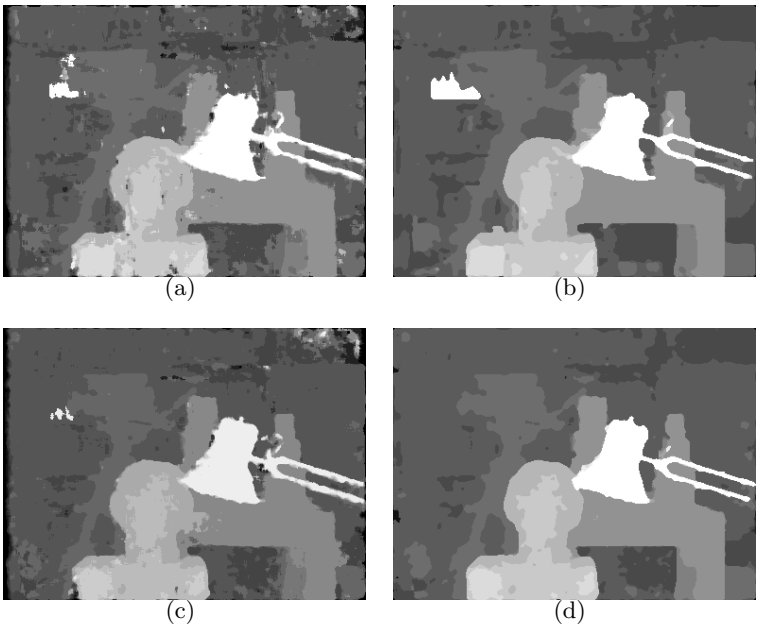


Figure 8.6 Disparity results for the stereo pair *Tsukuba*: (a) without preprocessing and without postprocessing, (b) without preprocessing and with postprocessing, (c) with preprocessing and without postprocessing, (d) with preprocessing and with postprocessing

also for application to 3D face reconstruction, figure 8.8(d). Our algorithm also successfully estimated the disparity map of the tomato seedling. *Tomato seedling* stereo images represent a challenging task for a stereo matching algorithm in general, because the viewpoints significantly differ and the structure of the plant is narrow i.e. much smaller than the window dimension.

As far as the initial window size is concerned, our algorithm is not influenced by the window size above certain size. In principle, we could apply our algorithm using

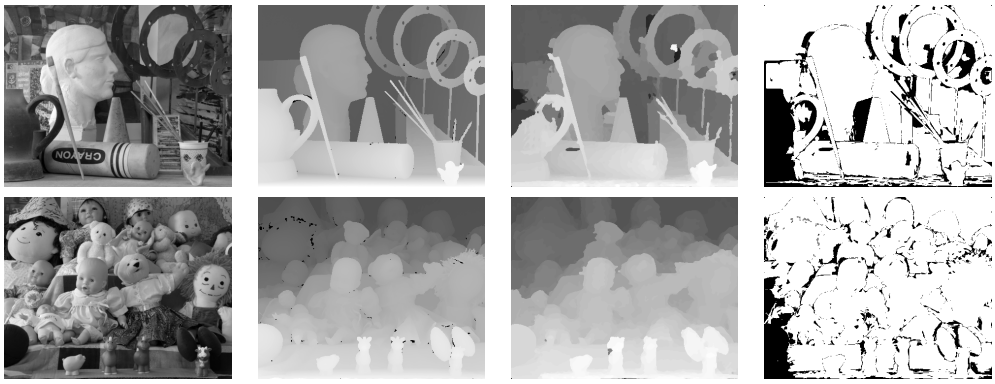


Figure 8.7 Disparity results for the stereo pairs (1st row: *Art*, 2nd row: *Dolls*) from the Middlebury database of the stereo images. Size of each image is 370 × 463 pixels. Disparity range in both stereo pair is 0 to 75. The columns show, from left to the right: The left image, The ground truth, The result computed by our stereo algorithm, The disparity error map larger than 1 pixel.

the whole image as the initial window around the reference pixel. This would result in a sufficiently large region selection for uniform regions in the image and make the the ordinal measure within the hybrid WTA more reliable. On the other hand, in matching windows with high local intensity variations, the selected region is always significantly smaller than the window and does not change if the window is enlarged because of the connectivity constraint with the reference central pixel.

8.4 Conclusion

In our local stereo algorithm, we have introduced a new approach for stereo correspondence based on the adaptive local segmentation by a dynamic threshold so that the fronto-parallel assumption holds for a segment. Further, we have established a relationship among the local intensity variation in an image and the dynamic threshold. We have applied the novel preprocessing procedure on both stereo images to eliminate the influence of noise and sampling artifacts. The mechanism for the final matching region selection prevents error propagation due to disparity discontinuities and occlusion. In the postprocessing step, we introduce a new histogram voting procedure for disparity refinement and for filling in the eliminated inconsistent disparities. Although, the starting point in matching is the large rectangular window, disparity of narrow structures is accurately estimated.

We evaluated our algorithm on the stereo pairs from the Middlebury database. It ranks highly on the list, outperforming many local and global algorithms that use

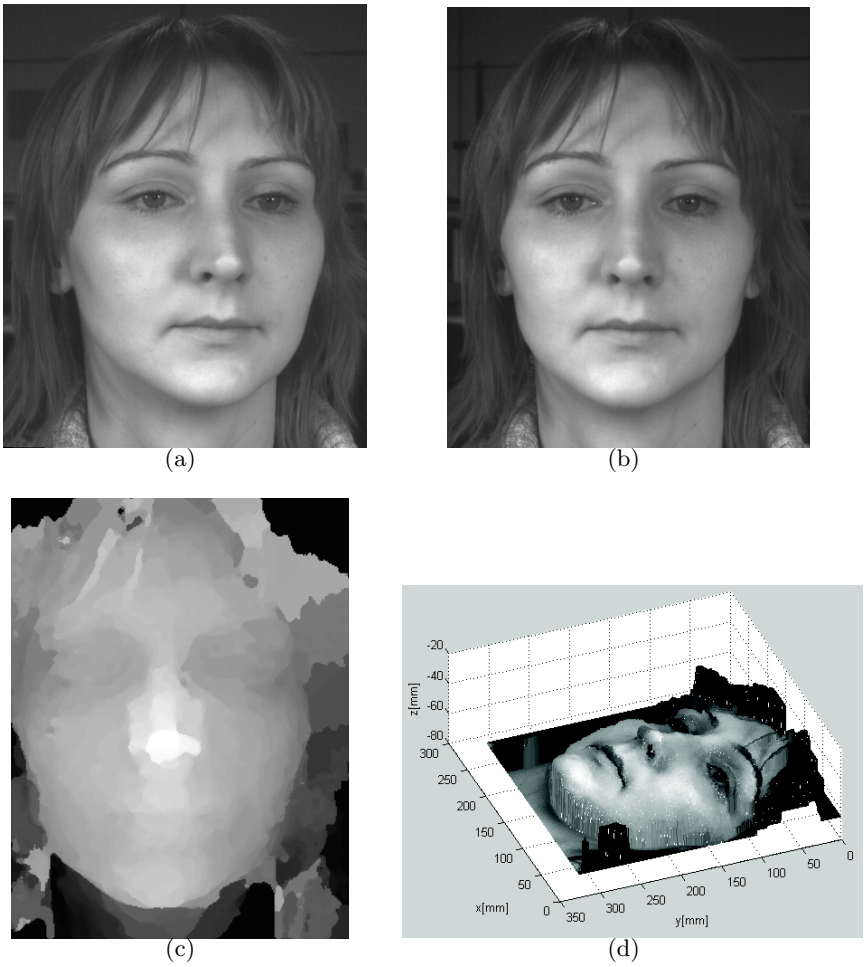


Figure 8.8 Disparity results for the stereo pair *Sanja*, taken at the vision laboratory of Signals and Systems Group, University of Twente. Size of each image is 781×641 pixels. Disparity range is 0 to 40. (a) Left stereo image (b) Right stereo image (c) Disparity map corresponding to the right image (d) Depth map with texture overlay

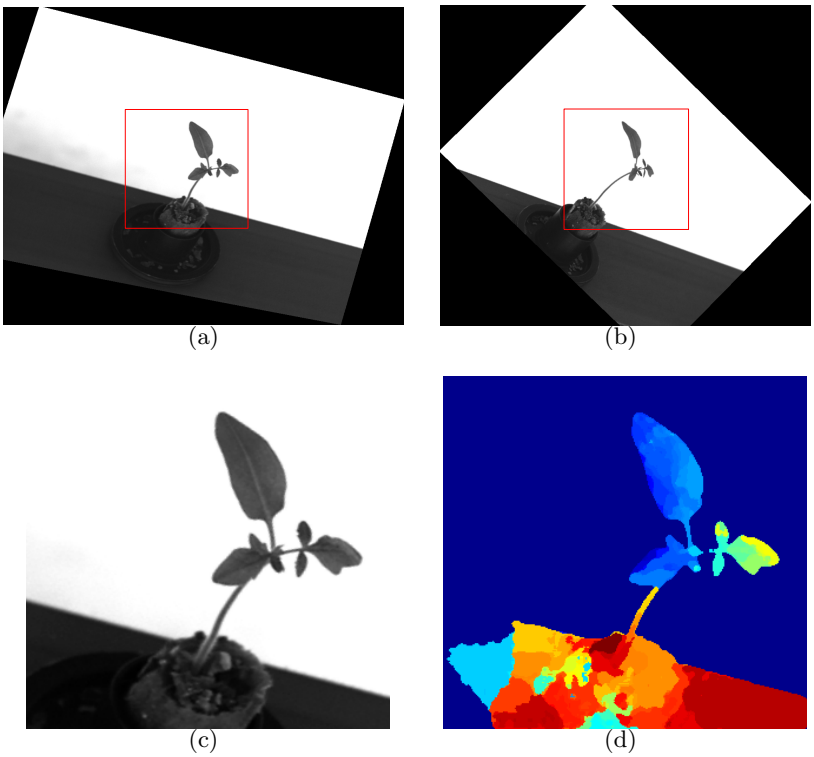


Figure 8.9 Disparity results for the stereo pair *Tomato seedling*, taken within MARVIN project at the vision laboratory of Intelligent System Group, Wageningen UR - Food and Biobased Research. Size of the region of interest in each image is 300×310 pixels. Disparity range is 0 to 90. (a) Left stereo image (b) Right stereo image (c) Region of interest in the left image (d) Disparity map corresponding to the left image

color information while we use only intensity images. Our algorithm is the best performing algorithm in the class of local algorithms which use intensity images and the fronto-parallel assumption without weighting the intensities of the matching region. Furthermore, our algorithm matches textureless as well as textured surfaces equally well, handles well the local radiometric differences, preserves edges in disparity maps, and successfully recovers the disparity of thin objects and the disparities of the occluded regions. We demonstrated the performance of our algorithm on two additional examples from the Middlebury database and on two practical examples. The results on this additional examples show that the disparity maps of scenes of different natures are successfully estimated: smooth low textured objects as well as textured cluttered scenes, narrow structures and textureless surfaces. Moreover, our algorithm has also other positive aspects making it suitable for real time implementation: it is local; it has just five parameters; intensity variations are locally calculated and there is no global segmentation algorithm involved.

9

Conclusion and Recommendations

The aim of our research was to investigate a stereo correspondence problem. We tackled the problem from the aspect of similarity measure. In this chapter and at the end of our pursuit of ideal similarity measure, we draw conclusions by answering the research questions posed in Chapter 1, Section 1.4, and give future directions for stereo matching research.

9.1 Conclusions

The purpose of this research was to investigate stereo matching. Stereo matching is a difficult problem because of the existence of occlusion, and because of the unknown gains, offsets and texture. In this research study, we focused on stereo matching using probabilistic global algorithms and later local algorithms, tackling a non-regularity of a problem from different aspects and answering the following questions:

How can we design a method for disparity estimation that is optimal in a probabilistic sense?

We defined the probabilistic framework for stereo matching using one-dimensional hidden Markov models. The state variable is disparity and the number of the states is equal to the disparity range of the scene. The state transition probabilities correspond to the scene characteristics and we chose it according to the expected disparity change along the epipolar line. As the observation probabilities are not known, they are expressed using the likelihood function. In Chapters 3 and 4, we investigated the optimal parameters and probabilistic algorithms for disparity estimation.

How can we define a disparity estimation as a one-dimensional state estimation problem?

We defined a stereo matching problem as a one-dimensional state estimation problem for the rectified pair of stereo images, by assuming that the scene statistics is described by the first order hidden Markov model. The stereo matching is treated as the state estimation problem where the state variable is the disparity. Evolution of the state variable then happens along the epipolar line, and the transition probabilities allow for continuous and abrupt transitions, i.e. changes of disparity along the epipolar line. The likelihood was derived using the normalized crosscorrelation.

Which probabilistic algorithms can be used to estimate disparity map from stereo images using one-dimensional hidden Markov model?

If the disparity map estimation is observed as a one-dimensional space-state estimation problem using one-dimensional hidden Markov models, a number of well known algorithms can be applied. The algorithms used for state estimation using hidden Markov models are: forward algorithm, forward-backward algorithm, Viterbi algorithm, dynamic programming, particle filter and particle filter followed by smoothing. We demonstrated the application of these algorithms for disparity estimation.

How can particle filter be applied to estimate disparity?

We defined the stereo matching as a state estimation problem and applied particle filter for stereo reconstruction. Within our probabilistic framework for stereo reconstruction, we applied a particle filter and particle filter followed by smoothing to disparity estimation. In Chapter 3, we demonstrated that a particle filter is a suitable for disparity estimation. The advantage of particle filter over other approaches

is its flexibility and ease to include more complex knowledge of the scene into the probabilistic model.

How do the different state estimation algorithms compare for different state space parameters?

In Chapter 4, we compared different probabilistic algorithms, for different parameters of the one-dimensional hidden Markov model for the fixed likelihood function. The algorithms compared showed expected behavior: online algorithms have similar behavior, and success rate in disparity estimation. There are several main factors that contribute to the success rate of the algorithm in an accurate disparity estimation, those being model parameters as well as the likelihood function. We investigated the influence of the state model parameters and concluded that it is necessary to have a more reliable likelihood function. As a possible improvement, we investigated a more suitable likelihood measure.

How can we define a likelihood measure that is optimal in a probabilistic sense?

Optimal likelihood function for stereo matching, in a probabilistic sense, is invariant to unknown texture, gains, and offsets. In Chapter 5, we derived a new likelihood function starting from the acquisition model of the point in an image.

How can we derive a likelihood measure which is invariant to unknown texture, gains, and offsets?

We introduced a new likelihood function for window-based stereo matching, based on a probabilistic model that can cope with unknown textures, uncertain gain factors, uncertain offsets, and correlated noise. We derived the likelihood function by modeling the uncertainties of the unknown texture mapping of the surface to the two image planes including the unknown gain, offset and noise, and then marginalizing the expression for the whole range of values. This resulted in the formula for likelihood which takes into account degrees of uncertainties and contributes to the more accurate matching results in comparison to using likelihood based on the sum of squared differences. We also showed that the sum of squared differences is an asymptotical case of our new likelihood formula.

How can we define an optimal region for matching?

An optimal matching region is comprised of pixels which belong to the projection of the smooth scene surface to an image and does not contain any disparity discontinuities. Without knowledge about scene geometry previous to matching, it is a very challenging task. In Chapters 6, 7 and 8, we approached this problem in a novel way by establishing a relationship between local texture and disparity.

How can we suitably select a sparse subset of pixels for matching from the initial matching windows with the aim of diminishing the influence of the occlusion and the depth discontinuity to the matching and how can we calculate a matching cost?

The suitable likelihood measure which takes into account uncertainties of texture, gain, offsets and noise can not deal in a straightforward way with occlusion. There were limitations of the window-based matching to deal with. For these reasons, we diverged from using the whole squared windows for similarity/cost calculation and looked into the mechanism of proper pixel selection for matching within the local stereo matching framework. In Chapter 6 we investigated sparse window matching. Thus we used only a subset of window pixels selected by a threshold with respect to the central pixel and showed that the results improve in comparison with using the whole windows for likelihood calculation. As the number of selected pixels for matching plays a role, it was necessary to normalize a matching cost to the number of participating pixels in cost calculation: so we introduced a hybrid *winner-take-all* (WTA) algorithm. Also, the most suitable threshold for pixel selection depends on the scene characteristics, as shown in Chapter 7. We showed that by choosing a suitable threshold the accuracy of the estimated disparity maps improves.

How can we establish a relationship between the fronto-parallel assumption and the local intensity variation for application in stereo matching? How do we select a segment for matching so that the fronto-parallel assumption holds for the segment?

The main assumption in window-based stereo matching is the fronto-parallel assumption, which states that disparities within the matching window vary slightly. Most errors in window-based stereo matching occur when the window contains dept discontinuity. The presence of the depth discontinuity within the window makes the pivoting fronto-parallel assumption of stereo matching invalid. It would be ideal if we knew beforehand if there is a discontinuity within the window, but it is not possible to learn without performing matching. We came up with the idea observing the problem differently, namely, from the aspect of local texture using local intensity variation. We established a relationship between the fronto-parallel assumption and the local intensity variation. We defined local intensity variation as the maximum difference between the pixel and its four neighbors. In addition, we performed adaptive local segmentation by thresholding, where the threshold is directly proportional to the local intensity variation. This may seem counter intuitive, but a small threshold will select the whole uniform region, while in a highly textured region the highest threshold value is necessary. Thus, the region for which the fronto-parallel assumption holds can be selected as a central window region using a suitably selected threshold for adaptive local segmentation.

What kind of intensity transformation on the image pixels makes the im-

age more favourable for local adaptive segmentation?
In adaptive local segmentation we performed segmentation by several fixed thresholds. In thresholding, the influence of image imperfections of the acquisition process such as noise and sampling effects can significantly influence the outcome of the segmentation. We invented a new intensity transformation which has favorable effects for adaptive local segmentation. A new intensity transformation has a smoothing effect and suppresses noise in low-textured image areas, while in high-textured areas it has a sharpening effect and suppresses sampling effects.

Which postprocessing steps deal successfully with inconsistently estimated disparities?
Postprocessing step detects and fills in the inconsistent disparities. We introduced a postprocessing step that fills in the inconsistent disparities using the local intensity variation and histogram voting. We observed the surrounding pixels along eight radial directions in the whole image around the inconsistent disparity and filled its disparity in as a disparity with a maximum histogram disparity. We took into account only pixels close in intensity. The idea behind this approach was also that it is expected that pixels close in intensity also have a close disparities.

9.2 Recommendations and Future Directions

In Chapter 7, we showed the importance of proper parameter selection for different stereo pairs. The optimal algorithm parameters depend on the scene characteristics in the stereo images. However, in Chapter 8 we kept the parameters fixed as required by the Middlebury evaluation framework for all four stereo pairs from the benchmark. It would be relevant to find out what the optimal parameters of the stereo matching algorithms for different stereo pairs would be and how much the disparity map estimation accuracy could be improved.

In Chapter 8, the starting point in adaptive local segmentation was a squared window. Generally, there is no need to use a rectangular window instead the whole image can be used as a starting point for segmentation. This would imply that the starting window size is unlimited or limited only by image size. Thus, a whole image can be used as a starting window, and a segment for matching could be selected by adaptive local segmentation.

In addition, we can investigate the effect of our interpretation of a relationship between a fronto-parallel assumption and the local intensity variation for large slanted surfaces. The question arises whether it is necessary to limit the size of the initial window for slanted surfaces. For instance, we could consider whether a textureless surface is slanted and then limit a window width in that case.

The intensity transformation that we used in a preprocessing step for adaptive local segmentation based matching could be analysed and quantified further. An interesting question is what the influence of the median and mean relationship for the transformation is.

For future consideration, it would be interesting to analyse the speed of the algorithm when implemented for real time implementation, for examplr in programming language C.

The next question is whether we could extend the stereo matching to matching three views, and which contribution would be expected? How should occlusion treated in such a multiview case?

References

[1] R. Szeliski, *Computer Vision: Algorithms and Applications (Texts in Computer Science)*. Springer, 1st edition. ed., Nov. 2010.

[2] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993.

[3] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 993–1008, 2003.

[4] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[5] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second ed., 2004.

[6] "Middlebury stereo website, http://vision.middlebury.edu/stereo/," March 2012.

[7] D. Marr and T. Poggio, "A computational theory of human stereo vision," *Proceedings of the Royal Society of London*, vol. B-204, pp. 301–328, 1979.

[8] A. S. Ogale and Y. Aloimonos, "Stereo correspondence with slanted surfaces: critical implications of horizontal slant," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, pp. I–568–I–573 Vol.1, June-2 July 2004.

[9] J. D. Oh, S. Ma, and C.-C. Kuo, "Stereo matching via disparity estimation and surface modeling," *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pp. 1–8, June 2007.

[10] J. Sun, Y. Li, S. Kang, and H.-Y. Shum, "Symmetric stereo matching for occlusion handling," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 399–406 vol. 2, June 2005.

[11] U. R. Dhond and J. Aggarwal, "Analysis of the stereo correspondence process in scenes with narrow occluding objects," in *Pattern Recognition, 1992. Vol.I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on*, pp. 470–473, Aug-3 Sep 1992.

[12] P. Moallem and K. Faez, "Effective parameters in search space reduction used in a fast edge-based stereo matching," *Journal of Circuits, Systems, and Computers (JCSC)*, vol. 14, pp. 249 – 266, 2005.

[13] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, "A comparative study of energy minimization methods for markov random fields with smoothness-based priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 1068–1080, 2008.

[14] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[15] P. F. Felzenszwalb and D. R. Huttenlocher, "Efficient belief propagation for early vision," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, pp. I–261–I–268 Vol.1, June-2 July 2004.

[16] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2, pp. 508–515 vol.2, 2001.

[17] M. Bleyer and M. Gelautz, "Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions," *Image Commun.*, vol. 22, no. 2, pp. 127–143, 2007.

[18] Y. Boykov, O. Veksler, and R. Zabih, "Markov random fields with efficient approximations," in *In IEEE Conference on Computer Vision and Pattern Recognition (CVPR98)*, pp. 648–655, 1998.

[19] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs, "A maximum likelihood stereo algorithm," *Comput. Vis. Image Underst.*, vol. 63, no. 3, pp. 542–567, 1996.

[20] A. F. Bobick and S. S. Intille, "Large occlusion stereo," *Int. J. Comput. Vision*, vol. 33, no. 3, pp. 181–200, 1999.

[21] P. N. Belhumeur, "A Bayesian approach to binocular stereopsis," *Int. J. Comput. Vision*, vol. 19, no. 3, pp. 237–260, 1996.

[22] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo matching using belief propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 787–800, 2003.

[23] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Constructing free energy approximations and generalized belief propagation algorithms," *IEEE Transactions on Information Theory*, vol. 51, pp. 2282–2312, 2005.

[24] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," in *Exploring artificial intelligence in the new millennium*, pp. 239–269, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003.

[25] M. C. Sung, S. H. Lee, and N. I. Cho, "Stereo matching using multi-directional dynamic programming and edge orientations," in *Intelligent Signal Processing and Communications, 2006. ISPACS '06. International Symposium on*, pp. 697–700, Dec. 2006.

[26] C. Lei, J. Selzer, and Y.-H. Yang, "Region-tree based stereo using dynamic programming optimization," in *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (Washington, DC, USA), pp. 2378–2385, IEEE Computer Society, 2006.

[27] O. Veksler, "Stereo correspondence by dynamic programming on a tree," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2, pp. 384–390 vol. 2, June 2005.

[28] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, 2008.

[29] H. Hirschmuller, "Stereo vision in structured environments by consistent semi-global matching," in *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (Washington, DC, USA), pp. 2386–2393, IEEE Computer Society, 2006.

[30] Z.-F. Wang and Z.-G. Zheng, "A region based stereo matching algorithm using cooperative optimization," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, vol. 0, pp. 1–8, 2008.

[31] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3, pp. 15–18, 0-0 2006.

[32] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," *Int. J. Comput. Vision*, vol. 35, no. 3, pp. 269–293, 1999.

[33] H. Hirschmueller and D. Scharstein, "Evaluation of stereo matching costs on images with radiometric differences," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 9, pp. 1582–1599, 2009.

[34] H. Hirschmüller, P. R. Innocent, and J. Garibaldi, "Real-time correlation-based stereo vision with reduced border errors," *Int. J. Comput. Vision*, vol. 47, no. 1-3, pp. 229–246, 2002.

[35] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, (Washington, DC, USA), p. 839, IEEE Computer Society, 1998.

[36] Y. S. Heo, K. M. Lee, and S. U. Lee, "Illumination and camera invariant stereo matching," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, June 2008.

[37] J. Kim, V. Kolmogorov, and R. Zabih, "Visual correspondence using energy minimization and mutual information," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pp. 1033–1040 vol.2, Oct. 2003.

[38] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, (Washington, DC, USA), pp. 807–814, IEEE Computer Society, 2005.

[39] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *ECCV '94: Proceedings of the third European Conference on Computer Vision (Vol. II)*, (Secaucus, NJ, USA), pp. 151–158, Springer-Verlag New York, Inc., 1994.

[40] D. Bhat, S. Nayar, and A. Gupta, "Motion estimation using ordinal measures," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 982–987, Jun 1997.

[41] F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda, "Near real-time stereo based on effective cost aggregation," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*, pp. 1–4, Dec. 2008.

[42] J. Lu, G. Lafruit, and F. Catthoor, "Fast variable center-biased windowing for high-speed stereo on programmable graphics hardware," in *ICIP (6)*, pp. 568–571, 2007.

[43] S. Rogmans, J. Lu, P. Bekaert, and G. Lafruit, "Real-time stereo-based view synthesis algorithms: A unified framework and evaluation on commodity gpus," *Image Commun.*, vol. 24, no. 1-2, pp. 49–64, 2009.

[44] C. L. Zitnick and T. Kanade, "A cooperative algorithm for stereo matching and occlusion detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, pp. 675–684, 2000.

[45] A. Fusiello, V. Roberto, and E. Trucco, "Efficient stereo with multiple windowing," in *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, (Washington, DC, USA), p. 858, IEEE Computer Society, 1997.

[46] M. Okutomi, Y. Katayama, and S. Oka, "A simple stereo algorithm to recover precise object boundaries and smooth surfaces," *International Journal of Computer Vision*, vol. 47, pp. 261–273, 2002.

[47] S. B. Kang, R. Szeliski, and J. Chai, "Handling occlusions in dense multi-view stereo," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I–103–I–110 vol.1, 2001.

[48] H. Tao and H. Sawhney, "Global matching criterion and color segmentation based stereo," in *Applications of Computer Vision, 2000, Fifth IEEE Workshop on.*, pp. 246–253, 2000.

[49] D. Comaniciu and P. Meer, "Mean shift: a robust approach toward feature space analysis," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, pp. 603–619, May 2002.

[50] Y. Taguchi, B. Wilburn, and C. Zitnick, "Stereo reconstruction with mixed pixels using adaptive over-segmentation," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, June 2008.

[51] M. Bleyer, M. Gelautz, C. Rother, and C. Rhemann, "A stereo approach that handles the matting problem via image warp," in *CVPR09*, 2009.

[52] G. Egnal and R. Wildes, "Detecting binocular half-occlusions: empirical comparisons of five approaches," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, pp. 1127–1133, Aug 2002.

[53] T. Kanade and M. Okutomi, "A stereo matching algorithm with an adaptive window: Theory and experiment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 16, no. 9, pp. 920–932, 1994.

[54] K.-J. Yoon and I.-S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 650–656, 2006.

[55] F. van der Heijden, R. Duin, D. de Ridder, and D. Tax, *Classification, Parameter Estimation and State Estimation An Engineering Approach Using MATLAB*. WILEY, 2004.

[56] L. Cheng and T. Caelli, "Bayesian stereo matching," *Comput. Vis. Image Underst.*, vol. 106, pp. 85–96, Apr. 2007.

[57] W. Kim, J. Park, and K. M. Lee, "Stereo matching using population-based MCMC," *Int. J. Comput. Vision*, vol. 83, no. 2, pp. 195–209, 2009.

[58] T. Son and S. Mita, "Stereo matching algorithm using a simplified trellis diagram iteratively and bi-directionally," *IEICE - Trans. Inf. Syst.*, vol. E89-D, no. 1, pp. 314–325, 2006.

[59] F. H. van der, F. Berendsen, L. Spreeuwers, and E. Schippers, "Particle smoothing for solving ambiquity problems in one-shot structured light systems," in *Proceedings of the International Conference on Computer Vision Theory and Applications*, pp. 531–537, SciTEPress, 2011.

[60] A. D. W. Fong, S.J. Godsill and M. West, "Monte Carlo smoothing with application to audio signal enhancement," *IEEE Transaction on Signal Processing*, vol. 50, pp. 438–449, February 2002.

[61] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. of the IEEE*, vol. 77, pp. 257 – 286, 1989.

[62] R. M. Gray and L. D. Davisson, *Introduction to Statistical Signal Processing*. Cambridge University Press, 2010.

[63] G. Egnal, "Mutual information as a stereo correspondence measure," Tech. Rep. Technical Report MS-CIS-00-20, Comp. and Inf. Science, U. of Pennsylvania, 2000.

[64] S. Roy and I. J. Cox, "A maximum-flow formulation of the n-camera stereo correspondence problem," in *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, (Washington, DC, USA), p. 492, IEEE Computer Society, 1998.

[65] L. J. Spreeuwers, "Multi-view passive acquisition device for 3d face recognition," in *BIOSIG 2008: Biometrik und elektronische Signaturen*, 2008.

[66] F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda, "Classification and evaluation of cost aggregation methods for stereo correspondence," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, June 2008.

[67] L. Nalpantidis, G. C. Sirakoulis, and A. Gasteratos, "Review of Stereo Vision Algorithms: from Software to Hardware," *International Journal of Optomechatronics*, vol. 2, no. 4, pp. 435–462, 2008.

[68] K. Zhang, J. Lu, and G. Lafruit, "Scalable stereo matching with locally adaptive polygon approximation," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pp. 313–316, Oct. 2008.

[69] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann, "Geodesic adaptive support weight approach for local stereo matching," in *Computer Vision Winter Workshop 2010*, pp. 60–65, 2010.

[70] S. Mattoccia, "A locally global approach to stereo correspondence," in *3DIM09*, pp. 1763–1770, 2009.

[71] I. J. Cox, "A maximum likelihood n-camera stereo algorithm," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 733–739, 1994.

[72] S. Damjanović, F. van der Heijden, and L. J. Spreeuwers, "A new likelihood function for stereo matching: how to achieve invariance to unknown texture, gains and offsets?," in *VISIGRAPP 2009, International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, Lisboa, Portugal*, (Lisboa), pp. 603–608, INSTICC Press, February 2009.

[73] S. Damjanović, F. van der Heijden, and L. J. Spreeuwers, "Sparse window local stereo matching," in *VISIGRAPP 2011*, pp. 689–693, 2011.

[74] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann, "Local stereo matching using geodesic support weights," in *Proceedings of the 16th IEEE international conference on Image processing*, ICIP09, (Piscataway, NJ, USA), pp. 2069–2072, IEEE Press, 2009.

[75] K. Zhang, J. Lu, G. Lafruit, R. Lauwereins, and L. Van Gool, "Accurate and efficient stereo matching with robust piecewise voting," in *Proceedings of the 2009 IEEE international conference on Multimedia and Expo*, ICME'09, (Piscataway, NJ, USA), pp. 93–96, IEEE Press, 2009.

[76] R. K. Gupta and S.-Y. Cho, "Real-time stereo matching using adaptive binary-window," in *3DPVT 2010*, 2010.

[77] X. Sun, X. Mei, S. Jiao, M. Zhou, and H. Wang, "Stereo matching with reliable disparity propagation," in *IEEE Int. Conf. on 3D Digital Imaging, Modeling, Processing, Visualisation and Transmittion (3DIMPVT)*, 2011.

[78] C. R. M. Bleyer and P. Kohli, "Surface stereo with soft segmentation," in *CVPR*, 2010.

[79] H. Tao, H. S. Sawhney, and R. Kumar, "A global matching framework for stereo computation," in *ICCV*, pp. 532–539, 2001.

[80] R. Szeliski and D. Scharstein, "Sampling the disparity space image," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, pp. 419–425, March 2004.

[81] Q. Wu, F. Merchant, and K. R. Castleman, *Microscope Image Processing*. Academic Press, 2008.

[82] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB, 2nd ed.* Gatesmark Publishing, 2nd ed., 2009.

[83] S. Damjanović, F. van der Heijden, and L. J. Spreeuwers, "Sparse window stereo matching," in *Proceedings of the International Workshop on Computer Vision Applications (CVA)*, pp. 83–86, 2011.

[84] M. Bleyer, C. Rhemann, and C. Rother, "Patchmatch stereo - stereo matching with slanted support windows," in *British Machine Vision Conference 2011*, 2011.

Summary

The aim of stereo matching is to find a corresponding point for each pixel in a reference image of a stereo image pair in the other image. Corresponding points are projections onto the stereo images of the same scene point. Finding corresponding points is an essential problem in dense stereo matching. The relative displacement between the corresponding points in rectified stereo images is termed disparity. Stereo matching is ambiguous because of photometric issues, surface structure and geometric ambiguities. For highly reflective or specular surfaces, the appearance in different images may differ significantly. Also, finding corresponding points within uniformly colored regions or surfaces with repeating texture or structure is a huge problem. Some points do not have corresponding points due to occlusion or due to the limited field of view.

After presenting a literature overview on stereo correspondence, we began our journey into stereo matching by defining a probabilistic framework. We defined disparity estimation as a state estimation problem using a one-dimensional hidden Markov model with a number of states equal to the number of disparities in the disparity range. We showed that the particle filter and the particle filter followed by smoothing can be used in disparity estimation. We introduced and qualitatively compared five probabilistic algorithms for disparity estimation: the forward algorithm, the forward/backward algorithm, the Viterbi algorithm, the particle filter and the particle filter in combination with smoothing.

We derived a new likelihood function for correspondence that is optimal in a probabilistic sense for stereo matching and invariant for unknown texture, gains and offsets.

We deviated from the squared window based likelihood in order to include only relevant pixels in the likelihood function. We introduced local stereo matching using sparse windows. This approach gave us a significant improvement compared to matching based on the complete windows. Optimal parameters for sparse matching depend on the nature of the scene. Whether the scene is characterised by highly textured or low textured surfaces influences the parameter choice for accurate disparity estimation. We considered the most suitable parameters for sparse stereo matching.

Further led by the idea that a different nature of texture requires a different approach to likelihood estimation, we redefined several of the most common assumptions and established a relationship between the texture and the fronto-parallel assumption and introduced local adaptive segmentation based on the local intensity variation. We redefined the Lambertian assumption for offset compensation and introduced novel preprocessing and postprocessing steps for accurate disparity map estimation.

Samenvatting

Het doel van stereomatching is een corresponderend punt te vinden voor elk pixel in het referentie beeld van een stereo paar in het andere beeld. Corresponderende punten zijn projecties van hetzelfde punt in een scène op de stereobeelden. Het vinden van corresponderende punten is een belangrijk probleem in stereomatching. De relatieve verschuiving tussen de corresponderende punten in gerectificeerde stereobeelden wordt dispariteit benoemd. Stereomatching is geen eenduidig oplosbaar probleem door verschillen in fotometrische eigenschappen van stereocamera's, de aanwezigheid van oppervlaktestructuren van afgebeelde objecten en geometrische verschillen in stereobeelden. Sterk reflecterende of spiegelende oppervlakken worden verschillend afgebeeld in stereobeelden. Ook, het vinden van corresponderende punten op egale delen in de beelden of oppervlakken met een herhalende structuur maken de stereocorrespondentie een lastig probleem. Sommige punten hebben zelfs geen corresponderende punten door occlusie of door een beperkte kijkhoek.

Na de presentatie van een literatuuroverzicht van stereo correspondentie, begonnen we onze expeditie naar stereomatching met de definitie van een probabilistisch raamwerk. We definieerden de schatting van de dispariteit als een toestandsschattingsprobleem met behulp van een 1-dimensionaal hidden Markov model waarvan het aantal toestanden gelijk is aan het aantal verschillende dispariteiten in het dispariteitsbereik. We lieten zien dat het partikelfilter en het partikelfilter met smoother voor schatting van dispariteit kunnen worden gebruikt. We introduceerden, kwantificeerden en vergeleken vijf probabilistische algoritmen voor dispariteitsschatting: het forward-algoritme, het forward/backward-algoritme, het Viterbi-algoritme, het partikelfilter en partikelfilter in combinatie met smoothing.

We introduceerden een nieuwe likelihoodfunctie voor correspondentie die optimaal is in probabilistische zin voor stereo matching en ook invariant voor onbekende textuur, gains en offsets.

We weken af van de standaard op vierkante windows gebaseerde likelihood, door alleen de belangrijke pixels in de likelihoodfunctie mee te tellen d.w.z. we introduceerden lokale stereomatching met sparse windows. Deze aanpak gaf een substantiële verbetering in vergelijking met de matching die op gehele windows gebaseerd is. De optimale parameters voor sparse matching zijn sterk afhankelijk van de soort van de scène. Met name bij de aanwezigheid van sterke texturen of juist de afwezigheid van textuur, heeft de keuze van de parameters een grote invloed op de nauwkeurigheid van de schatting van de dispariteit. We onderzochten de best geschikte parameters voor sparse stereomatching.

Voortbordurend op het idee, dat verschillende soorten textuur een verschillende

aanpak voor de bepaling van de likelihood nodig hebben, hebben we een aantal van de meestvoorkomende aannamen geherdefinieerd en we stelden een relatie tussen de textuur en de fronto-parallel aanname vast. Ook introduceerden we een methode voor lokale adaptieve segmentatie, gebaseerd op de lokale variatie van de intensiteit. We hebben de Lambertiaanse aanname geherdefineerd voor offset compensatie en introduceerden nieuwe methoden voor preprocessing en postprocessing voor accurate schatting van de dispariteit.

Acknowledgements

Six years ago, I came to The Netherlands to start my Ph.D. research at the chair of Signal and Systems of the University of Twente. At the end of my Ph.D. journey, I would like to thank people that made my journey possible and unique.

I would like to express my sincere gratitude to my promotor Kees Slump for giving me the opportunity to work in his group on a research topic I have always considered fascinating.

I am grateful to my assistant promotors Luuk Spreeuwers and Ferdi van der Heijden for all the discussions, for their feedback, for constructive criticism and for their work to improve my writings.

I would like to thank all other members of the graduation committee for their interest in my Ph.D. research and willingness to be in the graduation committee.

I thank my colleagues from the Signals and Systems group for the atmosphere, for gezellige Friday borrels, and for motivation and help in learning Dutch.

I am grateful to the group secretaries Anneke and Sandra for their kindness and help with administrative stuff.

I wish to thank the board members of the Female Faculty Network (FFNT) of the University of Twente for all the nice times we had while I was a member of the board.

There is a number of friends to whom I want to thank a lot for support and a good time spent in Enschede: Jasminka and Alek, Jelena, Marijana, Miladin, Olja and Mina, Dragan and Tanja, Banu and Oktay, Olena and Remco, Anja and Bojan.

After five years living in Enschede, a year ago I moved to Deventer and started to work at Wageningen University and Research Centre. I wish to thank all colleagues from my group for their support during finalizing this PhD thesis.

I am grateful to my parents and my sister Jelena for their support and love and for always believing in my success.

I am most grateful to my life partner Milan. Milan, thank you for all your love, support and cheering me up in the difficult moments.

Sanja Damjanović
October 7^{th}, 2012
Deventer, The Netherlands

Biography

Sanja Damjanović received the Dipl.-Ing. in electrical engineering degree with specialization in telecommunications and Magister of Science in electrical engineering degree with specialization in digital signal processing from the University of Belgrade in 2001 and 2005, respectively. From 2001 till 2005, she was a research and development engineer in the Department of Telecommunications, Mihajlo Pupin Institute, Belgrade, where she worked on the topics of wavelet multirate filter banks and software development for network equipment. From October 2005 till August 2006, she was a DAAD research fellow at Digital signal processing group (DISPO), Ruhr-Universität Bochum, where she worked on the research topic of IIR filters banks. Since September 2006 till August 2011, she was a Ph.D. candidate at Signals and Systems Group, University of Twente, working on the research topic of stereo matching. Since September 2011, she works as a research scientist at Consumer Science and Intelligent Systems, Wageningen University and Research Centre, in the field of computer vision applied to agrotechnology and consumer science. Her research interests include computer vision, signal and image processing. She published a number of papers in journals and international conference proceedings in the fields of signal processing and computer vision.