STEREO MATCHING FOR SCENES OF NATURAL ORIGIN

MICHAL KRUMNIKL

Field of study: Computer Science and Applied Mathematics
Supervisor: doc. Dr. Ing. Eduard Sojka

Ostrava, 2013
Dedicated to Weronika
Strive for perfection in everything you do. Take the best that exists and make it better. When it does not exist, design it.

—Sir Henry Royce

Vision without execution is hallucination.

—Thomas Alva Edison
Computational stereo generally refers to a problem of determining the three-dimensional structure of a scene from two or more images taken from different positions. Stereo vision is currently an active research domain in computer vision. Moreover, it is considered to be one of the core topics in image analysis.

An interest in the stereo reconstruction has gradually grown for the past 30 years. The modern research in the stereo vision began in the mid-1970s and was largely supported by the DARPA. The first applications were mainly military and included cruising missiles and autonomous vehicles navigations. Nowadays, the results of three-dimensional reconstruction algorithms are used in many applications including robotics, industrial automation, autonomous land rovers, aircraft navigation, remote sensing, automated cartography, and also in stereomicroscopy.

This thesis presents the further developments of computer vision and stereo reconstruction in the field of biology. We describe novel algorithms for the stereo matching, depth segmentation and three-dimensional reconstruction and focus (not only) on the applications in the area of natural science, biology and environmental protection.

In the sequel, we will introduce the stereo matching problem, describe the state-of-the-art approaches and then present our own contributions. First, we will describe the construction of a unique scanning device, specially adapted for biological studies. Next, we will cover the necessary steps leading to the three dimensional reconstruction of the scanned samples. And finally, we will present three novel algorithms, invented during the work on the device.
ACKNOWLEDGMENTS

I would like to thank my supervisor, doc. Dr. Ing. Eduard Sojka, for the patient guidance, encouragement and advice he has provided throughout my time as his student. My special thanks go to my family and friends for their support and understanding.
CONTENTS

1 INTRODUCTION 1
   1.1 Stereo Correspondence Problem ........................................ 2
   1.2 Thesis Goals ............................................................... 3
   1.3 Main Contributions ....................................................... 3
   1.4 Thesis Outline ............................................................. 4

2 A BRIEF HISTORY OF STEREOPSIS 6
   2.1 Ancient Times .............................................................. 6
   2.2 Modern Era ................................................................. 7

3 STEREO VISION 9
   3.1 Stereo Geometry ............................................................ 9
      3.1.1 Epipolar Geometry .................................................... 10
      3.1.2 Rectification ........................................................... 11
   3.2 Correspondence and Reconstruction Problem ............................. 12
   3.3 Matching Constraints ...................................................... 13
      3.3.1 Ordering constraint ................................................... 13
      3.3.2 Uniqueness Constraint ............................................... 14
      3.3.3 Geometric Constraints .............................................. 14
      3.3.4 Continuity Constraint ............................................... 14

4 STATE-OF-THE-ART STEREo MATCHING ALGORITHMS 15
   4.1 Basic Classification ...................................................... 15
   4.2 Window Based Methods .................................................... 16
      4.2.1 Multiple Windowing ................................................... 17
      4.2.2 Adaptive Windows .................................................... 18
      4.2.3 Multiple Baselines ................................................... 19
      4.2.4 Modern Approaches .................................................. 19
      4.2.5 Parallel Concepts ..................................................... 20
   4.3 Dynamic Programming ...................................................... 20
      4.3.1 Simple Connectivity Constraint Approaches ....................... 21
      4.3.2 Intra- and Inter- Scanline Search ................................ 22
      4.3.3 Dynamic Programming on a Tree .................................... 23
   4.4 Graph Cut Algorithms ...................................................... 23
      4.4.1 Markov Random Fields ............................................... 24
      4.4.2 Minimum Cut .......................................................... 25
CONTENTS

4.4.3 Swap and Expansion Move Algorithms ........................................ 26
4.4.4 Parallel Concepts ............................................................. 27
4.5 Belief Propagation ............................................................... 27
4.5.1 Loopy Belief Propagation .................................................... 28
4.6 Semi-Global Matching ........................................................... 29
4.7 Segmentation Based Methods .................................................... 30
4.7.1 Segmented Based Algorithm with Self Adapting Dissimilarity ....... 30
4.7.2 Cooperative Optimization .................................................... 32

5 METHODOLOGY OF ALGORITHM EVALUATION .......................... 33
5.1 Comparison Criteria ............................................................. 33
5.2 Datasets ............................................................................. 34
5.3 Conclusions ....................................................................... 35

6 STEREO MATCHING IN BIOSCIENCE ........................................ 37
6.1 Bryophyte Scanning Device ...................................................... 37
6.2 Former Researches .................................................................. 38
6.3 Hardware Constructions .......................................................... 41
6.3.1 Linear Motion Controller .................................................... 41
6.3.2 Microstepping Unit Driver ................................................... 44
6.4 Stereo 3D Reconstruction ......................................................... 46
6.4.1 Camera Calibration ............................................................... 46
6.4.2 Additional Depth Estimators ............................................... 48
6.4.3 Comparison to Other Algorithms ......................................... 52
6.5 Surface Roughness ................................................................ 53
6.5.1 Clustering Algorithms with the Stereo Matching Constraint .... 55
6.6 Scanning Results in Biological Studies ....................................... 55
6.7 Conclusions ....................................................................... 57

7 BIMODAL DEPTH SEGMENTATION .......................................... 60
7.1 Bimodal Depth Segmentation as a Minimisation Problem ............. 60
7.2 Review of Chan and Vese’s Method .......................................... 61
7.3 Description of the Method ....................................................... 63
7.4 Tests and Results .................................................................. 68
7.4.1 Tests with Artificial Images ............................................... 69
7.4.2 Real-Life Applications of the Method ................................... 70
7.5 Summary ........................................................................... 75

8 STEREO MATCHING VIA MEAN SHIFT ATTRACTORS .............. 78
8.1 Mean Shift Algorithm ............................................................. 78
8.2 Matching in the Attractor Space .............................................. 80
CONTENTS

8.2.1 Attractor Clustering ........................................ 80
8.2.2 Attractor Matching .......................................... 81
8.2.3 Disparity Calculation ....................................... 82
8.3 Tests and Results ................................................ 83
8.4 Summary .......................................................... 85

9 STEREO SEGMENTATION VIA FUZZY C-MEANS ............ 86
  9.1 Fuzzy C-Means .................................................. 87
  9.2 Introducing the Matching Constraint to Fuzzy C-Means .... 89
    9.2.1 Stereo Matching Algorithm with Modified Fuzzy C-Means as the Cost Aggregation ...... 92
    9.2.2 Cluster Count Problem ................................... 94
  9.3 Tests and Results .............................................. 96
  9.4 Summary ........................................................ 98

10 SUMMARY .......................................................... 101
  10.1 Future Work .................................................... 102

11 APPENDIX .......................................................... 104
  11.1 Authors Bibliography ........................................ 104
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 3.1</td>
<td>Epipolar geometry</td>
<td>10</td>
</tr>
<tr>
<td>Figure 3.2</td>
<td>Image rectification</td>
<td>11</td>
</tr>
<tr>
<td>Figure 3.3</td>
<td>Ill-posedness of the correspondence problem</td>
<td>13</td>
</tr>
<tr>
<td>Figure 4.1</td>
<td>Example of the fixed window algorithm result</td>
<td>17</td>
</tr>
<tr>
<td>Figure 4.2</td>
<td>Illustration of the adaptive window</td>
<td>19</td>
</tr>
<tr>
<td>Figure 4.3</td>
<td>Illustration of the dynamic programming</td>
<td>22</td>
</tr>
<tr>
<td>Figure 4.4</td>
<td>Example of the dynamic programming algorithm result</td>
<td>23</td>
</tr>
<tr>
<td>Figure 4.5</td>
<td>Example of the disparity map calculated using a graph cut algorithm</td>
<td>25</td>
</tr>
<tr>
<td>Figure 4.6</td>
<td>Graph cuts and the graph construction</td>
<td>26</td>
</tr>
<tr>
<td>Figure 4.7</td>
<td>Belief propagation and message passing</td>
<td>29</td>
</tr>
<tr>
<td>Figure 4.8</td>
<td>The result of the segmentation based method</td>
<td>31</td>
</tr>
<tr>
<td>Figure 5.1</td>
<td>The disparity error occurrence</td>
<td>35</td>
</tr>
<tr>
<td>Figure 5.2</td>
<td>Middlebury stereo evaluation dataset (version 2)</td>
<td>36</td>
</tr>
<tr>
<td>Figure 6.1</td>
<td>Examined bryophyte samples</td>
<td>38</td>
</tr>
<tr>
<td>Figure 6.2</td>
<td>Rice's scanning device</td>
<td>39</td>
</tr>
<tr>
<td>Figure 6.3</td>
<td>A comparison of S. Rice and our scanning results</td>
<td>40</td>
</tr>
<tr>
<td>Figure 6.4</td>
<td>Bryophyte scanning devices</td>
<td>43</td>
</tr>
<tr>
<td>Figure 6.5</td>
<td>Nonlinear behaviour of the electrical circuit in stepper motor coils</td>
<td>45</td>
</tr>
<tr>
<td>Figure 6.6</td>
<td>Calibration images</td>
<td>48</td>
</tr>
<tr>
<td>Figure 6.7</td>
<td>Preview of the results from the different stereo matching algorithms</td>
<td>49</td>
</tr>
<tr>
<td>Figure 6.8</td>
<td>Disparity map enhancement based on additional depth estimators</td>
<td>51</td>
</tr>
<tr>
<td>Figure 6.9</td>
<td>Sample images from the experiments with Polytrichum formosum</td>
<td>56</td>
</tr>
<tr>
<td>Figure 6.10</td>
<td>Examples of the reconstructed bryophyte surfaces</td>
<td>58</td>
</tr>
<tr>
<td>Figure 7.1</td>
<td>Typical scene with two height levels</td>
<td>63</td>
</tr>
<tr>
<td>Figure 7.2</td>
<td>Projective transforms between images without occlusions</td>
<td>64</td>
</tr>
<tr>
<td>Figure 7.3</td>
<td>Illustrations of the level set terms</td>
<td>67</td>
</tr>
<tr>
<td>Figure 7.4</td>
<td>Artificial test images for bimodal depth segmentation</td>
<td>69</td>
</tr>
<tr>
<td>Figure 7.5</td>
<td>Results of bimodal depth segmentation in noisy images</td>
<td>70</td>
</tr>
<tr>
<td>Figure 7.6</td>
<td>Noise immunity test results</td>
<td>71</td>
</tr>
<tr>
<td>Figure 7.7</td>
<td>Comparison of depth segmentations with different algorithms (P. 1)</td>
<td>72</td>
</tr>
<tr>
<td>Figure 7.8</td>
<td>Comparison of depth segmentations with different algorithms (P. 2)</td>
<td>73</td>
</tr>
<tr>
<td>Figure 7.9</td>
<td>Level set results on the real-world images</td>
<td>75</td>
</tr>
<tr>
<td>Figure 7.10</td>
<td>The results of the plant layer segmentation using the level set method</td>
<td>76</td>
</tr>
<tr>
<td>Figure 8.1</td>
<td>Results of the attractor matching</td>
<td>83</td>
</tr>
<tr>
<td>Figure 9.1</td>
<td>The “Map” dataset</td>
<td>87</td>
</tr>
<tr>
<td>Figure 9.2</td>
<td>Illustration of the rationale behind the algorithm</td>
<td>90</td>
</tr>
</tbody>
</table>
List of Figures

| Figure 9.3 | Visualisation of the data points and their clusters | 91         |
| Figure 9.4 | The influence of cluster count on the output disparity map | 95         |
| Figure 9.5 | Experimental results of the modified fuzzy c-means algorithm | 97         |
| Figure 9.6 | Performance of the modified fuzzy c-means algorithm | 99         |
| Figure 9.7 | The segmentation results of the modified fuzzy c-means algorithm | 100        |
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>The performance on the manually entered dataset</td>
<td>54</td>
</tr>
<tr>
<td>8.1</td>
<td>The performance of the algorithm based on the mean shift attractors</td>
<td>84</td>
</tr>
<tr>
<td>9.1</td>
<td>The performance of the modified fuzzy c-means algorithm</td>
<td>99</td>
</tr>
</tbody>
</table>
This thesis presents our contributions to the fields of computer vision and robotics. We will introduce novel algorithms for stereo matching, depth segmentation and three-dimensional reconstruction and focus (not only) on the applications in the area of natural science, biology and environmental protection.

We will show how the general algorithms, used for the three-dimensional reconstruction of buildings, models and parts, can also be adapted to very specific biological observations. Our main focus will be on the three-dimensional reconstruction of the plant surfaces (mainly the bryophytes and lichens), but we will not neglect other, more common applications. While this application may seem quite unusual, we will show that the reconstruction process uses exactly the same methods of image processing as any other, e.g., industrial application. Of course, the algorithms cannot be exactly the same, but the main principles remain. This work describes the adaptation of the algorithms to the specific tasks. The following text is written from the perspective of computer science, in a clear and comprehensive way ensuring the reader is not required to have any prior knowledge or understanding of biology.

In the sequel, we will introduce the stereo matching problem, describe the state-of-the-art approaches and then present our own contributions. First, we will describe our construction of a unique scanning device, specially adapted for biological studies. Next, we will cover the necessary steps leading to the three dimensional reconstruction of the scanned samples. And finally, we will present three novel algorithms, invented during the work on the device. The invented algorithms will be also presented in other applications, not only the biological ones.

How did we come to such extraordinary topic? The original idea came from the discussion with our colleagues working at the Institute of Environmental Engineering. The researchers there are studying the structural changes in plant tissues caused by various environmental agents, e.g., air pollutants, soil toxins, or groundwater contamination. One of the biggest problems they were facing was how to acquire and measure the subtle changes of the plant tissues. They already tried several techniques used for scanning the surfaces, three devices based on a contact scanning probe and projected laser beam, but without any significant results. These methods, when used to scan the plant surfaces, were found inappropriate mainly due to the immobility of the scanning devices and the emitted laser radiation that influenced the photosynthetic reactions.

We were asked whether we could help them with their specific problems. Our first suggestion was to use a passive stereoscopic scanner, based on analysing the slight differences between the images taken by two cameras, mounted slightly apart above the
1.1 Stereo Correspondence Problem

The stereo correspondence problem (also called “the stereo matching”) has been studied for years in a more or less futile attempt to find a perfect matching algorithm. An algorithm that can fully mimic the human brain in visual perception, an algorithm capable of recovering high quality depth maps even from the highly complex scenes. Fortunately, no successful substitute for brain was found and will never be found, but the research is still going on. In our opinion, the idea that the computer should use the same mechanisms (the stereopsis) as the animals is rather misleading. We can think of other, more effective and efficient ways of obtaining depth information, but the binocular stereo systems and the stereo matching algorithms form the main streams in image processing.

The two-frame stereo matching algorithms have been intensively studied for the last few decades and we have to admit that there are many algorithms and heuristic methods that provide very good results. But most of them provide the exceptional results only for the specific types of images, especially for the images from the popular Middlebury dataset (which will be described later on). In addition, we might even say that some of these algorithms are over-optimized to give the best results for exactly these four specific image pairs. We consider this as a major drawback in the development of the algorithms, as the test images do not represent all possible applications of the stereo matching algorithms, neither the reconstruction. An example of the failure of the general algorithms is presented in our application of the plant surface scanner.
1.2 Thesis Goals

In this thesis we are not going to challenge the mainstream approaches, nor do we create a new general purpose stereo matching algorithm. Instead of this, we would like to address the problems that are beyond the scope of the traditional algorithms. We will focus on solving the correspondence problem in the specific scenes and environments, sharing some particular characteristics, e.g., the a priori knowledge of the scene configuration.

1.2 Thesis Goals

The goal of this thesis is to propose methods, techniques and tools suitable for the stereo vision systems. Beside the theoretical novelty of our approaches, we would like to emphasize the practical aspects. In the following text we will address several practical applications, e.g. the surface reconstruction of the bryophyte canopies, the detection of free parking lots, etc. Our research was motivated by the practical needs and failures of the classical methods. The goals of the thesis can be summarised as follows:

- Provide a review and analysis of the state-of-the-art stereo matching algorithms.
- Propose new algorithms and improve the current ones in order to enhance the matching accuracy, reliability and robustness in the specific fields of interest.
- Demonstrate the benefits of a built-in a priori knowledge of the scene context for the stereo matching and segmentation purposes.
- Develop a mobile 3D scanning device for the analysis and evaluation of the biological structures and their surface properties.

The following section summarizes the principal findings and implications during our doctoral studies and describes the significant contributions of the thesis as a whole.

1.3 Main Contributions

The dissertation makes the following main contributions to the fields of computer vision and robotics:

- We have constructed a unique scanning device based on the 3D stereo reconstruction technique. The device was invented for scanning the surface structures of small biological samples, mainly the bryophytes. We have performed several tests and comparisons and according to them, our device surpassed all other similar solutions. The description of the device and the comparison results were published in [84] and [85]. The device was successfully used in several biological studies, e.g. [106, 104, 91]. This topic is described in depth in Chapter 6.
• The important part of our scanning device is the camera positioning system. We were forced to develop a new microstepping unit as we detected certain faults of the commercially available units. According to our experiments, we have achieved higher accuracy than the generally available units. The design of the microstepping unit was published in [82] and [112].

• In Chapter 7 we have proposed a novel algorithm for bimodal depth segmentation. This method separates the background and the objects of arbitrary shapes lying on two distinct disparity planes. The problem is solved as a problem of minimising a functional using the Euler-Lagrange equation and the level-set function. The algorithm was compared with other approaches and the results were very satisfactory. We used this method for segmenting the layers of the bryophyte canopies and recently, we have also applied this algorithm for detecting the cars on parking lots. The algorithm was presented in [83].

• A novel concept of the stereo matching algorithm using the mean shift attractor space is introduced in Chapter 8. The experiments showed that by using this approach, we are able to perform fast and precise matching of large segments, especially on images containing large homogeneous areas (e.g., the blocks of buildings). The description of the algorithm and the experimental results were published in [81].

• Finally, we have proposed a modification of the popular fuzzy c-means algorithm. Our new stereo segmentation method performs the clustering in the disparity space. The motivation was to provide an algorithm that can separate objects based on their different colour and spatial depth. The algorithm was originally developed for segmenting the moss clusters but can be used for other similar purposes.

1.4 THESIS OUTLINE

This thesis is divided into two main parts. The first part is more general and examines the various approaches to understanding the stereo algorithms. This part starts with a brief historical excursus on the binocular vision research and the development of the first stereo reconstruction systems. Chapter 3 presents an introduction to the topic, describes the epipolar geometry and its meaning for the stereo matching algorithms. Chapter 4 provides an in-depth look into the state-of-the-art algorithms. Although, the content is not exhaustive, it gives a fairly representative sample of the different approaches used nowadays in the area of the stereo matching. Chapter 5 describes the methodology used in the design of the experiments.

The second part of the thesis focuses on the main thesis goals – the algorithms for the depth segmentation and stereo matching, and their applications. The following chapters present our own contributions and research results.
Chapter 6 describes the construction of a unique scanning device, invented especially for biological experiments. Moreover, this chapter presents our enhancement of the linear motion controller used for the positioning system of the camera. The results show that our device significantly outperforms other scanning devices used for the same purpose. Chapter 7 describes a novel algorithm for bimodal depth segmentation. The method, we have invented, separates the planar objects of arbitrary shapes lying in a certain height above a base plane using the data from the stereo image pair. The algorithm was successfully applied for moss layer segmentation and a parking lot occupancy detector. The following Chapter 8 presents an algorithm based on matching the mean shift attractors. The results show the advantage of this matching algorithm on the images containing large homogeneous surfaces (e.g., blocks of buildings). Chapter 9 introduces a novel extension of the popular fuzzy c-means clustering algorithm. The algorithm can be used for the segmentations (in our context mosses and base structures), as well for obtaining the depth maps.

Finally, Chapter 10 summarizes the main conclusions of this thesis and presents an outlook for the future work.
Before we begin describing the algorithms and mathematical models, we would like to take a short detour to give the reader an idea of the long history of the stereopsis. We find it very intriguing, but the reader can skip this and go directly to Section 3, if he or she is not interested in the history.

The stereopsis has been the subject of interest for generations of thinkers, philosophers, researchers and practitioners. They were seeking answers to questions like, “How do we perceive things in three dimensions?” or “Can we perceive depth using only one eye?” The answers to these questions create the content of the most fundamental level of the depth perception. As we will see, the history of the stereopsis goes deep into the past.

2.1 ANCIENT TIMES

The earliest vision researches and experiments are attributed to the Ancient Greeks. Probably the first mechanism of vision was described by Empedocles, a philosopher of Agrigentum. Two hundred years later, Eukleides wrote the earliest known textbook on optics, where he introduced the elementary mathematical models. Most of his definitions remained influential for the next centuries and some are still valid until now [67]. The book became a classic for all vision researchers.

After centuries, Alhazen’s Book of Optics (De Aspectibus), which was inspired by Greek science, made the first attempt to formulate the binocular vision geometry. The first definition of the binocular disparity appears surprisingly centuries before Hering, to whom it is attributed. Alhazen created a comprehensive theory of vision, which was not surpassed until the seventeenth century. His book contains definitions of basic terms related to the stereo vision – the correspondent points, binocular fusion, double vision of non-corresponding points, etc. [93].

However, these books were unavailable to the most of European thinkers and the knowledge was almost lost. The legacy of Greek and Roman scientists such as Eukleides, Aristoteles, Galen, and Alhazen was revived much later in medieval Europe by philosophers Roger Bacon, John Peckham, and Erazmus Witelo. In the fifteenth century, Leonardo da Vinci noticed that two eyes obtain different views of the same scene. He drew many figures and sketches illustrating occlusion zones in various configurations, pointing out that occlusion zones can create an impression of depth. Today’s researchers, Nakayama and Shimojo suggested that the depth created this way should be called da
Vinci stereopsis [2]. Later, in the sixteenth century, Giovanni Battista della Porta invented the first visual tests of eye dominance and described the binocular rivalry [138].

A big breakthrough came in 1838 thanks to Charles Wheatstone, who published a paper addressed to the Royal Society proving an undeniable connection between the disparity and perceived spatial depth. Wheatstone is also known as the inventor of the first modern stereoscope, a device allowing each eye to see completely different images [13].

Until late sixtieth, binocular vision was attributed to the functions of higher human cognitive processes. It has been widely assumed that only mammals with frontally placed eyes are able to perceive spatial depth. Ramón y Cajal discovered that the inputs from the corresponding regions of two retinas converge on so-called isodynamic cells and form a uniform perception of depth [48]. Cajal’s work also covered the topics of the binocular vision, describing a predecessor of the modern random dot stereograms. In 1979, Marr and Poggio presented a theory of human stereo vision [99]. Marr merged the results from psychology, artificial intelligence, and neurophysiology, and founded the new discipline of Computational Neuroscience. This marked the beginning of the modern era of the binocular stereopsis that continues today.

2.2 MODERN ERA

The first modern research in the stereo vision began in the mid-1970s and was largely supported by the Defense Advanced Research Projects Agency (DARPA). The start of the computer era was, in fact, the turning point in the vision research. At the beginning, the stereo reconstruction was considered to be too computationally expensive but, thanks to the rapidly increasing computational power of computers, the problem was solvable.

However, the stereo vision is not an exclusive domain of the computer scientists as it may appear. Researchers have often taken inspiration from other areas such as psychology, neurophysiology, or cartography. Early stereo systems have been primarily concerned with human ability of perceiving the depth. Computational models of human stereo vision have investigated the human stereo fusion. These early experiments have used the synthetic random dot stereograms invented by Bélo Julesz [71].

In the cartography, the stereo vision was primarily used for the automatic stereomapping, capable of handling different kinds of terrain and sensors. Aerial imagery usually contains both natural features as well as man-made objects that can confuse the stereo system.

The stereo matching has also found many applications in military and aerospace constructions such as cruising missiles and autonomous vehicles. For example, Lockheed study from 1980s focused on the navigation of an aerial vehicle capable of tracking ground control points along the flight path. Matching was accomplished using the normalized cross-correlation in the spiralling grid search [56]. In 1977, Stanford Artificial
Intelligence Laboratory started a work on a visually guided roving robot using the stereo vision [103].

With increasing number of various algorithms, first comparisons and benchmarks appeared. Bernard and Fischler [6] conducted an in-depth review of the early computational stereo algorithms. The later development and evolution of the stereo matching algorithms has been traced by several other authors [20, 40, 116]. In the recent decade, we can see a shift from the local methods to the global approaches. New ideas, such as modelling in terms of energy minimization, probabilistic modelling, etc., were introduced.

Still, the current stereo matching algorithms are not yet capable of duplicating the steps and processes taking place in the human brain. At present, it is impossible to incorporate all semantic and physical knowledge of human mind to filter out the errors produced by the matching process. Constraints and certain principles that are innate to humans must be directly embedded into the algorithms. While modern approaches can partially compensate the lack of human knowledge and automatically resolve some ambiguities and matching errors, they are far from being perfect.
The correspondence problem is one of the main problems solved in the stereo vision. Given two or more images representing the different views of the same scene, we need to establish a correspondence among homologous features. By homologous, we mean identical or similar features that are projections of the same object in each view. In most cases, the aim is to determine the disparity of all pairs of corresponding features of both images. The disparity is the relative displacement – the difference in the position of corresponding features. It is interesting to note that the term disparity was firstly used in psychology of human vision. The disparity and the imaging geometry is the main key to retrieve the depth information. But the disparity function is not the only possible output of the stereo matching algorithm. Other approaches may use voxel-based or layer-based representations.

Before we can discuss the stereo matching, we must examine the camera configuration and the consequent geometry. This description is required as a necessary prerequisite for describing the function of the stereo matching algorithms.

3.1 Stereo Geometry

Two images $I_1$ and $I_2$ captured by pinhole cameras depict a physical point $X$. Figure 3.1 shows such a situation. We assume that the two cameras are identical (in practice, they may differ) and are calibrated. Interior parameters, as their optical centres, focal lengths, and lens distortions and extrinsic parameters, i.e., the relative positions and orientations of each camera are known.

As we can see from Figure 3.1, there is one 3-D space $(x, y, z)$, referred as the world reference frame and two 2-D systems on each retinal plane $(u_1, v_1)$ and $(u_2, v_2)$. The position and orientation of both cameras is usually given by the relative orientation, which is specified by a pair of rotation matrix $M$ and translation vector $T$. The distance between the two optical (perspective) centres $C_1$ and $C_2$ is called a baseline and is specified by the translation vector $T$. What we analyse are two images formed in the retinal planes $R_1$ and $R_2$. In the next chapters, we will also refer to them as the left and right image.

The correspondence problem can be formulated as follows: for a point $p_1$ located in the retinal plane $R_1$, find the correspondent point $p_2$ in the retinal plane $R_2$. By the term correspondent we mean that the points are the projections of the same physical point $X$. 
3.1 Stereo Geometry

The disparity refers to the (usually small) differences in coordinates of corresponding points.

At first, it might seem almost impossible to find the correspondent point since we have to search the entire second image. As we will see later, it is not necessary to look through the entire image. We are able to narrow down the search to a small area of possible matches.

3.1.1 Epipolar Geometry

The relationship between two views is defined by the epipolar geometry (illustrated in Figure 3.1). As you can see, the directions of the camera optical axes are not parallel to the world z-direction. This is the typical case for biological vision systems, where eyes have a convergent angle toward the target object.

Suppose we know the exact locations and orientations of both cameras, and we want to locate the correct position of the point $X$. Based on the figure, we can make the following observations. Let us have a look at the half-line $\langle p_1, C_1 \rangle$. We can place the point $X$ anywhere on this half-line without changing the location of projected point $p_1$. The points $C_1$, $C_2$, and $X$ define an epipolar plane. The lines $e_1$ and $e_2$ are the intersection of the epipolar plane with retinal planes $R_1$ and $R_2$. The lines $e_1$ and $e_2$ are called the epipolar lines. Since the centres of projection of the cameras are distinct, each centre of projection projects onto a distinct point into the other camera’s image plane. These two points $E_1$ and $E_2$ are called epipoles.
The epipolar constraint guarantees that, for a given point \( p_1 \) in the retinal plane \( R_1 \), all possible matches must lie on a line. Suppose we are able to find the correct matching point \( p_2 \). We can make a half-line \( (p_2, C_2) \), which is a back-projection of \( p_2 \). The point of the intersection \( (p_1, C_1) \) and \( (p_2, C_2) \) is the point \( X \). In a general case of non-parallel stereo geometry, \( X \) is not on a horizontal epipolar line (as seen in Figure 3.1).

The most important advantage of the epipolar constraint is that the correspondence problem is simplified to one dimensional problem when both planes \( R_1 \) and \( R_2 \) are parallel to the line going through the projection centres \( C_1 \) and \( C_2 \). Non-parallel epipolar geometry can be transformed to parallel by the process of stereo rectification.

3.1.2 Rectification

Rectification transforms the images so that the epipolar lines are aligned horizontally. In the past, images were rectified mainly using optical techniques [128], but this approach was quickly replaced by modern computational algorithms.

The operation of rectification creates simple epipolar geometry – the epipoles are at infinity and the epipolar lines are parallel to the image rows, aligned horizontally. One of the possible solutions described by [44] is illustrated in Figure 3.2. Rectification can be achieved by projecting the images onto a single plane \( R \), using the same optical centres. Given a pixel \( m_i \) in the original image (retinal plane \( R_i \)), we construct a new pixel on plane \( R \) by intersecting the line \( (C_i, m_i) \) with \( R \).

In order to get the epipoles at infinity, the plane \( R \) must be positioned parallel to the line connecting the optical centres \( (C_1, C_2) \). Since there are an infinite number of such planes, we need additional criterion. From the remaining two degrees of freedom in the system, only the orientation is relevant. The distance of plane \( R \) to the line \( (C_1, C_2) \) has...
a direct impact on the scale. The remaining orientation parameter can be chosen either
to minimize the distortion of the projected images or by imposing the assumption that
$R$ is also parallel to the line $(P_1, P_2)$, which is the intersection line of the original images.

The process of image rectification and resampling of the stereo images taken from
widely differing viewpoints is described in details in [57]. An algorithm for rectification
of stereo image pair without the need for any calibration was proposed in [113].

3.2 CORRESPONDENCE AND RECONSTRUCTION PROBLEM

Stereo correspondence or matching is the fundamental part of stereo vision. The cor-
respondence problem consists of determining the locations of features in each retinal
plane that are the projection of the same physical feature in space. Given the corre-
sponding projections, we want to compute the 3-D coordinates of the projected physical
features in the world reference frame. This is the reconstruction problem.

The task of identifying the correspondent features and determining the disparity is a
difficult one. The problem is said to be ambiguous. For better understanding, see the
example in Figure 3.3. Each camera views three primitive dots. The problem is: which
dots in the left image correspond to the dots in the right image. There are multiple
possibilities for matching the inputs. The nine dots $X_1, \ldots, X_9$ represent all possible
correspondence combinations of these three points. To solve the difficulty of matching
process, we use several constraints to reduce the space of potential matches [44]. The
constraints can be divided into the following groups:

1. Constraints imposed by the geometry of imaging system. The most important
constraint in this case is the epipolar constraint (see Section 3.1.1).

2. Geometric constraints arising from the object structure. An example of this cate-
gory is the continuity constraint (see Section 3.3.4).

3. Physical constraints arising from the object interaction with the illumination. (Since
they require different techniques for acquiring the images, we will not discuss
them further. For more details, see [37]).

Stereo matching is inherently an ill-posed\footnote{Ill-posedness (questions mal posées) is a mathematical term, defined in 1902 by Jacques Hadamard, a French
mathematician [8]. The term refers to a problem which may have more than one solution, or in which the
solutions depend discontinuously upon the initial data. Also known as improperly posed problem.} inverse optical problem due to the difficul-
ties arising from the untexture regions, depth discontinuities, and occluded pixels. This
practically means, there might be more than one solution for the given problem. The
inherent ill-posedness of the problem is solved by adding a smoothness term, reflecting
some a-priori preference of the solution.

Ill-posedness (questions mal posées) is a mathematical term, defined in 1902 by Jacques Hadamard, a French
mathematician [8]. The term refers to a problem which may have more than one solution, or in which the
solutions depend discontinuously upon the initial data. Also known as improperly posed problem.
3.3 MATCHING CONSTRAINTS

It is a combination of several constraints (sometimes working even against each other) that severely reduces the total number of potential matches. The simplest kind of constraint is used in the intensity-based approach. It is assumed that two corresponding points have the same or almost the same colour. Unfortunately, this assumption is generally valid only in the Lambertian world. In the real world this assumption is frequently violated by the nature of used image sensors. Another constraint, the epipolarity, was already described in Section 3.1.1.

In this section we briefly introduce additional constraints that help in selection of the valid matches among the possible candidates. Most constraints arise from the knowledge of the scene configuration and are usually implemented as relations between the matched features.

3.3.1 Ordering constraint

Since the structures and features depicted in both images are very similar, the relative position of two features should not differ from the relative position of the corresponding features in the second image.

The ordering constraint preserves a pixel order along the epipolar lines in both images. This means that points from the left and right images have the same order on
the epipolar line. This constraint is met most of the time, but is violated in so called forbidden zone [44].

3.3.2 Uniqueness Constraint

Uniqueness constraint is enforcing a one to one mapping between the pixels of both images. It was firstly formulated by Marr and Poggio in 1976 [99], and it practically means that there is at most only one valid match for each primitive in the scene.

This assumption is true for opaque objects but may be violated in the scenes containing transparent objects or reflections. Uniqueness constraint is also violated on horizontally slanted surfaces that produce areas of unequal pixel width. The experiments showed that the good correspondences can be established using just the previous two constraints.

3.3.3 Geometric Constraints

In the simplest form, the geometric constraints restrict the object shapes to be locally planar, i.e., assumes that they can be approximated by the tangent planes. More general cases are also possible. An example of such approach is the algorithm based on the assumption that the viewed scene contains two planar surfaces [96].

3.3.4 Continuity Constraint

The continuity assumes that the disparity varies smoothly across the scene. According to this, the physical world is composed of continuous surfaces with the minimum boundaries. This constraint is used in, e.g., the dynamic programming approach.

It is clear that this constraint must fail in situations where the continuity breaks on object boundaries. The scenes containing thin foreground objects or planes with narrow holes also violate the rule. Hopefully, the discontinuities caused by the object borders appear only on a small fraction of the image.

There are several approaches to address the continuity constraint, e.g., [118] showed that the smoothness of the scene surfaces is closely related to the disparity gradient and can be used to calculate the disparity between a pair of close matches. The rule says that the ratio of the disparity difference of two candidate matches and the distance between them must be less or equal to a given threshold [116]. A similar coherence principle by evaluating the similarity of the disparity of nearby matches was formulated by Prazdny [119]. For practical reasons, we usually have to determine the proper disparity range. The greater the range of disparity is, the greater the number of false matches can be. A set of global restraints is used to limit the maximum allowable disparity between the images, as well as the maximum value of the disparity gradient.
The basic idea behind many stereo matching algorithms is the observation that in the real world, a sufficiently small area usually changes only a little from image to image. The question is what type of measure should be used to find the correct match. As a result, a variety of measures have been introduced, ranging from the intensity based approaches through the vector and matrix data evaluation to more complex statistical methods. In this chapter we will provide a summary of the main approaches to stereo matching problem.

In general, the stereo matching algorithms perform the computation in four successive steps. These steps involve: (i) an estimation of the initial disparity and matching cost for each pixel, (ii) an aggregation of the pixels sharing the same disparity, (iii) a computation of the disparity map and, (iv) a refinement of the disparities with respect to the aggregation and other constraints.

The stereo matching algorithms are developed and based upon different theoretical approaches and are subject to various tests and evaluations. At this moment, the most commonly cited taxonomy of the stereo matching methods was published by Scharstein and Szeliski [125]. Their framework unifies the testing procedure and simplifies the result comparison between the different algorithms. The methodology is discussed later in Chapter 5.

4.1 BASIC CLASSIFICATION

The basic classification system of the stereo matching algorithms divides the methods into two groups based on the type of output disparity map they produce – the dense and sparse. In the practical applications, the most demanded are the dense disparity maps, with almost all pixels having the disparity value assigned. The sparse disparity maps provide the disparity values only for a limited number of pixels (significant points, edges, etc.). These estimations are, however, more reliable (when compared to the dense maps). The sparse algorithms can be followed by a surface fitting step, and so produce the dense outputs.

The other classification system for the stereo matching algorithms refers to the size of the processed area. In this way, we recognize the local and global methods. Local stereo matching algorithms generally place a small matching window around a pixel in one image. A window of the same size and shape is then placed in the second image. Thereafter, all possible positions on the epipolar line are searched for the point
4.2 WINDOW BASED METHODS

Window based methods may be considered outdated or inadequate, but in our opinion, they create the most fundamental approach for the stereo matching, at least from the historical perspective. They are quite simple and easy to implement in hardware. Whenever a new parallel architecture is introduced, these, or very similar algorithms are adapted to demonstrate the benefits of that specific architecture.

Window based methods are matching the pixels of the left image to the appropriate pixels of the right image, based mostly on the colour equality (similarity). Frisby and Pollard stated that since in practice the stereo projections do not preserve photometric invariance, any attempt to match two individual pixels (of the left and right image) is doomed to failure [87]. But, as we will see, the situation is not so hopeless.

In the local stereo matching methods, the correspondence of a pixel is based on the similarity of its neighbourhood. In this way, we can reduce the ambiguity of the match. The methods are mainly utilizing the colour intensities to determine the correct disparity value. Generally, we take a pixel of the left image together with its nearest finite neighbourhood and match it with the pixel of the right image with the most similar neighbourhood. In the classical correlation algorithms, the neighbourhood is usually defined as a rectangular window centred on the processed pixel. The similarity itself can be computed using the measures such as the sum of the absolute differences (SAD), sum of the squared difference (SSD), normalized SSD, normalized cross-correlation etc. A comparison of the different similarity measures can be found in [35].

The basic algorithm computes the disparity in the following simple way: First of all, a small window (usually square-shaped) is created around each matched pixel. Then, the matching costs for all allowed disparities are calculated and the one that minimizes
4.2 WINDOW BASED METHODS

Figure 4.1: An example of the fixed window (9 × 9) algorithm result (the ground truth data of this standard dataset can be found in Chapter 5). The left image shows the disparity map as obtained from the algorithm. The typical errors are highlighted in the green boxes and zoomed on the right. The errors (in box 1) are caused by the simple winner-take-all strategy and the implicit assumption of frontal-parallel surfaces. Aggregating the matching costs in the window area results in another typical error, an inaccurate localization of the depth transition (visible smoothing in box 2).

the error value is declared to be the best match. The winner-takes-all strategy usually produces quite satisfactory results.

One of the main problems connected with this approach is how to determine the optimum size of the window. We have to decide whether to use small or large windows. Small windows are processed fast, but are more likely to produce false matches. In contrast, larger windows can capture sufficient variation for handling featureless regions, but tend to blur the disparity edges. Moreover, it is unlikely that large window will contain only a single disparity value and thus correct disparities are averaged. Therefore, the implicit assumption for this type of algorithms is that the disparities of all pixels within the window are the same. As a result, window correlation is ineffective at the depth discontinuities (see Figure 4.1).

4.2.1 Multiple Windowing

The main reasons why the fixed-size windows fail are: (i) the assumption of frontal-parallel planes, (ii) no mechanisms for handling the depth discontinuities, (iii) the inability to handle uniform areas, (iv) the ambiguity of repetitive patterns.

Therefore, a lot of efforts have been made to solve the problem of selecting proper window size and shape. For example, Fusiello et al. [51] tried to use windows with different sizes. The correlation was calculated with several different windows and the
disparity with the smallest error rate was retained. The windows were asymmetric and represented different disparity profiles. Since the method uses only a limited number of windows, it cannot cover all possibilities of different shapes and sizes. Additionally, the algorithm assigns the uncertainty value to all pixels, defining the variance of the disparity measure for different windows. These uncertainty values are used to help fill the gaps in poorly textured regions.

In a similar fashion, Hirschmüller et al. [62] proposed an algorithm using multiple matching windows, but in contrast to the previous one, not necessarily rectangular. Surprisingly, later approaches (based on the publishing date) used even simpler strategies, e.g., Scharstein and Szeliski [125] employed only fixed size windows while moving only the support centre. According to our experiments, such solution provides only slightly better border localization.

4.2.2 Adaptive Windows

An ideal algorithm would extend the window to include as much points at the same depth as possible. Notable improvements in this direction have been achieved by Kanade and Okutomi [73] using a variable window that dynamically adapts its size according to the pixel surroundings. The authors developed a statistical model employing the intensity differences, noise characteristics and disparity uncertainty as the main parameters influencing the window size and shape.

The window shape is iteratively expanded as far as the level of the uncertainty remains low (Figure 4.2). The uncertainty of the disparity estimation is derived from the intensity fluctuation inside the window. The algorithm iteratively searches for the best windows, starting from the smallest ones and expanding to larger areas of various shapes. The left image of Figure 4.2 shows the situation where the expansion is still possible since the expansion step will absorb the pixels of the same intensity. In contrast, the right figure shows the situation where the further expansion is not possible since the neighbourhood pixels significantly differs from the pixels within the window. The stereo matching process is iterated until the disparity estimates converges, or until the maximum number of iterations is reached.

Initially, it is assumed that the disparity is constant across the window, but later, during the calculations, it is expected that it may fluctuate more, especially at points located farther from the window centre. In a similar way, it is assumed that the pixels colour inside the window should be the same as the centre point, but the certainty of this presumption is decreasing farther we move from the centre. We may say, the algorithm inherently prefers regions with constant pixel intensity.

An advantage of this algorithm, compared to the fixed size window algorithms, is that the smooth surfaces are recovered correctly while sharp disparity edges are retained. On the other hand, it is sensitive to the initial disparity estimations.
4.2 WINDOW BASED METHODS

Figure 4.2: Adaptive window algorithm – the process of window expansion. The left figure shows the situation where the expansion is still possible since the expansion step will absorb the pixels of the same intensity. In contrast, the right figure shows the situation where the further expansion is not possible since the neighbourhood pixels significantly differs from the pixels within the window.

4.2.3 Multiple Baselines

The window based approaches are not limited to only two-frame stereo matching algorithms. An improvement in accuracy and reliability can be reached using more than two image frames.

A classical example of a multiple baseline approach was published by the well-known authors, Okutumi and Kanade [111]. The method is reducing the number of SSD mismatches by combining the SSD values from the different baseline images.

The motivation is the following: it should be easier to find a match between the stereo pairs with shorter baselines, since the areas are more similar and disparities smaller. However, small differences in disparities lead to greater errors in distance estimation due to a narrow triangulation. In contrast, stereo pairs with longer baselines provide better distance estimations at the expense of a higher error rate of the stereo match. The minimization of the sum of SSD values of all pairs provides a better estimation than the single baseline method. A similar algorithm has been earlier investigated in [135].

4.2.4 Modern Approaches

The original adaptive window algorithm has been significantly improved in many aspects. For example, an arbitrarily shaped connected window improves the performance near discontinuities, and also propagates information from textured regions into nearby regions without a texture [17].

Gerrits and Bekaert [53] exploited an assumption that the disparity discontinuities occur mainly at the colour boundaries. The reference image is segmented and all win-
4.2.5 Parallel Concepts

An efficient implementation using a specialized hardware can offer a significant speed-up over the straightforward CPU implementations. Simple techniques are based on reusing previously computed values [45] or reducing the number of steps by a winner-update strategy [25].

Recently, implementations on GPGPUs have become a field of active research. GPGPUs are extremely suitable for massive parallel calculations without many dependencies between the processed data. A multi-resolution adaptive window algorithm [151] is optimized for the GPU. Beside GPGPUs, several stereo matching algorithms were also successfully implemented on FPGA circuits [100, 70, 148]. For a reference, see Kalarot and Morris’s [72] comparison of FPGA and GPU implementations of real-time stereo matching algorithms.

To conclude this section, we remark that the biggest advantage of the window based methods, despite many drawbacks, is their speed and simplicity, which make them suitable for a wide range of real-time applications.

4.3 Dynamic Programming

The stereo matching problem can be solved by, e.g., the minimization of the energy function, simplified for ease of understanding as

\[ E(d) = \sum m(d_p) + \lambda \sum s(d_p, d_q), \]  

(4.1)

where \( m(d_p) \) is the matching penalty (e.g., absolute pixel difference) for assigning disparity \( d_p \) to pixel \( p \) and \( s(d_p, d_q) \), the smoothness penalty for assigning disparities \( d_p \) and \( d_q \) to neighbouring pixels \( p \) and \( q \).

Dynamic programming is an algorithm design that, in general, solves optimization problems involving a sequence of decision making. The algorithm solves different parts of the problem and then combines the partial solutions to an overall solution. The method is based on Bellman’s principle of optimality, which asserts that optimal policies
have optimal sub-policies. This principle is also known as the "optimal substructure" property [89].

The problem can be also viewed as a path finding problem on a 2D plane. The path goes through a set of nodes representing the possible matches between the image scanlines. Unlike iterative and cooperative methods, the dynamic programming does not need any set of weights and real values that must be threshold or evaluated. Instead of that, the dynamic programming approach searches for the optimal set of binary weights that lead directly to the correspondence candidates. An N-stage decision process is solved as N single-stage processes, consequently logarithmically reducing the computational complexity.

Therefore, the method is much faster than the most iterative stereo algorithms. Furthermore, the proper operation is dependent only upon the validity of the ordering constraint. The stages must be ordered that means that the results which are needed for the current stage must be already processed before. Formally, the decision process must be Markovian [47] – the selection of an action must depend only on the current state of the system.

The basic stereo matching algorithm using the dynamic programming technique can be roughly described as follows: Compute a cost matrix $M_h$ for each scanline of the left $I_l$ and the right $I_r$ image. $M_h$ is the cost matrix between the correspondent pixels $p(i, j)$ and $q(i + d, j)$, for a disparity value $d$, where $i + d < N$. The matrix $M_h$ has dimensions $N \times M$, where $N$ and $M$ are the number of measurements on each image scanline (if matching individual pixels, $M = N$). The cost of a matching $M_h(i, j)$ is defined as

$$M_h(i, j) = \min(C_d(i, j) + M_h(i - 1, j - 1), M_h(i - 1, j) + \lambda, M_h(i, j - 1) + \lambda),$$

where $\lambda$ is a penalization to enforce the continuity constraint (more details in [32]), and $C_d(i, j)$ is a local similarity function, e.g., simple intensity difference.

In other words, the matrix $M_h$ holds the values of the sub-problems. Each element of $M_h(i, j)$ is the minimum cost of a correspondence between the first $i$ pixels of $I_l$ and the first $j$ pixels of $I_r$. The disparity map of each line is found by calculating the shortest path matrix. To avoid recalculation of the shortest path, the algorithm holds a predecessor matrix that stores the predecessor of each node on the shortest path.

Thanks to the efficiency, the dynamic programming is used in many algorithms, e.g., [32, 52, 136] and despite the occurrence of vertical inconsistencies (streaking artefacts), it is recommended for applications where speed is an important factor, e.g., in real-time applications [30].

4.3.1 Simple Connectivity Constraint Approaches

The first implementation of this approach was an edge-based line-by-line stereo correlation algorithm [5]. The algorithm consists of several steps for extracting edges, linking
4.3 DYNAMIC PROGRAMMING

Figure 4.3: The idea of dynamic programming. The correspondence problem can be formulated as a finding a path in the graph. Minimum cost path is shown together with matched pixels.

them to obtain the edge connectivity structure and correlating the edge connectivity structure of the two input images. The process of the correlation requires a huge combinatorial space. Even with extensive heuristic pruning, it runs extremely slow. Therefore, the authors proposed Viterbi algorithm, a dynamic programming technique to perform the correlation process. The simple continuity constraint is assured by matching the edges first and then interpolating the untextured areas.

4.3.2 Intra- and Inter-Scanline Search

Processing each scanline independently is one of the biggest drawbacks of the first implementations. Small local errors are clearly visible in Figure 4.4.

The problem with these vertical inconsistencies was addressed in an intra-scanline search algorithm \[110\]. The correspondence problem was transformed to the problem of finding the matching path on a two-dimensional search plane, where the vertically connected edges in the images provide consistency constraints across the search planes. In contrast to \[5\], the intervals between the edges are matched instead of matching the edges.

The main contribution of the Ohta-Kanade algorithm \[110\] is the so-called “inter-scanline search”. The inter-scanline search can be viewed as a search for a set of paths in a 3D space, created as a stack of 2D planes from the standard (intra-scanline) search described in the previous section. The inter-scanline search seeks for the possible correspondences of the connected edges.

The algorithm works effectively on images containing long connected edges, where the inter-scanline constraint can be fully utilized. In the absence of long edges,
algorithm is reduced to the ordinary 2D search, matching isolated edges within each scanline separately.

4.3.3 Dynamic Programming on a Tree

The disadvantage of dynamic programming is that it is generally built upon a single scanline. The results from the single scanline approach can be, however, improved by applying a more suitable optimization on a tree structure. It has been shown that dynamic programming can be applied to graphs without loops, in particular, to trees. Two possible tree structures were proposed – the MID tree and the MIDDT tree [136]. The MID tree stands for the minimum intensity difference tree. The tree contains the minimum sum of the intensity differences across the edges. The algorithm performs the minimization through a construction of a minimum spanning tree. The MIDDT tree extends the MID tree by a distance transform information. The proposed optimization is, in fact, a hybrid approach between a 1D optimization and 2D methods, since it operates only among chosen directions given by the tree structure. The algorithms operating on the tree structures typically run faster.

4.4 GRAPH CUT ALGORITHMS

The stereo matching problem is often formulated in terms of minimizing an energy function subject to a set of constraints. The energy can be given by:

\[ E(d) = E_d(d) + \lambda_s E_s(d) + \lambda_o E_o(d), \]  

(4.3)
where \( E_d(d) \) is the data term, \( E_s(d) \) smoothness term, \( E_o(d) \) occlusion penalty and \( \lambda_s, \lambda_o \) are weights between the terms. The data term is usually defined as a cost of the matching (usually the absolute intensity difference); meanwhile the smoothness term penalizes disparity inconsistencies. The minimization of the functional is, however, difficult and computational expansive. The convergence is not always guaranteed.

Although there are a lot of methods for solving the minimization problem, only a few of them are actually used. One of the popular techniques involves a construction of a specialized graph representing the energy function. The graph is constructed such that the minimum cut on the graph also minimizes the energy. Graph cuts provide a fast way to approximate the local minimum.

From the historical point of view, the first global stereo matching method based on graph cuts was stated as an attempt to find a more general 2D formulation of the ordering constraint used by the dynamic programming techniques.

Roy and Cox [124, 123] transformed the multi-camera stereo correspondence problem into a maximum-flow problem. They proposed to assemble all minimum-cost paths of every single epipolar line into a single minimum-cost surface. Thus the surface contains all disparities of the original image.

By analyzing the surface properties, it came clear that the disparities tend to be locally similar. This property is known as a local coherence. In the dynamic programming approaches, the local coherence is enforced by the ordering constraint on each scanline. However, by putting all epipolar lines together we take an advantage of the fact that the local coherence occurs in all directions and not only on the single scanline. As a result, the graph cuts optimization significantly improves the output disparity map.

However, this approach does not provide a general solution. Ishikawa and Geiger [68] showed that Roy and Cox algorithm [124] cannot deal with non-convex disparity maps. They proposed an improved version that guarantees the ordering constraint, uniqueness of a match, and can model occlusions and discontinuities. Today the most renowned method of finding a local minimum using the graph cuts was described by Boykov [18] and Kolmogorov [77] (see Figure 4.5 for the results on the Tsukuba stereo pair).

### 4.4.1 Markov Random Fields

Boykov et al. [16] showed that the correspondence problem can be reformulated in terms of a Markov random field. We should note that the Markov random field with its neighbourhood system is quite suitable for modelling the image grid. The random variables used in the field are in fact the labels representing the disparity values. The problem solved in this type of approach is the maximum a posteriori probability (MAP) estimate. The task is to find an estimation of a configuration \( f \) based on a data \( D \) that maximizes the likelihood function. The likelihood function can be expressed by use of a Bayesian statistics as an energy function \( E(f) \) and the minimization of this energy
function coincides with the maximum a posteriori estimate of \( f \), which can be solved by a minimum cut using a maximum flow algorithm \([16, 18]\).

In \([16]\) the following form of energy is considered:

\[
E(f) = \sum_{\{p,q\} \in N} V_{pq}(f_p, f_q) + \sum_{p \in P} D_p(f_p),
\]

(4.4)

where \( N \) is a set of interacting pairs of pixels (typically adjacent pixels), \( f \) is a labelling function, \( D_p \) is a measure how well the label fits the pixel given the input data, and \( V \) is a metric on the space of labels. \( D_p \) is a nonnegative function, e.g., \( D_p(f_p) = |f_p - I_p| \), where \( I_p \) is the label corrupted by noise and \( f_p \) is, in fact, the unknown true label. The metric \( V \) is given by a piecewise constant model \( V(\alpha, \beta) = K \cdot T(\alpha \neq \beta) \), where \( T = 1 \) when argument is true, and \( T = 0 \) otherwise. \( K \) is a penalty for label discontinuities between adjacent pixels.

### 4.4.2 Minimum Cut

The goal is to find a labelling \( f \) that is piecewise smooth and consistent. Greig et al. \([54]\) showed that certain energy functions can be minimized using the Ford-Fulkerson algorithm \([49]\) for finding the maximum flow in a certain capacitated network.

For this purpose, a graph with vertices forming disparity layers and with two additional vertices called terminals is created. Two types of edges are used: \( n \)-links connect pairs of neighbouring pixels, and \( t \)-links connect pixels with the terminals. The cost of \( n \)-links corresponds to a discontinuity penalty between the pixels \( (V_{pq}) \), while \( t \)-link cost is the penalty for assigning the corresponding label to the pixel \( (D_p) \). Figure 4.6a depicts a sample weighted graph.
Figure 4.6: Stereo matching using the graph cuts. The disparity edges (horizontal edges on the subfigure b) have weights that correspond to the data energy term. The penalty (transverse edges) weights are related to the smoothness energy term. A graph cut separates the graph into two parts, one that connects to the source (S) and one that connects to the sink (T). If you think of the edge weights as pipe capacities, finding the minimum cut is equivalent to finding the maximum flow in the flow network.

The term graph cut refers to a set of edges separating the terminals into two disjoint sets in the induced graph. The graph cut cost is composed of the sum of the costs of $t$-link edges and $n$-link edges. A minimum cut is the cut with the lowest cost of all possible cuts. To establish the disparity of the pixel $p$, the minimum cut must pass through exactly one $t$-link. The construction of the graph representing the correspondence problem is illustrated in Figure 4.6b. Each disparity value has its layer of connected nodes. A surface traversing all the cut $t$-links is the disparity surface that is identical to the minimum cut. The cost of the minimum cut is equal to the value of the maximum flow.

### 4.4.3 Swap and Expansion Move Algorithms

Due to the difficulty of finding a global minimum, many authors have opted for a local minimum. In general, for a given energy $E$, a labelling $f$ is a local minimum if

$$E(f) \leq E(f') \text{ for any } f' \text{ near to } f.$$  \hspace{1cm} (4.5)

For a discrete labelling, the term "near to" refers to a single move of $f$, where only one pixel can change its label in a single iteration. Algorithms using standard moves are, for example, simulated annealing [75], iterated conditional model [9], or variational methods. Simulated annealing is guaranteed to achieve global optimization, but at the cost of computational time.

Recently, two labelling algorithms with large moves – $\alpha$-expansion and $\alpha\beta$-swap algorithms [18] were introduced. It was shown that a strong local minimum can be found.
using entirely a sequence of these moves [78]. This technique is also applicable to a wide variety of non-convex energy functionals. In this way, the hardness of the optimization problem was overcome by an approximate solution.

4.4.4 Parallel Concepts

The most computationally intensive task in graph cut stereo matching algorithms is the computation of the maximum flow/minimum cut. The original Boykov-Kolmogorov algorithm was parallelized by an adaptive bottom-up approach [94]. Other optimizations are also available [66, 129], but the computer vision community interest has moved on to the other platforms. GPU based algorithms, e.g., a push-relabel algorithm [137], are superior as far as the transfer between the main system memory and graphics card does not create a bottleneck.

Graph cuts have found many applications ranging from image reconstruction [54, 69], through a variety of stereo vision algorithms [68, 74, 92], to image segmentation [15] and medical imaging [14, 19].

Morales et al. [102] made a comparison between the graph cuts and belief propagation. The tests run on the Middlebury dataset showed that the belief propagation is much faster. However, when applied to real world images, belief propagation produced unsatisfactory results. Graph cut algorithms usually produce better results at the cost of higher processing time. The most commonly mentioned weakness of graph cuts is their restriction to discrete-valued disparities [92]. The bi-dimensional continuity constraint of the graph cuts may cause an excess of regularization, visible as over smoothing. Window based algorithms with large windows produce the similar effects.

4.5 Belief Propagation

Another way of looking at the stereo matching problem is as a probabilistic stereo model, solved by a Bayesian Belief Propagation algorithm [76, 130]. This method offers some advantages over previous ones. It is suitable for estimating the uncertainties arising from the matching processes, can handle various constraints and is easily expandable for other cues used to reconstruct the depth (e.g., shape from focus, shape from shadow, shape from texture).

Belief propagation algorithms usually consist of three Markov random fields: a smooth disparity field $D$, a spatial line process $L$ modelling discontinuities, and a binary process $O$ for occlusions. The joint posterior probability over $D$, $L$, and $O$ for a given pair of stereo images $I = \{I_L, I_R\}$, is defined as [130]:

$$P(D, L, O | I) = \frac{P(I | D, L, O) P(D, L, O)}{P(I)}.$$  \hspace{1cm} (4.6)
Informally, we can say that the belief propagation solves the problem via a series of local message-passing operations. For example, to compute the outgoing message of the node (see Figure 4.7), the node must combine all incoming messages with its local observation. In graphs with cycles (our case), belief propagation is approximate, but has reasonable theoretical justifications and produces excellent results.

4.5.1 Loopy Belief Propagation

Belief propagation is a popular approximation technique for computing the posterior probability distributions. It is a message passing algorithm originally formulated on trees, and later extended to polytrees. Its alternative, loopy belief propagation can be used even for general graphs.

Belief propagation can be viewed as a form of variable elimination. The creation of a message is equivalent to removing leaf nodes of a subtree. The algorithm breaks the maximization problem for a specific node into independent subproblems corresponding to the subtrees rooted at the neighbours of that node. In each subproblem, all variables except the node are eliminated.

The principle of belief propagation is simple: each node sends messages to its neighbours about its belief regarding its own state (disparity value). Received messages are multiplied by the local potential functions (reflecting the “intuition” as to how probable is the current node’s disparity) in order to update the neighbour’s beliefs. The process is repeated until belief fluctuations fall below a given threshold, or a maximum number of iterations is exceeded. An illustration of how messages are passed in the pixel network is shown in Figure 4.7.

Due to a large amount of calculations, several techniques for optimizing the heavily used matrix-matrix multiplications were proposed (e.g., [63, 41, 28]). An additional speedup of the message propagation can be achieved by observing the distribution statistics [27, 145, 143].

The standard belief propagation can be extended by incorporating additional visual cues such as the image segmentation [130]. It is commonly used to correct potential errors in plane fitting steps of a segmentation based algorithms. The results of plane fitting algorithms are used together with other constraints and fed to a hierarchical BP module. The disparity map is iteratively updated driven by the belief propagation feedback [144, 88].

The disadvantage of belief propagation is that it is not guaranteed to converge. Despite this, in practice, it performs remarkably in many applications. In stereo matching algorithms, it provides an effective inference algorithm for computing the MAP of the posterior distribution.
This method combines the concepts of the local and global stereo matching methods for accurate, pixel-wise matching at relatively low runtime.

The original semi-global matching method is also based on minimizing the cost function. The cost function is optimized similarly to scan line optimization used in the dynamic programming approaches. However, the novel idea of this approach is the computation along several different paths, symmetrically from all directions through the image. Each path carries information about the cost for reaching the pixel with a certain disparity. The disparity with the lowest cost is then chosen. The approach was described in detail by [60].

Numerous segment mismatches may result in significant degradation of the stereo matching performance. Matching errors are frequently caused by a foreground object situated in front of a partly untextured background. In order to fix such situations, several assumptions must be made. One of the most significant assumptions presumes that discontinuities in the disparity map do not occur within untextured areas. The identification of untextured areas might be performed by, e.g., a mean-shift segmentation or k-means. The untextured areas are considered to be surfaces and are evaluated later on.

The preceding ideas were implemented in Hirschmüller’s method [60], which produces sharp object boundaries in the presence of texture objects and falls back to a
planar model on untextured areas. A real-time version of the algorithm was suited for GPU hardware [122].

4.7 SEGMENTATION BASED METHODS

Segmentation based algorithms are currently highly ranked in the Middlebury stereo evaluation [125]. The idea behind the segmentation methods is that the pixels of the same (similar) colour share the same disparity value. For example, the adaptive weight approach assigns all pixels within the window area weights that are inversely proportional to their colour differences. The Graph Cut algorithms solve this by finding the minimum cut in the graph.

The algorithms, presented in the next sections, are closely related to our approaches which will be discussed later on.

4.7.1 Segmented Based Algorithm with Self Adapting Dissimilarity

We have seen that processing pixels independently is not very efficient. In many situations we can achieve significantly better results by utilizing the disparity planes instead of processing pixels individually. For instance, let’s assume a scene composed of a set of non-overlapping planes in a disparity space. Each plane is coincident with at least one homogeneous colour segment. Such situations are suitable for techniques based on a combination of colour segmentation and a plane fitting technique.

At the time of writing this thesis, the algorithm [76] working as described was ranked at the first place in the Middlebury stereo vision page test bed [125]. Because we are going to present similarly based algorithms in the next chapters, we will describe this algorithm in more detail.

The algorithm [76] is based on the segmentation, plane fitting technique and optimization using Belief Propagation. The assumption is that the scene is composed of a set of non-overlapping planes in the disparity space and each plane is coincident with at least one homogeneous colour segment. The stereo matching process is summarised in the steps below:

1. Decompose the image into regions of homogeneous colour (mean-shift colour segmentation is used).

2. Apply local matching algorithm (SAD) and gradient based measure to calculate the dissimilarity measure of all pixels. The dissimilarity measure for the given pixel \((x,y)\) and disparity \(d\) is given by

\[
C(x,y,d) = (1 - \omega) \cdot C_{\text{SAD}}(x,y,d) + \omega \cdot C_{\text{GRAD}}(x,y,d),
\]

(4.7)
where $C_{GRAD}(x, y, d)$ is a gradient based measure and $\omega$ is a weight determined by maximizing the number of reliable correspondences, established by left-to-right and right-to-left cross checking. The gradient measure is defined as

$$C_{GRAD}(x, y, d) = \sum_{u,v\in W_x} \left| \frac{\partial}{\partial x} I_l(u,v) - \frac{\partial}{\partial x} I_r(u+d,v) \right| + \sum_{u,v\in W_y} \left| \frac{\partial}{\partial y} I_l(u,v) - \frac{\partial}{\partial y} I_r(u+d,v) \right|,$$

where $W_x$ is a $3 \times 3$ surrounding window without the most right column, $W_y$ is a surrounding window without the lowest row [76].

3. Estimate the disparity planes for each segment. Repeat this step for all regions grouped to the same disparity plane in order to increase the accuracy.

4. Find the optimal segment to disparity plane assignment using the energy minimization of the functional 4.3. In the algorithm [76], the terms are defined as follows:

$$E_d(d) = \sum_{S_k \in R} \sum_{(x,y) \in S} C(x, y, d) \quad \text{(4.9)}$$

$$E_s(s_i) = \sum_{\forall (s_i,s_j) \in S \mid d(s_i) \neq d(s_j)} \lambda_{disc}(s_i, s_j), \quad \text{(4.10)}$$
where \( S_N \) represents a set of all adjacent segments, \( R \) is a set of all segments, and 
\( \lambda_{\text{disc}}(s_i, s_j) \) is a discontinuity penalty. The functional is approximated using Loopy Belief Propagation (as described in Section 4.5.1).

### 4.7.2 Cooperative Optimization

The calculations above can be accelerated by so-called “cooperative strategy”. The functional is rewritten into a sum of sub-target energy functionals for each segment. The original task is thus decomposed to several simpler sub-targets [139, 11]. For this case, the continuity constraint must be enforced by taking into account several associated sub-targets of adjacent regions. Both methods are based on the same idea, but in contrast to [139], [11] use an alternative planar model with multiple layers.

The stereo matching can be solved by not only looking at individual pixels. The idea of so called “symmetric strategy” [134], is to apply a segmentation on a reference and target image, and then try the matching. Unlike [53], the aggregation strategy is symmetrical and integrates information from both images. Some authors (e.g., [64]) reported that the geodesic distance provides better results in the segmentation process than the standard Euclidean distance, but the validity of such statement is questionable.

The improvements mentioned in this section were used in [142] with an additional pixel classification system. The discrimination of stable and unstable pixels was connected with the correlation confidence, while the occluded regions were detected by mutual the consistency check. The algorithm [142] provides very good results on the artificially created scenes, especially those composed of planar surfaces (e.g., “Venus” from [125]), but for smooth curved surfaces, the performance significantly drops.
In the beginning of the second section of this thesis, presenting our own contributions, we need to define a collection of metrics we will use for describing and comparing the outputs of the algorithms. Generally, there are two approaches to evaluate the results of the stereo matching algorithms. The first one is based on obtaining the scene ground truth data using a laser scanner or similar device. The second evaluation method is based on creating the synthetic images warped according the disparity map. In the latter case, such images are frequently restricted to simple geometric shapes and textures. Furthermore, they usually do not model phenomena such as lens aberrations, noise, different gain or bias. Consequently, the synthetic images differ in many aspects from the reality. This makes them less suitable for algorithm tests. In our tests, we will try to provide the comparisons on both real and synthetic images.

The performance of the stereo correspondence algorithms is nowadays evaluated using the Middlebury dataset [125]. The main dataset contains four image pairs used for the primary evaluation and comparison. Authors have also provided a web service for evaluating and comparing the results online.

5.1 Comparison Criteria

The following quality measures are most frequently used to compare the stereo matching algorithms. The root-mean-square error (RMSE) measures the differences between the values obtained from the algorithm disparity map \( d_C(x, y) \) and the ground truth data \( d_T(x, y) \):

\[
RMSE = \left( \frac{1}{N} \sum_{(x,y)} |d_C(x, y) - d_T(x, y)|^2 \right)^{\frac{1}{2}},
\]

where \( N \) is the number of pixels. The second popular measure is a pixel error rate (PER) defined as

\[
PER = \frac{1}{N} \sum_{(x,y)} e(x, y),
\]

\[
e(x, y) = \begin{cases} 
1 & \text{if } |d_C(x, y) - d_T(x, y)| > \delta_d \\
0 & \text{if } |d_C(x, y) - d_T(x, y)| \leq \delta_d
\end{cases},
\]

where \( \delta_d \) stands for the disparity error tolerance. For most evaluation \( \delta_d = 1.0 \) is used.
5.2 Datasets

In the Middlebury dataset [125] the statistics are computed as PER for the whole image and also for special regions of interest: untextured regions (where the squared horizontal intensity gradient averaged over a square window is below a given threshold), occluded regions, and depth discontinuity regions (containing pixels whose neighbouring disparities differ by more than a given threshold). The RMSE and percentage of bad pixels is calculated for these specific regions in order to evaluate the algorithm performance on occlusions and discontinuities.

The pixel error rate (PER, the percentage of bad pixels) generally gives a better indication of the algorithm performance. Therefore, we will use this measure for the comparisons in our experiments. The root-mean-square error is thereafter used once the pixel error rate is below a few percent and the quality of the output is evaluated based on a sub-pixel fitting.

5.2 Datasets

The test images (Figure 5.2) used in this work are taken from the Middlebury Stereo Evaluation, originally published in [125, 126]. The set "Cones" and "Teddy" consists of complex geometry objects with provided pixel-accurate ground-truth disparity map. The disparities are in range $0.25 \ldots 63.75$ pixels, encoded in ground-truth images using a scale factor of 4.

The "Tsukuba" (head and lamp) is the dataset from University of Tsukuba. The set is available with hand-labelled integer ground-truth disparities. The disparity histogram shows strongly quantized values in the range $5 \ldots 14$ pixels (on the last row of Figure 5.2). The ground truth disparity map is encoded using a scale factor of 16. The "Venus" dataset is a piece-wise planar scene with ground truth disparity data scaled by a factor of 8. In the Chapter 9 we have also used "Map" dataset published in [131], taken with a Point Grey Research trinocular stereo camera. The ground truth disparity map was calculated using a piecewise planar model (Figure 9.1).

When we analysed the published results performed on these images, we have noticed that the untextured areas nowadays do not present a real problem. This is mainly due to the efficiency of the global methods used in modern algorithms. You may also notice that the disparity is established even in half-occluded regions, what is, by the nature of the problem, an impossible task. The guessed missing values are often extrapolated in order to provide full disparity map for applications, e.g., augmented reality or view synthesis.

We were naturally interested, which parts of the dataset images can be generally considered as a difficult and are the source of the most errors. For this purpose, we have created aggregated error disparity maps (Figure 5.1) from all results available in the Middlebury evaluation. The pixels are coloured according the total sum of error of all algorithms in the set. The warmer colours represent areas where the majority (or
5.3 Conclusions

Figure 5.1: The disparity error occurrence calculated as the sum of errors of all algorithms appearing in the Middlebury dataset. These images were created in order to identify the most problematic parts of the images. As you may see, the algorithms are confident in most of the image areas except the borders and, naturally, occluded regions.

all) algorithms failed to determine the correct disparity value. The current algorithms produce very good results on the most of the image area. As you can see, the errors are mainly connected with thin objects (e.g., the lamp cable in “Tsukuba”), or small occluded areas surrounded by huge disparity steps (e.g., the area where two planes overlap in “Venus” or “Cones”).

5.3 Conclusions

By studying the results of various algorithms and based on our experience with the stereo matching algorithms, we came to a conclusion that thanks to the wide acceptance of the Middlebury dataset, the authors often perform the optimization and performance tuning to these specific four images. As a result, the extraordinary result in the comparison table does not necessarily guarantee the overall high success rate in more general and practical situations.

In our following applications, we have often discovered that the best algorithms (according to the evaluation) did not always produce the anticipated results. Therefore, we tried to include to our datasets images that are close to the expected inputs. For us, the practical results and implications are more important than the bare numbers in the comparison tables.
5.3 Conclusions

Cones  Teddy  Tsukuba  Venus

Left Image

Ground truth

Discontinuities

Non-occluded reg.

Disparity hist.

Figure 5.2: Middlebury stereo evaluation dataset (version 2). Test images with ground truth data and disparity histograms.
STEREO MATCHING IN BIOSCIENCE

Before we begin with a detailed description of our algorithms and methods, we would like to give the reader basic guidelines on different types of contributions we made and present the structure of the following text.

Our contributions fall into two main categories: the first group is related to the construction and utilization of the unique scanning system, created especially for the biological researches. This topic coincides with the thesis title and, in this chapter, we will present newly developed methods used for the ongoing researches in the Institute of Environmental Engineering of the Technical University of Ostrava. The second group of contributions consists of three new algorithms for the stereo matching and depth segmentation (described in Chapters 7 - 9). This part of the thesis is much more general and is applicable to a wide range of image processing problems.

The impact of the first group of our contributions is mainly visible in the biology, but we believe that we made the text so appealing, it may also interest the reader who is not very familiar with the bioscience. Since we are aware that the thesis is intended mainly for the computer professionals, we would like to stress that we describe mainly the technical parts. The biological aspects are only briefly mentioned, and only to the extent needed for the proper understanding of the applications.

We regard the three novel algorithms as the main theoretical contributions of this thesis and the bryophyte scanning device as the significant practical result. The algorithms described here were thoroughly reviewed and published in renowned conferences [81, 83, 84, 85] and journals [106, 82]. Finally, we want to point out that our bryophyte scanning device is in use since 2008 (with several later modifications) up to now.

6.1 BRYOPHYTE SCANNING DEVICE

The following sections describe the methods involved in our scanning system, developed especially for the phytologists to study the bryophyte structures. While the subject of this thesis is highly technical, we would also like to emphasize the importance of the biomonitoring and environmental studies performed with our equipment. First of all, we will describe the scanning device and summarize all necessary steps for obtaining the results and then, we will move to the more detailed descriptions of the algorithms we created and used in our applications.
The main objects, studied with the invented scanning device, are bryophytes. Bryophytes are plants of a rather simple structure (see Figure 6.1); they lack roots and conductive tissues of any kind or even structural molecules that would allow establishment of the elements such as lignin [34]. Unlike higher plants, they have no root system; hence, mineral adsorption occurs over their entire surface. Therefore, they are excellent biomonitors for trace elements in air (e.g., heavy metals, sulphuric oxides). This makes them extremely interesting objects of study in our highly polluted environment of Moravian-Silesian region.

The reason for studying these small plants is that the air pollutants affect the structure and function of the plant surfaces. Recent infrequent studies concerning bryophyte canopy structure applied various modern techniques to determine moss layer characteristics. These studies draw mainly on the outcomes of a soil surface research [36]. Surface roughness index has been hereby used as a monitor of quality and condition of the moss layer. Other indices, i.e. the scale of roughness elements and the fractal dimension of the canopy profile have been found to be important as well.

The scope of this work was to create a scanning device capable of capturing the bryophytes surface structure and calculate its characteristics. The goal was to overcome the disadvantages of the other known scanning methods (e.g., [120, 121]) and provide an opportunity to scan other samples (e.g., bark, seeds) as well.

6.2 Former Researches

One of the most significant studies of the bryophyte canopy structure appears in the work of S. Rice [120]. His studies compared the contact probe, LED and 3D laser scanners in the light of efficiency and serviceability. 27 canopies of different growth forms were scanned and outputs evaluated.
6.2 Former Researches

Figure 6.2: Rice’s scanning device in our laboratory. The left image depicts the tripod construction with manual linear motion (with digital sliding gauge on the top) moving the laser line projector. On the left side, you can see a camera (blue coloured) tilted to the sample surface. The right image depicts the detail of the camera and the laser line projector. As you can see, the camera is manually shifted together with the laser.

The results were clear – none of the old methods is convenient for the field research, especially due to the immobility of used equipment, mostly requiring the dislocation of the surveyed moss material into the laboratory. Moreover, the original canopy structure is usually destroyed due to a transfer and excision from its natural environment, but also due to different – much dryer – conditions affecting the moss surface in the laboratory. All of these factors negatively influenced the surface measurements and reliability of the results.

Therefore, Rice et al. [120, 121] proposed a new scanning device that is more suitable and efficient for measuring the structural parameters. Despite his LED scanner is presented in the paper as a portable device, we consider his construction more as laboratory equipment. The construction still requires precise setting and stable working conditions.

Our goal was to create a device without these drawbacks. To enhance our research efforts, we discussed the topic with S. Rice and he was willing to borrow his device for conducting a comparative study. Figure 6.2 shows his scanning device in our laboratory. The tripod is used as a base for the camera and laser LED line projector. The scanning technique is based on the camera capturing the projection of the line appearing on the sample surface. The uneven surface is changing the appearance of the projected line into the curve copying the shape of the surface. The camera, together with the laser projector, is moved above the sample using the manual linear motion. The relative position of the camera is measured by a digital sliding gauge. The line projections, with their relative positions, are further processed to obtain the reconstructed surface.
Figure 6.3: A comparison of the results obtained from the different scanning techniques. The first row shows the scanned bryophyte samples. The second row depicts the scanning results (the depth maps) of the original method proposed by Rice et al. [120] (rotated 90 degrees due to a different coordinate system). And finally, the third row depicts our results (the depth maps from the reconstruction step). It can be clearly seen that our system provides higher resolution and much more details than the Rice’s original device. While the surface roughness can be calculated from both outputs, our system provides much more detailed analysis and supports longitudinal observations.
For the comparison, we have precisely followed the operating instructions given by the author and performed several moss layer measurements with additional experiments. The second row of Figure 6.3 presents some of the scanning results. The subfigures depict the depth maps of three samples taken by Rice’s device and our scanner (described in the next sections). Even though Rice’s outputs provide enough data for calculating the surface roughness, they are unsatisfactory for more advanced structural analysis. The general problem is the diametrically different resolutions in both directions. Due to the construction of the motion system, the device is unable to precisely reproduce the camera positions. Therefore, it is not suitable for repetitive measurements of the same sample, changing over the time (one of the requested feature). Moreover, the most obvious problem of the scanner is the low resolution in the horizontal direction, caused by an inappropriate actuator.

The vertical resolution is limited mainly by the camera and with some additional efforts; it can be increased quite easily. However, this is not the case of the horizontal resolution. The horizontal position is measured using a digital scale with the movement controlled by the manual actuator, which is, in our opinion, one of the main sources of errors. The limitations can also be seen in the scanning technique using a projected laser beam. Repeated plant exposures to a laser radiation influence the photosynthetic reactions and alter the observations.

We have attempted to overcome these limitations and create a device based on a different scanning method – the stereo image reconstruction. The objective was to create a portable device that can be used in the field, needs a minimum time for setting and is able to operate in a variety of environments not affecting the plant growth. Based on the experience with Rice’s device, we have focused on the positioning system of the camera, as we consider this part as a major weakness of his solution. Therefore, we will describe our solution for the camera positioning system in more details.

6.3 HARDWARE CONSTRUCTIONS

Our first scanning device (developed before the cooperation with S. Rice) was composed of two cameras mounted on the tripod. The images were acquired using the IDS Imaging cameras (2240-M-GL, monochromatic, 1280x1024 pixels, 1/2” CCD with lenses PENTAX, f=12 mm, F1.4) firmly mounted in a distance of 32.5 mm (the minimum achievable distance due to the dimensions of the camera enclosures). Images have been taken in normal daylight without any additional light source (Figure 6.4, left). Since the cameras (mounted in a given distance) produced images with relatively huge disparities and difficult configurations for the stereo matching algorithms, we were forced to create a scanning device that is capable of taking images in arbitrary positions above the sample (imitating two cameras separated by a variable distance). Therefore, instead of using two (or more) cameras, we decided to use a single, but moveable camera. This configu-
ration has several advantages: (i) allows arbitrary displacement of captured images, (ii) provides continuous image sequence during the movement (we can obtain an image for each movement step), (iii) simplifies the device settings.

Our currently used scanning device is built on an aluminium chassis, compact and tough enough to be used even in the field. The main construction element is a linear motion mounted on the inner upper centre section of the aluminium cage. The camera (IDS 2240-M-GL) is fastened to the linear motion and is capable of moving over the entire surface of the scanned sample. Kuroda SG2605A-250P linear motion was chosen in order to achieve a position accuracy of 0.02 mm with position reproducibility less than 0.003 mm. Such accuracy is crucial when analysing samples during the long-running experiments. For some experiments we need to place the device above the scanned sample and leave it operating for several days to observe the changes in the plant surface. We have to make hundreds of scans and even the smallest error in the reposition of the camera (repeated many times) can ruin the whole experiment. From our experience, we cannot place calibration marks on the sample, as they are hard to set in a fixed position (especially in a muddy ground). As we want to analyse the surface changes of the plant and not the changes caused by the inaccurate movement of the camera (depicting the samples from slightly different positions), we have to rely on the precision of the camera positioning system.

The image capture is synchronized with the movement and triggered by the embedded computer based on AMD Geode processor. Motor driver, additional sensors and light controlling units are mounted on the chassis as well. The device is designed to be used mainly as a laboratory instrument but can be used in the field as well. Thanks to a low consumption, the unit can operate more than 8 hours on 12 volt 7 Ah battery. The embedded board is connected through our interface board to various sensors to record additional environmental data such as temperature, humidity or the soil pH level.

The above described construction is successfully used and after the four years of service in the Institute of Environmental Engineering, the feedback is very positive. The biologists especially appreciate the easiness of leaving the device turned on for several days and then simply gathering the processed results. The primary goal to create and develop a new generation of bryophyte scanning device was reached. The next sections describe each part of the device in more detail.

6.3.1 Linear Motion Controller

The camera positioning system is an integral hardware part of the whole system. Rice slightly underestimated this part and that is one of the reasons, why his device provides only a limited resolution across the whole sample. We want to avoid these errors (of the positioning components) and make the system as accurate as possible. For our application, the accuracy is even more significant, as we assume a large number of
repeated measurements. Since the scanning system is calibrated only once (usually in the beginning of the measurement) we need to maintain the camera positions for all images as accurate as possible; hence the origin of the strict requirements for the linear motion controller.

At first, we tried to use commercially available linear motion controllers (mainly Microcon CD30M) but after a while, we noticed that after hundreds of measurements, the camera does not return to its original position. The difference was small, but noticeable (approximately three pixels). We tried to identify the source of the problem, but due to the closed source nature of the commercial units, we were unsuccessful. Therefore, we created our own linear motion controller. The parameters of the newly developed unit even surpassed the parameters of the commercially available drivers (designed for small motors). In the following sections, we will briefly describe the microstepping unit, as we consider it as an important contribution. The design of our unit was published in Electrical Review journal [82] and also in [112].

The overall mechanical construction of our scanning device is simple, but robust (see Figure 6.4 on the right). The imaging camera is mounted on a linear motion carriage moved by a stepper motor. The rotary motion of the stepper motor is converted to a linear motion using a lead screw gear drive system. The stepper motor, connected to the gear system, is designed to maintain a fixed step for one turn, thus it is possible to
achieve highly accurate movements. But in practice, there is a lot of applications that require even more sensitive control than the fixed steps, and the proposed scanning system is one of such application.

One possibility how to achieve higher step resolution is by using a special control technology, commonly known as a microstepping. In short, the microstepping can be described as a limitation of a current flow in the motor winding. In this way, the stepper motor can be forced to operate in more positions than it was originally designed to. While theoretically it is possible to control the stepper motor continuously, in reality it is not achievable as the internal friction will divide the continuous motion into the small steps. Therefore, we usually use at most 8 or 16 microsteps between the two fixed mechanical positions. The microstepping is actually a sine cosine driving in which the winding current approximates a sinusoidal AC waveform. This allows stopping and holding a position between two standard step positions and provides smoother operation in low speeds. The basic principles of microstepping can be found in, e.g., [1], and also in our paper [112].

The current, flowing through the motor winding, can be effectively controlled by the pulse width modulation (PWM). The PWM is easy to grasp intuitively, but due to the motor characteristics, is not as easy to implement as one would expect. Generally, there is a distortion or, let’s say a nonlinear gain between the reference and real output voltage in the PWM system. Therefore, a nonlinear feedback is often used to reduce the deviations of the real PWM output stage from the ideal theoretical behaviour [33].

In our article [82], we have presented a pure software solution for performing the PWM corrections. The proposed solution does not require any type of feedback, as it relies only on the calculations performed inside the controller.

6.3.2 Microstepping Unit Driver

The main component of any microstepping unit, and also of ours, is an H-bridge. The H-bridge enables voltage to be applied across a load in either direction. For our experimental microstepping driver, we have chosen IC L6202 as the H-bridge and two phase stepper motor SPA 42/100 with 100 steps per revolution as an actuator (later replaced with SX17-0905).

To measure the positioning precision, we have mounted a laser pointer on a motor axis and measured the microsteps using the beam projections on the wall at a distance of 4 meters. The control unit firmware was set to 8 microsteps using the standard PWM. The expected shift between the two microsteps is $0.45^\circ$, but the experiment showed unexpectedly large differences between the expected values and real measurements. The main cause of this phenomenon is a nonlinear behaviour of the electrical circuit in the stepper motor coils, as described in [146]. The direct use of PWM leads to the irregular current feeding. During continuous motion, these inaccuracies are eliminated by the
6.3 HARDWARE CONSTRUCTIONS

Figure 6.5: Nonlinear behaviour of the electrical circuit in stepper motor coils (6.5a) and the accuracy using the proposed software solution (6.5b). As you can see, the improvements are significant (OPWM vs. PWM).

motor inertia. However, if our intention is to use the motor in the positioning device, we must try to avoid such inaccuracies.

In order to achieve higher positioning precision, we need to improve the precision of the current feeding. Figure 6.5a proves our expectations; the real duty cycle has to be approximated from the curve of real measurements. The formula for the duty cycle calculation must be, therefore, more general than the simple PWM. The appropriate approximation is computed in the control software of the stepper motor driver and stored in the conversion table. The conversion table used for the control can be easily recalculated for any stepper motor. The detailed description of this technique and implementation issues are described in our papers [82, 112].

The results obtained in our experimental configuration are illustrated in Figure 6.5b. The angle deviation of the optimized PWM (OPWM) is less than 0.05 degree from the correct values, compared to the 0.19 degree error of the standard PWM. By taking into account the internal friction of the stepper motor, the results are superior. Thanks to this improvement, we are not longer limited in our construction by the fixed motor steps, but only by a precision of the linear stage. In comparison to the commercial units for small motors, our unit is more stable and does not produce random positioning error, as described in the beginning of this section.
6.4 Stereo 3D Reconstruction

As we have covered the hardware parts of the bryophyte scanning device, we will continue with the software parts. The following main steps (leading to the reconstruction of a sample surface and its statistical analysis) are performed by particular parts of the system: (i) calibration of the optical system (i.e., the pair of cameras), (ii) 3D reconstruction of the sample surface itself, (iii) statistical analysis of the surface data. In the sequel, we will describe the steps in more details.

6.4.1 Camera Calibration

Camera calibration is a necessary step in 3D computer vision in order to gain precise information from the scene images. In the calibration step, the parameters of the optical system are determined, which includes determining the intrinsic parameters of both cameras (focal length, position of the principal point, coefficients of non-linear distortion of the lenses) and the extrinsic parameters of the camera pair (the vector of translations and the vector of rotation angles between the cameras). The calibration is also needed for our system with the moving camera.

The common procedure to calculate the calibration constants is to use a calibration chessboard pattern (Figure 6.6). The chessboard has several unique features: it is a plane, it has a set of uniquely identifiable points, and moreover, the points can be easily extracted and always lie on straight lines, no matter how the camera is positioned.

The calibration is carried out in the following four steps: (i) creating and managing the set of calibration images (pairs of the images of calibration patterns captured by the cameras), (ii) processing the images of calibration patterns (finding the chessboard calibration pattern and the particular calibration points in it), (iii) preliminary estimation of the intrinsic and the extrinsic parameters of the cameras, (iv) final iterative solution of all calibration parameters. Typically, the calibration is done only from time to time and not necessarily at the place of measurement.

For solving the tasks that are included in the Step 2, we used an implementation of well-regarded algorithms. For the initial estimation of the parameters (Step 3), the method proposed by Zhang [149, 150] was used (similar methods may now be regarded as classical; they are also mentioned, e.g., by Heikilla and Silven [59], Heikilla [58], Bouguet [12] and others). Bouguet’s method [12] is also available in Open Source Computer Vision Library (OpenCV).

The process can be summarized as follows. The camera is modelled by a standard pinhole model, where the relationship between a 3D point \( \mathbf{M} = [X, Y, Z, 1]^T \) and its projection \( \mathbf{m} = [u, v, 1]^T \) is given by

\[
\mathbf{s} \mathbf{m} = A[R \ t] \mathbf{M},
\]  

(6.1)
where \( s \) is in an arbitrary scale factor, the extrinsic parameters (rotation and translation) are denoted by \((R, t)\), and the camera intrinsic matrix \( A \) is given by

\[
A = \begin{bmatrix}
\alpha & \gamma & u_0 \\
0 & \beta & v_0 \\
0 & 0 & 1
\end{bmatrix},
\]

where \( \alpha \) and \( \beta \) are the scale factors in image \( u \) and \( v \) axes, \((u_0, v_0)\) are the coordinates of the principal point, and \( \gamma \) is the skewness parameter. The solution of the calibration is found by minimizing the following function

\[
\sum_{i=1}^{n} \sum_{j=1}^{m} \|m_{ij} - \tilde{m}(A, R_{ij}, t_{ij}, M_j)\|^2,
\]

where \( \tilde{m}(A, R_{ij}, t_{ij}, M_j) \) is the projection of the point \( M_j \) in the image plane \( i \). The sum of the squares of the distances between the theoretical and the real projections of the calibration points is minimized by the Levenberg-Marquardt method.

If the optical system has been successfully calibrated, the surface of the observed sample may be reconstructed, which is done in the following four steps: (i) capturing a pair of images of the sample, (ii) performing the correction of geometrical distortion in the images, (iii) rectifying the images, (iv) stereo matching, (v) reconstructing the sample surface.

While all these steps are important, the main interest of this thesis is the stereo matching problem. For this reason, we will provide only a very brief description of remaining steps, since they are very well documented (e.g., [35, 44, 87]) and the selected algorithms are available as parts of OpenCV and CVIPTools libraries.

Distortion correction removes the geometrical distortion of the camera lenses. The polynomial distortion model with the polynomial of the sixth degree was used in the application. The distortion coefficients were determined during the calibration.

The rectification step is necessary since it is impossible to guarantee that the optical axes are parallel in reality, despite the fact that the camera is moving on the linear motion only in one direction. The rectification algorithm takes the original projection matrices of the cameras (determined during the calibration) and computes two new projection matrices of fictitious cameras whose optical axes are parallel and the projection planes are coplanar. After the rectification, the corresponding points in both images have the same \( y \)-coordinate.

It may seem that our system with the linear stage considerably differs from the classical stereo camera systems. Instead of two separate cameras, we use only one, and move its position. But as we are (at this stage of development) utilizing only two frame stereo matching algorithms, we emulate the two stereo cameras by a movement of the one. The camera is shifted to a position of the virtual left camera and then, after taking an image, is moved to a position of the virtual right camera. In addition, we take extra
images between these two terminal positions. These additional images are prepared for the future development of an algorithm based on the motion tracking / optical flow.

The situation for finding the correspondence between the pixels of the captured images is quite difficult, especially in the given context. The structure of the samples is quite irregular and, in a sense, similar to noise (see Figure 6.6). We have tested several well known algorithms [32, 60, 61, 74, 76, 79, 80, 109] to find the best one for our specific needs. A preview of the obtained disparity maps from these algorithms can be found in Figure 6.7. The algorithms were executed on the same input pair and their parameters were tuned to achieve best results.

We have performed several other tests of these algorithms in the reconstruction process of the bryophyte structure, but the results were not satisfactory for the biologists’ purposes. Therefore, we have applied a combination of several depth estimation techniques to perform better than the methods based only on the general-purpose two frame stereo matching algorithms.

### 6.4.2 Additional Depth Estimators

This section describes methods that were used in our scanning device to acquire the depth map and fill the gaps caused by occlusions. As we wanted to provide the best
Figure 6.7: A preview of the disparity maps obtained with the different algorithmic approaches on the sample image pair depicting the moss structure (*Sphagnum girgensohnii*). The algorithms were executed on the same input pair and their parameters were chosen to achieve best results. The implementation of the algorithms (a, b, c, f, i) was taken from the OpenCV, the algorithm (e) is available in OpenVis3D library and algorithms (d, g, h) are authors implementations according the corresponding papers. Good results were achieved using the algorithms [60], [61], [109] (denoted as b, c, e).
possible results, we combined several other approaches for the depth estimation. For clarity, the algorithms used in this subsection are not of our authorship and since they are not directly related to the thesis topic, we will describe only the main ideas and refer the reader to the specific papers for more details.

In our application, we perform the following steps: (i) calculate the depth map using the standard two frame stereo matching algorithm (will be specified later), and then improve and fill the depth map with values obtained by (ii) depth from focus and (iii) depth from lighting cues. The following paragraphs will briefly explain them.

We take advantage of some known properties of the scanned samples and, moreover, use some hardware tweaks. As we can modify our equipment to meet the biologists’ specific needs (samples have small dimensions and complex surface structures), we have equipped the camera based scanning system with additional lighting system and motorized lens.

As a starting point, we use the disparity map obtained from the Ogale and Aloimonos algorithm [109]. This algorithm was chosen for its stable matching of small areas and edge-preserving characteristics.

The depth map obtained from the stereo matching algorithm is further processed (by heuristic means) in order to increase the resolution and fill the gaps caused by the missing data. To achieve this, we use the following two monocular cues – depth from the focus and lighting cues.

The depth estimation based on lighting cues was already presented in several papers [90, 97, 114]. The general idea of these approaches is to calculate the depth from multiple images taken from the different angles or with varying lighting conditions. The proposed methods were usually stand-alone algorithms requiring either calibrated camera or controlled light source. More simplified set-up was proposed in [90], using a light source mounted on a linear stage.

For our application, we have used the method described in [90]. The configuration was, however, simplified to only two light sources of the same type (slightly different parameters of the light sources must be assumed), placed in different distances from the scanned sample. Assuming that we already have a depth map (i.e., the disparity map from the stereo matching algorithm) we can find the correlation of the depth and the light intensity.

According to the inverse square law, the measured luminous flux density from a point light source is inversely proportional to the square of the distance from the source. The intensity of light \( I \) at distance \( r \) is \( I = \frac{P}{4\pi r^2} \), \( I \propto \frac{1}{r^2} \), here \( P \) is the total radiated power from the light source. The analysed surfaces are approximated by Lambertian reflectors.

The points that show high level of certainty in the stereo matching step (the stereo matching algorithm must provide the level of certainty for each pixel) are used as the reference points. A look-up table, based on the inverse square law, mapping the spatial depth to the intensity is calculated and used for the depth estimation.
6.4 Stereo 3D Reconstruction

![Figure 6.8: Disparity map enhancement based on additional depth estimators. Figures from left to right show the original image, output from the stereo matching algorithm and output after enhancement. As you can see on the final image, the gaps and clearly incorrect disparity values are corrected, making the final depth map visibly smoother.](image)

The second implemented monocular cue was based on the focus measurement. The depth from focus approaches were quite popular in the past, especially for the real time depth sensors used in robotics, e.g., [24, 107, 140, 141]. The key to determine the depth from focus is the relationship between the focused and defocused images. Blurring can be represented in the frequency domain as a low-pass filter, so the focus measure can be estimated by means of frequency analysis. The discrete Laplacian is very popular as the focus operator. For each differently focused image, the depth is estimated as the maximal output of the focus operator. In our application, we adapted [107] approach. Since the images are taken from a short distance, the lens focus depth is optimal to capture the scenes with relatively high details.

The combined (final) depth map is calculated as the weighted average from the values given by the stereo reconstruction process and the two monocular estimators. The weights were set empirically according the previous experiments. An illustration of the enhancement achieved with the above mentioned methods can be seen in Figure 6.8.

In order to compare the scanning results more rigorously, we have performed several tests with the stereo matching algorithms using the images of the bryophytes. The results are summarized in the next section.
6.4 Stereo 3D Reconstruction

6.4.3 Comparison to Other Algorithms

The comparison of the different algorithms was performed using a benchmark data with a known set of matching points. The motivation was to find the most suitable stereo matching algorithms for the stereo images depicting the bryophyte structures. Since we do not have a ground-truth data of the scanned samples and attempts to acquire them automatically in other way than the stereo reconstruction method failed, we were forced to obtain them manually. We asked collaborating biologists to match the points manually using the software (developed in the past) for the 3D reconstruction of the manually entered points. They have selected 24 image pairs of 6 most commonly studied species. The dataset contains at least 100 matched points for each image pair. The points were either a very well recognizable features or points of special interests for their studies.

As we are aware that the dataset is non-standard and was specially compiled for our evaluation, we would like to emphasise that the results presented in this section should be considered merely informative and statistically less valuable (than the standardized datasets, e.g. [125]).

The benchmark was performed with nine stereo matching algorithms, representing the current different approaches (as described in Chapter 4). The selection of the algorithms was influenced mainly by the availability of the authors’ implementations. Regrettably, we were unable to cover the recently added algorithms of the Middlebury stereo evaluation due to the inability to obtain the authors implementations or ambiguous descriptions, which make impossible to create our own implementations. However, we consider our selection as a representative sample of the renowned algorithms.

The results of the tests (performed on the dataset of manually matched points) are given in Table 6.1. The tests were performed on the following algorithms: Block-Matching (BM) [79], Semi-Global Block-Matching (SGBM) [60], Consistent Semi-Global Matching (CSGM) [61], Dynamic programming (DP)[32], Shape Stereo Matching (SSM) [109], Graph Cuts [74], Belief Propagation [76], SSD and Variational Stereo Correspondence [80]. The belief propagation has usually good accuracy for stereo correspondence but suffers from large run-time memory cost. Due to this limitation, BP algorithm was run only on half-sized images, reducing its quality. The results of the combination of the stereo matching algorithm SSM [109] and the two additional monocular cues are given in the last line (denoted as SSMc2).

From the 24 samples, the combined approach provided the better results on 12 samples (the average correct ratio was 80%), while the second best CSGM algorithm was better only on 6 samples (the average correct ratio was 76%). As a correct pixel we consider the one with the disparity value in a given range from the (manually entered) reference. The maximum allowable disparity deviation from the reference value was set to 2. The third and fourth best algorithms were SGBM and SSM.
The number of samples is insufficient for the correct statistical evaluation. We are aware of this, but using other source of reference data (e.g. printed 3D models with the known parameters) we regard as inappropriate since we tried (from the beginning) to optimize the device for scanning the bryophytes. Besides, the method is a combination of multiple approaches, using additional information not available to the classical two frame stereo matching algorithms and comparing them would be (from the methodological point of view) improper.

In our opinion, much more important than the raw performance is the performance in the intended applications. We included these results in the following Section 6.6, but before that, we have to describe the parameters used in the bryophyte studies as they are used in these tests.

### 6.5 Surface Roughness

The reconstructed 3D surface is usually not the sort of output the phytologists could use for a bryophyte structure comparison. Surface roughness (a measure of the texture of a surface) is a more convenient parameter used to describe the bryophyte surface. The roughness measures the small-scale variations in the height of a physical surface of the canopies.

In order to calculate the surface specific parameters, we have to minimize the impact of the subsoil segmentation. Therefore, we have performed a regression using the polynomial expression to interpolate the subsoil surface. This step produces the surface that represents the average canopy level. The difference from the reconstructed z-coordinates (the height of the plant in the specific place) and the fitting surface is then statistically evaluated.

The commonly used measure of the statistical dispersion is the standard deviation or the variance. For the bryophyte structure, phytologists prefer the \( L_r \) parameter defined by Rice \[120\] as the square root of the maximum semivariance increase by a factor of two. The semivariance is defined as

\[
\hat{\gamma}(h) = \frac{1}{2n(h)} \sum_{i=1}^{n(h)} (z(x_i + h) - z(x_i))^2,
\]

where \( z \) is a value of the plant height at a particular location, \( h \) is the distance between ordered data, and \( n(h) \) is the number of paired data at a distance of \( h \). The semivariance is computed for the range of values \( h \) (the range is usually chosen based on the size of scanned sample). For a more detailed description of the surface roughness calculation, see \[120\]. Figure 6.9 shows the changing roughness parameter during the experimental study of the structural changes caused by the adaptation to low environmental humidity.

To conclude, we are convinced that the real bryophyte surface is much more complex and no single parameter can describe the complexity of its surface. To provide more
6.5 Surface Roughness

<table>
<thead>
<tr>
<th>Sample (Ref. c.)</th>
<th>Polytrichastrum formosum</th>
<th>Sphagnum girgensohnii</th>
<th>Dicranum scoparium</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM [70]</td>
<td>65.8 (151)</td>
<td>59.1 (129)</td>
<td>61.2 (184)</td>
</tr>
<tr>
<td>SGBM [60]</td>
<td>71.4 (131)</td>
<td>68.3 (140)</td>
<td>72.9 (155)</td>
</tr>
<tr>
<td>CSGM [61]</td>
<td>78.0 (112)</td>
<td>76.8 (145)</td>
<td>79.6 (109)</td>
</tr>
<tr>
<td>DP [32]</td>
<td>68.4 (173)</td>
<td>64.6 (142)</td>
<td>65.6 (100)</td>
</tr>
<tr>
<td>SSM [109]</td>
<td>72.0 (142)</td>
<td>73.9 (155)</td>
<td>71.0 (100)</td>
</tr>
<tr>
<td>BP [76]</td>
<td>72.4 (155)</td>
<td>66.5 (155)</td>
<td>62.5 (100)</td>
</tr>
<tr>
<td>SDS</td>
<td>61.0 (120)</td>
<td>66.4 (100)</td>
<td>62.3 (100)</td>
</tr>
<tr>
<td>VAR [80]</td>
<td>63.4 (112)</td>
<td>59.4 (100)</td>
<td>63.6 (100)</td>
</tr>
<tr>
<td><strong>SSM2</strong></td>
<td>77.9 (100)</td>
<td>80.0 (100)</td>
<td>82.3 (100)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sample (Ref. c.)</th>
<th>Bazzania trilobata</th>
<th>Plagiothecium undulatum</th>
<th>Polytrichastrum commune</th>
</tr>
</thead>
<tbody>
<tr>
<td>BM [70]</td>
<td>59.2 (100)</td>
<td>60.9 (100)</td>
<td>60.9 (100)</td>
</tr>
<tr>
<td>SGBM [60]</td>
<td>69.7 (101)</td>
<td>67.1 (100)</td>
<td>67.1 (100)</td>
</tr>
<tr>
<td>CSGM [61]</td>
<td>84.9 (100)</td>
<td>73.6 (100)</td>
<td>73.6 (100)</td>
</tr>
<tr>
<td>DP [32]</td>
<td>63.6 (107)</td>
<td>64.5 (100)</td>
<td>64.5 (100)</td>
</tr>
<tr>
<td>SSM [109]</td>
<td>72.0 (107)</td>
<td>67.1 (100)</td>
<td>67.1 (100)</td>
</tr>
<tr>
<td>GC [74]</td>
<td>68.1 (100)</td>
<td>64.8 (100)</td>
<td>64.8 (100)</td>
</tr>
<tr>
<td>BP [76]</td>
<td>72.4 (107)</td>
<td>65.7 (100)</td>
<td>65.7 (100)</td>
</tr>
<tr>
<td>SDS</td>
<td>60.4 (100)</td>
<td>63.8 (100)</td>
<td>63.8 (100)</td>
</tr>
<tr>
<td>VAR [80]</td>
<td>63.4 (100)</td>
<td>65.0 (100)</td>
<td>65.0 (100)</td>
</tr>
<tr>
<td><strong>SSM2</strong></td>
<td>79.6 (100)</td>
<td>82.3 (100)</td>
<td>82.3 (100)</td>
</tr>
</tbody>
</table>

Table 6.1: The performance of different algorithms evaluated on the set of reference points manually entered by the biologists. The dataset is composed of 24 image pairs of 6 different species (labelled as bxxy, where x stands for the species identification and y for the sample number, the total number of reference points is given in brackets below). For each image pair we have established at least 100 manually entered reference points. Maximum allowable disparity deviation from the given reference point is 2 pixels. Pixels exceeding this range are marked as incorrect. The results are in percentage of correct matches. The results of the approach combining the SSM algorithm [109] and two monocular cues are given in the last line (SSM2). The best results are highlighted in bold for each column. The method combining the different approaches provides better results, but we have to stress that the tests were run on the limited data set (as described in the text) and moreover, the reference points were purposely chosen by the biologists. Therefore, the reader should consider the results with the awareness and consider them merely as informative.
information about the surface characteristics we shall use other measures, e.g., the maximum height of the profile, the average distance between the highest peak and lowest valley in each sampling length, etc. As a promising alternative to the surface roughness we see a description based on the bryophyte clusters analysis.

6.5.1 Clustering Algorithms with the Stereo Matching Constraint

The main purpose of this short section is to put into the context following chapters that describe the specific stereo matching and clustering algorithms.

The cluster analysis is frequently used to describe the spatial and temporal characteristics of the communities of organisms in heterogeneous environments. The bryophytes form a group of living plants – a canopy and thus, we can consider the canopies as the natural subjects to the cluster analysis. The canopy structure influences the exchange of the matter and energy and can be used as an important bioindicator of air pollution in the given area.

In the following chapters 7 and 9 we will describe the applications of the specific, newly developed techniques for the bryophyte analysis. The algorithm described in Chapter 9 provides localization and clustering technique for analysing the natural clusters of mosses formed on a heterogeneous substratum. The Chapter 7 presents an algorithm dividing the samples into two disjoint regions, representing the upper parts and ground level of the plant. Thanks to the long-term cooperation with the Institute of Environmental Engineering, we have a huge collection of long term data that can be analysed. Our colleagues are currently focusing their research on correlating the cluster properties and the changes in key environmental parameters. But what is already evident from the various researches is that we can consider the roughness index (which is already scientifically accepted method) as one the most important surface parameter and thus important bioindicator.

6.6 Scanning Results in Biological Studies

The analysis of the results obtained by using our device shows the major advantages of the newly developed construction. This was shown in the study [105] comparing the Rice’s scanning apparatus and our new construction. The results were clear - while the Rice’s device was able to distinguish only a half of the processed samples in the study, our device was able to distinguish them all. We consider this practical result as the most important and an indisputable proof that our device works correctly and is useful for the biological observations.

Here is a short digest from the study [105] comparing the Rice’s and our device. The following text may be difficult to understand without the appropriate knowledge of the
Figure 6.9: Sample images and disparity maps taken from the experimental setup to measure the *Polytrichum formosum* surface parameters during the drying up process (the images illustrate the changes). The graphs above depict the temperature, humidity and one of the observed roughness parameter calculated from the reconstructed images (the solid red curve depicts the values from the sensor located inside the canopy, while the green one are values from the combined humidity and temperature sensor located outside the sample).
bryophytic forms, but we consider it important to include this information (readers who are not interested may safely skip to the next section).

The scanning devices were compared in a study of six bryophyte species surface structure (Bazzania trilobata, Dicranum scoparium, Plagiojum undulatum, Polytrichastrum formosum, Polytrichum commune and Sphagnum girgensohnii), in laboratory and in situ.

The same samples were scanned by both devices, and the outputs were used for a calculation of the surface characteristics. The results of the surface analysis were then analysed with the following findings: (i) the data obtained from the Rice scanner did not yield significant differences on three of six species, while our device was able to significantly distinguish all of them; (ii) both methods showed distinctively different results on one specimens, caused probably by the manipulation with the samples; (iii) the resolution of our scanner is currently six times higher than the device sent by Rice.

In more detail, the indices calculated in eight specimen per each species (four laboratory and four in field measurements using our scanning device) were found to significantly distinguish the specimens in dependence on species kind; one-way analysis of variance showed high significance in our approach ($p = 0.0001$) when data were pooled discounting whether derived from laboratory or from field measurements. In contrast, data obtained (from the same samples) using Rice approach did not show statistically significant differences ($p = 0.2129$). Separate laboratory measurements using our device gave not that significant outcomes ($p = 0.0935$), for there were found distinctively different indices of Dicranum scoparium specimens in laboratory and in field caused probably by a disturbance of their canopies during the transfer and storage in the laboratory. This is supported by the fact that independent two-sample t-test showed significant difference between laboratory and field measurements outcomes only in case of this one species ($p = 0.0069$). The similar observation of the probably destroyed Dicranum scoparium specimens was also made using Rice scanner. The complete results of this study can be found in [105].

The results of the comparison clearly showed that our device is suitable to be utilized even under in situ conditions which are, according to the outcomes of the above mentioned study, considered to be much more convenient way to study bryophyte canopy structure. Figure 6.10 shows the structure of bryophyte canopies and the appropriate visualizations of the surface as obtained using our scanning device.

6.7 Conclusions

This chapter presented a novel scanning device designed for analysing the biological samples of small dimensions, mostly the bryophytes. We have created the special hardware construction, as well as developed the necessary software for the analysis. In the sections devoted to the hardware construction, we have described the overall scanner construction and our achievement in inventing an improved motor controller using the
6.7 Conclusions

Figure 6.10: Examples of the reconstructed bryophyte surfaces (Bazzania trilobata and Polytrichum commune) as presented in [106].

optimized pulse wide modulation (published in [112, 82]). The motor controller was important, since the improper motion control created significant errors in previous constructions.

From the software point of view, we have covered all main steps of the 3D reconstruction and analysis of the reconstructed points (the reconstruction process was published in detail in [84, 86]). The traditional algorithms used for the stereo matching of the images of the bryophyte samples were compared. The main motivation for developing new methods was that the results of the reconstruction process, based only on the traditional stereo matching methods, were not very satisfactory, usually contained a lot of noise without any further value for the bryophyte analysis. Meanwhile, the combination of several approaches with additional monocular cues produced better outputs that are more useful for further biological investigations. Our device was compared with other available devices and was found to be the most suitable among all [105] (the comparison was conducted by the cooperating biologists).

The number of studies performed with our device together with the number of analysed specimens and obtained measurements for the statistical analysis is unprecedented.
and such approach is, so far, the most successfully used in the field. The capabilities and operational ease of our device surpass the past constructions. The first study, with the results obtained using our bryophyte scanning device, appeared in 2008. Since then it has been used in various consequent experiments [85, 91, 106] up to now, 2013 [104].

In our opinion, the proposed solution is a significant improvement to the scanning devices used for the bryophyte studies and considerably contributes to the monitoring of the damages caused by the pollutants released by human activity.
Up to now, we have mostly discussed the algorithms producing the dense disparity maps. In this chapter, however, we will present an algorithm operating only on two disparity planes. The method, we have invented, separates the planar objects of arbitrary shapes lying in a certain height above a background (base plane) using the data from the stereo image pair. (More exactly, the objects and the background can lie on two distinct planes in a general mutual position.) As we are working with the gray-scale images, the brightness of particular objects in one image may be different and may vary along each object. The brightness of the base plane may vary too.

Our new method is based on the level set function. We had to modify the original functional and add completely new matching terms. The level sets may seem too complex for the task which, at first glance, seems quite simple and obvious. However, as you will see, the reality of our applications is much more challenging – the input images are difficult to process and simpler methods are unable to provide satisfactory results.

The general motivation for this approach is to exploit strong a priori known properties of the scene. By doing this, we believe it is possible to achieve better results, especially for the real-life images that are often difficult for the general-purpose algorithms. Therefore, our global minimization model is based on the new functional which differs from the standard sets. The newly introduced model, on the other hand, is also a strong limitation of this method; the method can be used only in specific cases.

In our applications, we used the following method for segmenting the layers of the bryophyte canopies and recently, we have also applied this algorithm for detecting the cars on parking lots for the purpose of detecting the occupancy of particular places. The results of both applications are very satisfactory and encouraging. Besides the practical usage, the algorithm was published in Acivs 2012 [83].

The following sections describe the algorithm and its practical use in more detail.

7.1 BIMODAL DEPTH SEGMENTATION AS A MINIMISATION PROBLEM

Our bimodal depth segmentation problem is formulated as a minimisation problem. We propose a new functional based on evaluating the mismatches between the images, which contrasts with the common approaches that evaluate matches (i.e., small brightness difference or big correlation between the corresponding points). The minimisation is carried out by making use of the Euler-Lagrange equation and the level set function.
In its essence, the algorithm was inspired by the algorithm proposed by Chan and Vese [21, 22].

The level sets are generally known to the computer vision community mainly as the image segmentation methods. The basic approach was originally introduced by Osher and Sethian [115], and since then it is being extensively developed and improved in many ways [23, 98, 65]. The main advantage of the level sets is the ability to extract objects of various shapes and surfaces without the need to parametrize them.

In the former studies, the level sets already showed the ability to solve the spatial problems such as the 3D shape extraction. A method for recovering the projections of 3D planar curves using the level sets was proposed in [38, 39]. Later, this approach was extended to solve the stereo matching problem for an arbitrary number of images [46], where a proposed functional is based on evaluating the brightness correlation between the corresponding points. Latest methods use regularization terms (for discontinuity preserving) based on the Mumford-Shah functional combined with an additional functional for occlusion handling [7].

7.2 REVIEW OF CHAN AND VESE’S METHOD

This section introduces the original Chan and Vese algorithm, which was the inspiration for the proposed method. Rather than describing the details [21, 22], we will focus only on the main ideas.

Let \( u(x) \) be the brightness function of the input image that is defined over a two-dimensional area, denoted by \( \Omega \). It is being assumed that the image consists of objects and a background. All the objects as well as the background have nearly constant brightness; the quantities \( c_o \) and \( c_b \) denote the average brightness of the objects and the background, respectively. Let \( C \) stand for the collection of closed curves that separate the objects from the background. The original Chan and Vese algorithm is based on minimizing the fitting energy functional that is defined by the equation

\[
F(c_o, c_b, C) = \mu \cdot \text{Length}(C) + \nu \cdot \text{Area}(\text{inside}(C)) + \\
+ \lambda_o \int_{\text{inside}(C)} (u(x) - c_o)^2 \, dx + \lambda_b \int_{\text{outside}(C)} (u(x) - c_b)^2 \, dx,
\]

(7.1)

where \( \mu \geq 0, \nu \geq 0, \) and \( \lambda_o, \lambda_b \geq 0 \) are suitably chosen parameters adapting the functional to a particular class of images.

In order to minimise the functional from Eq. (7.1), a function \( \phi(x), x \in \Omega, \) referred to as a level set function, is introduced. By the level set function, the image is partitioned into a set of regions. Inside the objects, the value of the level set function is \( \phi(x) > 0; \)
7.2 Review of Chan and Vese’s Method

\( \phi(x) < 0 \) holds outside the objects. The object contours are found at points where \( \phi(x) = 0 \). Using the Heaviside function

\[
H(z) = \begin{cases} 
1 & \text{if } z \geq 0 \\
0 & \text{if } z < 0 
\end{cases},
\]

(7.2)

the functional from Eq. (7.1) can be rewritten in the following way (we omit the parameter \( x \) of \( u \) and \( \phi \))

\[
\begin{align*}
F(c_o, c_b, \phi) &= \mu \int_{\Omega} |\nabla H(\phi)| \, dx + \nu \int_{\Omega} H(\phi) \, dx + \\
&\quad + \lambda_o \int_{\Omega} (u - c_o)^2 H(\phi) \, dx + \lambda_b \int_{\Omega} (u - c_b)^2 (1 - H(\phi)) \, dx.
\end{align*}
\]

(7.3)

Keeping the level set function \( \phi \) fixed and minimising \( F(c_o, c_b, \phi) \) with respect to \( c_o, c_b \), we can find the following expressions

\[
\begin{align*}
c_o(\phi) &= \frac{\int_{\Omega} u H(\phi) \, dx}{\int_{\Omega} H(\phi) \, dx}, \\
c_b(\phi) &= \frac{\int_{\Omega} u (1 - H(\phi)) \, dx}{\int_{\Omega} (1 - H(\phi)) \, dx}
\end{align*}
\]

(7.4)

where \( c_o(\phi) \) and \( c_b(\phi) \) stand for the brightness of the objects and the brightness of the background, respectively. Keeping \( c_o, c_b \) fixed and minimizing \( F(c_o, c_b, \phi) \) with respect to \( \phi \), we obtain the Euler-Lagrange equation in the form of

\[
\delta(\phi) \left[ \mu \operatorname{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) - v - \lambda_o (u - c_o)^2 + \lambda_b (u - c_b)^2 \right] = 0.
\]

(7.5)

For practical computation, a regularized version of \( H \) (denoted by \( H_\varepsilon \)) and its derivative are introduced as follows

\[
H_\varepsilon(z) = \frac{1}{2} \left( 1 + \frac{2}{\pi} \arctan \left( \frac{z}{\varepsilon} \right) \right), \\
\delta_\varepsilon(z) = H_\varepsilon'(z) = \frac{1}{\pi} \cdot \frac{\varepsilon}{\varepsilon^2 + z^2},
\]

(7.6)

where \( \varepsilon \) is a suitably chosen constant.

The Euler-Lagrange equation can be solved iteratively by introducing the parametrisation \( \phi(t, x) \) in time, by considering the boundary conditions, and by iteratively updating the values of \( \phi(t, x) \). The following is computed

\[
\begin{align*}
\frac{\partial \phi}{\partial t} &= \delta_\varepsilon(\phi) \left[ \mu \operatorname{div} \left( \frac{\nabla \phi}{|\nabla \phi|} \right) - v - \lambda_o (u - c_o)^2 + \lambda_b (u - c_b)^2 \right] \text{ in } \Omega, \\
\phi(0, x) &= \phi_0(x) \text{ in } \Omega, \\
\delta_\varepsilon(\phi) \frac{\partial \phi}{\partial n} &= 0 \text{ on } \partial \Omega.
\end{align*}
\]

(7.7)
where \( \mathbf{n} \) denotes the exterior normal to the boundary \( \partial \Omega \) of \( \Omega \), and \( \frac{\partial \phi}{\partial \mathbf{n}} \) is the normal derivative of \( \phi \) at the object boundaries. The initial value \( \phi_0(x) \) of the level set function is often determined from an initial estimation of the object boundaries. The values of \( \phi_0 \) are computed as the distance from the initial boundary curves.

We point out that an alternative method was presented in [108] converting the problem of minimising the functional from Eq. (7.3) into the problem of convex minimisation. General acquaintance of the original method was a motivation to use the original approach as a basis since we believe that the essential thing that is always important is to formulate the main functional first. The techniques how the minimisation can be carried out effectively are certainly important too, but to a lesser degree.

7.3 Description of the Method

We suppose that we deal with the following scenes (Figure 7.1). The scene is composed of a base plane and a set of objects. The objects have the form of pieces of another plane lying in a certain height above the base plane; this second plane is not necessarily parallel to the base plane. The shape of objects is arbitrary. For the points of the base plane and the objects, respectively, we will also use the terms bottom points and top points. We consider the grey-scale images and scenes. The brightness of particular objects may be different and need not be constant. The restriction is that all the objects should be either darker or brighter than is the background.

We suppose that we have two images of a scene that are obtained by two pinhole cameras at different positions. To distinguish between the cameras and images, we will use the term the left and the right camera or image. Due to the pinhole model, both the images are connected by a projective transformation. We suppose that this transformation is known. The goal is to segment the images, i.e., to find the whole areas of particular objects, i.e., to classify all pixels as the pixels containing the projections of either the top points or bottom points.
Figure 7.2: Projective transforms between images (without occlusions): If a top point is seen at $x_L$ in the left image, then it should also be seen at $\pi_{LTR}(x_L)$ in the right image; otherwise, if a bottom point is seen at $x_L$, it should be seen at $\pi_{LBR}(x_L)$. The point that can be seen at the position of $\pi_{LTR}(x_L)$ in the right image need not necessarily be only the top point that is seen at $x_L$; at the same place a bottom point can also be seen whose projection in the left image is at $\pi_{LTB}(x_L)$ (see the text). Similarly, the point that can be seen at $\pi_{LBR}(x_L)$ in the right image need not be the bottom point only that is seen at $x_L$; it can also be a top point that, in the left image, is seen at $\pi_{LBT}(x_L)$.

Firstly, we focus only on the geometric transformations without considering any occlusions. Say that we see a certain point whose position is described by $x_L$ in the left image (Figure 7.2). It can either be the image of a bottom point or the image of a top point. In the case of the top point, the position of the corresponding point in the right image is $\pi_{LTR}(x_L)$, where $\pi_{LTR}(\cdot)$ is the projective transformation; the subscript LTR stands for the transformation for the top points from the left to the right image (the other subscripts introduced in this paragraph are created according to the similar rule). In the case that $x_L$ is a projection of a bottom point, its corresponding projection in the right image should be at $\pi_{LBR}(x_L)$. We can also think about a point $x_R$ in the right image and introduce two possible positions of the corresponding points in the left image, namely $\pi_{RTL}(x_R)$ and $\pi_{RBL}(x_R)$ for the top and bottom points, respectively. From the geometrical point of view, the point that can be seen at the position of $\pi_{LTR}(x_L)$ in the right image need not necessarily be only the top point that is seen at $x_L$; at the same place a bottom point can also be seen whose position in the left image is $\pi_{RBL}(\pi_{LTR}(x_L)) \equiv \pi_{LTB}(x_L)$. Similarly, the point that can be seen at $\pi_{LBR}(x_L)$ in the right image need not be only the bottom point that is seen at $x_L$; it can also be a top point that, in the left image, is seen at $\pi_{RTL}(\pi_{LBR}(x_L)) \equiv \pi_{LBT}(x_L)$.

In the method we propose, the level set function $\phi$ distinguishes between the objects and the base plane, i.e., between the areas containing the projections of either the top
or the bottom points. We evaluate the level set function over the left image, i.e., we will often write \( \phi(x_L) \). The objects can be expected whenever \( \phi(x_L) > 0 \), whereas \( \phi(x_L) < 0 \) indicates that the left camera sees the base plane.

In accordance with the original Chan and Vese functional in Eq. (7.3), we use the term requiring the boundary between the objects and the background to be short (the first term in the following expression) and the term requiring the area of objects to be small (the second term), i.e., we use

\[
\mu \int_{\Omega} |\nabla H(\phi(x_L))| \, dx + \nu \int_{\Omega} H(\phi(x_L)) \, dx .
\] (7.8)

Instead of the terms in Eq. (7.3) containing the intensities (the third and fourth term), we use another term expressing the correspondence between the left and the right image. The new term is of the form

\[
\int_{\Omega} \left\{ \left[ u_R(\pi_{LBR}(x_L)) - u_R(\pi_{LTR}(x_L)) \right] - \left[ u_L(x_L) - u_L(\pi_{LBT}(x_L)) \right] \right\} \left[ H(\phi(x_L)) - 0.5 \right] \, dx .
\] (7.9)

In the sequel, we explain how this term was constructed (Figure 7.3).

First of all, let it be pointed out that, alternatively, one could also think about using the terms constructed in a more straightforward way, for example, according to the following formula

\[
\int_{\Omega} \left[ \left[ u_R(\pi_{LTR}(x_L)) - u_L(x_L) \right] \right]^2 H(\phi(x_L)) \, dx \\
+ \int_{\Omega} \left[ \left[ u_R(\pi_{LBR}(x_L)) - u_L(x_L) \right] \right]^2 \left[ 1 - H(\phi(x_L)) \right] \left[ 1 - H(\pi_{LBT}(x_L)) \right] \, dx
\] (7.10)

expressing the fact that the left and right image should correspond at the points where both the cameras see the objects or both the cameras see the base plane. (Later, however, we will show that such terms are not too useful.) For the objects, no occlusion occurs. Therefore, the objects are always seen by both cameras, i.e., the correspondence for the objects can be expected whenever \( \phi(x_L) > 0 \), which explains the term with \( H(\phi(x_L)) \). The value \( \phi(x_L) < 0 \) indicates that the left camera sees the base plane. Due to the occlusions by objects, however, not every point of the base plane that can be seen by the left camera can also be seen by the right camera. Matching the base plane can only be done on the area that is seen by both cameras, which is expressed by the terms \( 1 - H(\phi(x_L)) \) (which is the part of the bottom plane that is seen by the left camera) and \( 1 - H(\phi(\pi_{LBT}(x_L))) \) (the part that is seen by the right camera). Clearly, if \( \phi(x_L) \) correctly separates between the objects and the base plane, the value of the expression
stated above is minimal and, for theoretical images, it can even take the value of zero. Unfortunately, the expression is not suitable for the use in the functional that should be minimised, which is due to the fact that the minimum can also be achieved for incorrect object detections. Consider, for example, a scene that contains only the base plane with a constant brightness and no object. The zero, i.e., the minimal value of the difference

\[ \int_\Omega \nabla u_L(x_L) \cdot \nabla \phi(x_L) \, dx \]

is depicted for the case that the object is brighter than the base plane and providing that the correspondence term from Eq. (7.10) does not prevent from the detection of false objects, which is threatening in the situation that cannot be regarded as rare (objects on a base plane with a constant brightness). The terms of this type are used in more approaches \[ \text{[3, 7, 39]} \]; similar problems can also be seen in the methods based on correlation \[ \text{[46]} \]. Generally speaking, the methods based on concordance of brightness between the image points (small brightness difference or big correlation) are suspected to have problems since the brightness concordance between two points is ambiguous. In the case of concordance, we can only say that they can (but need not) correspond. In the case of a brightness mismatch, on the other hand, we are sure that the points do not correspond.

For evaluating the degree of correspondence, therefore, we introduce the term from Eq. (7.9) that takes into account the mismatch of brightness. (The whole functional is presented in Eq. (7.11).) The way in which the new term works is illustrated in Figure 7.3. The value of the difference \[ \int_\Omega \nabla H(\phi(x_L)) \cdot d\sigma \]

is depicted for the case that the object is brighter than the base plane and providing that the object and the base plane have a constant brightness. The value of the term \[ H(\phi(x_L)) \] from Eq. (7.9) is either 0.5 for \( \phi(x_L) > 0 \) indicating the hypothesis that the object is present in that area or \(-0.5\) otherwise (Figure 7.3). It can be easily seen that the correspondence term from Eq. (7.9) takes its lowest possible value if \( \phi(x_L) \) fits to the object boundary and in no other cases; the lowest possible value is negative. The correspondence term can also be viewed as a dot product between the difference term and \[ H(\phi(x_L)) \] \(-0.5\). The value of the dot product is the lowest (highest in absolute value) if \( \phi(x_L) \) fits to the object. The restriction of the term is that all the objects should be either darker or brighter than is the background.

We can conclude that the functional we use is the following one

\[
F(\phi) = \mu \int_\Omega \nabla H(\phi(x_L)) \cdot d\sigma + \nu \int_\Omega H(\phi(x_L)) \, dx + \lambda \int_\Omega \left\{ \left[ u_R(\tau_{LBR}(x_L)) - u_R(\tau_{LTR}(x_L)) \right] - \left[ u_L(x_L) - u_L(\tau_{LTB}(x_L)) \right] \right\} \left[ H(\phi(x_L)) - 0.5 \right] \, dx + \kappa \int_\Omega \left| \nabla u_L(x_L) \right|^2 H(\phi(x_L)) \, dx + \kappa \int_\Omega \left| \nabla u_R(\tau_{LTR}(x_L)) \right|^2 H(\phi(x_L)) \, dx.
\] (7.11)
Figure 7.3: The values of the terms from Eq. (7.9) for the image in which the base plane has a constant brightness of \( b_B \) and the object has the brightness of \( b_T \). We suppose that \( b_T > b_B \) and we introduce \( b_T - b_B \equiv \delta b \). Independently on \( \phi(x) \), the difference term 
\[
[u_R(x_{LBR}(x_L)) - u_R(x_{LTR}(x_L))] - [u_L(x_L) - u_L(x_{LTB}(x_L))]
\]
takes the value of \( +\delta b \) in the neighbourhood of object, the value of \( -\delta b \) inside the object (but in the neighbourhood of its boundary), and the value of zero otherwise. The term \( H(\phi(x_L)) - 0.5 \) takes the value of either \(-0.5\) or \(+0.5\). The value of Eq. (7.9) is minimised if \( \phi(x_L) \) fits to the object.
The last two terms in the last row are optional. They were added to express the fact that the edges (big gradients) are not usually expected in the area of objects. The terms are not needed in the theoretical images according to the model that was introduced before, but may be useful in real-life images.

The rationale behind the construction of the functional in this way can be easily seen. The new term (with the $\lambda$ coefficient) detects the places of mismatch between the images (Figure 7.3) and should be minimised as was explained before. The term with the $\mu$ coefficient contributes to connecting such places into the whole objects, i.e., it does not allow that $\phi$ only indicates the places with the differences that occur near the object boundary (Figure 7.3).

The Euler-Lagrange equation corresponding to the functional from Eq. (7.11) (including the last two optional terms) is of the form

$$
\delta(\phi(x_L)) \left[ \mu \text{div} \left( \frac{\nabla \phi(x_L)}{|\nabla \phi(x_L)|} \right) - \nu \right]
+ \lambda \delta(\phi(x_L)) \left[ \left( u_R(\pi_{LBR}(x_L)) - u_R(\pi_{LTR}(x_L)) \right) - \left( u_L(x_L) - u_L(\pi_{LTB}(x_L)) \right) \right]
- \kappa \delta(\phi(x_L)) \left[ |\nabla u_L(x_L)|^2 + |\nabla u_R(\pi_{LTR}(x_L))|^2 \right] = 0.
$$

The equation can be solved in the usual way by making use of $H_\epsilon(\cdot)$, $\delta_\epsilon(\cdot)$, and the time parametrisation [21, 22]. The projective transformations $\pi_{LBR}(\cdot)$ etc. can be determined from the positions of the planes containing the bottom and top points and from the positions of both cameras. Practically, they can also be determined from the point correspondences in a pair of calibrating images containing both bottom and top points. We will describe it in more detail in Section 7.4.2.

### 7.4 Tests and Results

The algorithm has been tested on artificial and real-life images complying with the model introduced before. We have prepared a set of synthetic images to test the impact of each term of the functional on the final segmentation. The images contain various combinations of foreground and background colours (intensities), and objects with different shapes and size. The input pairs represent the images of the left and right camera. The images were created artificially by shifting selected parts of the right image (the shift simulates the disparity). An example is given in Figure 7.4. The spiral patterns have the same size, shape and brightness but differ in disparity; thus only the left one should be labelled as the foreground object. The results in Figure 7.5 illustrate the superior depth selectivity of our algorithm and also the correct solution for the reference.

The real-life images, used for the testing, were taken by a pair of cameras mounted above a university parking lot. These cameras were mounted as a part of Volkswagen
7.4 Tests and Results

Figure 7.4: The artificial images contain a pair of dark spiral patterns. In order to highlight the advantage of our algorithm, the spiral pattern on the left is shifted to mimic the different spatial depth, whereas the right one is left in the same position.

A research project focused on developing the algorithms for monitoring the occupancy of the parking lots. The images of biological samples were taken by our scanning device described in Chapter 6.

In the sequel, we will describe the tests and the comparison of our algorithm with the traditional ones in more details.

7.4.1 Tests with Artificial Images

Firstly, synthetic images with objects of different shapes and added noise were generated. Figure 7.6 depicts several examples. The brightness of images was normalised into the range \([0, 1]\). The parameters from Eq. (7.12) were chosen to be \(\mu = 1, \nu = 0.01, \lambda = 1, \kappa = 0, \) and \(\Delta t = 10\).

The first experiment studies the effect of noise in the input images on the output segmentation. The pixel error rate (PER) was computed for different distorted test images with multiple levels of uniform noise. The images were disrupted by adding uniform noise controlled by an attenuation factor in the range \([0, 1]\), where 0.0 indicates original image and 1.0 indicates pure noise. Figure 7.6 shows plots of the pixel error rate (PER) for five different sample images. Misclassified pixels are the top pixels classified as the bottom pixels or vice versa. As can be seen, the algorithm provides still more than 50% correct matches on images corrupted by uniform noise attenuated by factor 0.5. Also please notice that there are no false matches on the other parts of the images.

To get an idea about how the algorithm performs compared to the standard stereo matching algorithms we have run a test on the same dataset with the following algorithms – Semi-Global Block-Matching [60], Consistent Semi-Global Matching [61], Dynamic programming [32], Graph Cuts [74], Sum of Squared Differences, Variational Stereo Correspondence [80]. The outputs of the algorithms were converted to a binary image, classifying each pixel as the layer that best fits the value of the pixel’s disparity.
7.4 Tests and Results

Figure 7.5: The results on the artificial images show that our algorithm maintains the unique features of the level sets, such as the automatic handling of topological changes and extends the original algorithm by the depth segmentation. In the first row of the figure, there are the original images with the segmentation borders. The second row shows the segments – foreground (green) and background (red). As you can see, the pattern on the background (right spiral) is correctly not detected.

The nearest plane for the given pixel disparity is chosen. This step, which is necessary for the comparison, may be considered as inappropriate, but we decided to use it rather than changing the algorithms. The imposed change in the algorithms could negatively influence the performance and make the further comparison more difficult.

The performance of the algorithms is depicted in Figures 7.7 and 7.8. Our algorithm based on the level set method is denoted as $ls$. The results show that our algorithm outperforms the remaining approaches on all test pairs up to the noise level 0.5. Besides, our algorithm maintains extremely low false rate of classifying the bottom layer as the top layer. We consider this behaviour as more convenient, especially in applications for detecting the objects. From our experience, the missing detection of the existing object is less harmful than the detection of the non-existing object. As you can see, the second best algorithm in our test, the graph cut approach [74], had more false classifications, especially on the bottom layer.

7.4.2 Real-Life Applications of the Method

In this section, we describe the experiments with real-world images. The method benefits from a strong a priori knowledge of the scene. The model of the scene, on the other hand, is also a strong limitation of the method; the method can be used only in specific cases. We used the method for detecting the cars on parking lots for the purpose of detecting the occupancy of particular places. The motivation for using the stereo pair
Figure 7.6: The graph shows the dependency of the pixel error rate (PER) on increasing level of uniform noise. The images depict the artificially created samples and the segmentation results. As you can see, our algorithm shows good noise immunity even with addition of uniform noise. Moreover, its design properties prevent false detection of foreground layer.
Figure 7.7: Comparison of depth segmentations with different algorithms (P. 1). The compared algorithms are: Semi-Global Block-Matching (sgbm) [60], Consistent Semi-Global Matching (hh) [61], Dynamic Programming (dp) [32], Graph Cuts (gc) [74], Sum of Squared Differences (ssd), Variational Stereo Correspondence (var) [80], and our algorithm based on Level Sets (ls).
Figure 7.8: Comparison of depth segmentations with different algorithms (P. 2). The graphs show a comparison of our algorithm with other major approaches on the data set containing the artificially created images. The experiment results show that our algorithm (denoted as ls) outperforms the remaining approaches on all test pairs up to the noise level 0.5. The graph depicts the number of correctly labelled pixels as the foreground (top layer) or the background (bottom layer). The increasing noise level in the images tends to confuse our algorithm to consider erroneously the foreground layer as the background. However, our algorithm has almost no false detections of the foreground parts, and significantly surpasses other approaches which tend to create false detections on noisy images.
for solving the mentioned problem is to overcome some difficulties that are reported by
the authors using only one image (e.g., [43]). Moreover, the detection of an object with
a certain height is a reliable marker that the place is occupied, which is not offered by
one-image approaches that may be deceived more easily.

The usual situation in the parking lots nearly corresponds to the model we introduced
before (Figure 7.9). The base plane is present and the roofs of cars may be regarded as
objects. Some differences from the model, however, exist. Namely: (i) the cars do not
have only the roof, but they also have the whole body containing the parts at different
heights; (ii) the cars may be of different heights. Nevertheless, the following can be
assumed: (i) the roof is usually the most prominent part of car in the images (Figure
7.9), (ii) for cars of a certain category, e.g., the passenger cars, more or less the same
height of them can be expected.

The images are provided by a pair of uncalibrated cameras; the lens distortion is not
compensated. The transformations \( \pi_{\text{LTR}}(x_L) \), \( \pi_{\text{LBR}}(x_L) \), \( \pi_{\text{LTB}}(x_L) \), and \( \pi_{\text{LBT}}(x_L) \) have
chosen to be two-value functions that have the form of bicubic polynomials whose coef-
ficients were found by the least squares method from two sets of calibrating points that
were determined manually (only the mentioned transformations are needed for each \( x_L \);
the exact type of projection and distortion need not be known). The first set of calibrat-
ing points contained selected corresponding points lying on the ground of parking lot;
the second set contained selected corresponding points on the car roofs (each pair of
images offers a lot of suitable calibrating points of both types). The bicubic polynomials
proved to be fully capable to model the values of the transformations in our rather heav-
ily distorted images. The size of the images was 1920 \( \times \) 1080 pixels. The initial value of
\( \phi(x) \) was set on the basis of the difference \( [u_R(\pi_{\text{LBR}}(x_L)) - u_R(\pi_{\text{LTR}}(x_L))] - [u_L(x_L) - u_L(\pi_{\text{LTB}}(x_L))] \). At the places, where the value of difference was less than a small neg-
ative threshold, the value of \( \phi_0(x) \) was set to 1 (object); otherwise, it was set to −1
(background). Objects brighter than is the background were expected. The results of
detection can be seen in Figure 7.9.

As for the images of the biological samples, we have used the image pairs (1280 \( \times \) 1024
pixels) taken from the observation of the Bazzania trilobata. The processed images were
already rectified and lens distortion was compensated. The bimodal depth segmentation
was intended to extract the area of the upper layer of the plant surface. This value is
needed to estimate and model water flux of their growing upper layers.

The upper layer is typically brightly coloured, but as we can see on the subfigure
7.10a, there are other bright objects on the surface (e.g., needles on the left top), so we
cannot rely on the segmentation based solely on the pixel colour (or the intensity). This
can be illustrated on the subfigure 7.10c, where the original Chan and Vese algorithm
cannot properly distinguish whether the bright object is part of the upper or lower layer.
Emphasizing this may seem as a triviality, but the biologists insist in calculating only
the surface above the base level. As you can see on the subfigure 7.10d, our approach
This chapter presents an algorithm for bimodal depth segmentation based on the level set formulation. The herein described method separates the planar objects of arbitrary shapes lying in a certain height above a background (base plane) using the information from the stereo image pair. The depth segmentation is formulated as a minimisation problem. For this purpose, we have proposed a new functional that is based on evaluating the mismatches between the images. In this, our method contrasts with the usual approaches that evaluate the matches. We have found out that the mismatches are stronger clues for driving the propagation of the level set function than the matching criteria. In the Section 7.3 we have discussed the importance of these observations. The minimisation is carried out by making use of the Euler-Lagrange equation and the level set function. The results show good noise immunity, as verified with synthetic and real images. We believe that the achieved results can be regarded as good and promising. The general explanation for this is that by exploiting strong a priori known properties of scene, it is possible to achieve better results, which especially holds for the real-life images that are often difficult for the general-purpose algorithms.

The proposed variational level set formulation of the depth segmentation has three main advantages over the traditional formulations: (i) The method is capable of very...
(a) The original image of *Bazzania trilobata*.

(b) The final composition of the segmentations depicting the top layer (green) and bottom layer (red) of the plant structure.

(c) The output of the original Chan and Vese level set.

(d) The output of our algorithm correctly separates only the top layers of the bright plant structures.

Figure 7.10: The results of the plant layer segmentation. Basically, the task is to separate the highest layer of the plant and measure its surface area. The upper layer is typically brightly coloured, but as we can see on the subfigure 7.10a, there are other bright objects on the surface (e.g. needles on the left top). The original Chan and Vese algorithm based on the pixel intensities cannot distinguish whether the object is part of the upper or lower layer (see subfigure 7.10c). Our approach is able to distinguish these differences and separate only the object on the foreground (upper layer). The result of our algorithm is depicted on the subfigure 7.10b.
7.5 Summary

Sensitive segmentation based on the novel terms introduced for detecting the depth. (ii) This approach benefits from a strong a priori knowledge of the scene and in such configurations supersedes other methods. (iii) Our modification preserves the advantages of the original Chan-Vese model that detects objects whose boundaries are not necessarily defined by a gradient.

This chapter also shows how the research in the field of level sets and bimodal depth segmentation has been linked to the practical implications. As an example, we have provided an application of the parking lot occupancy detection and the segmentation of the liverworts layers.
The following novel algorithm is based on an observation that the image can be segmented into the regions representing the objects or their parts. This method, based on the mean shift algorithm, tries to match the segments of the input images instead of working with the individual pixels.

The algorithm presented in the chapter was invented during our studies of the mean shift algorithm. We found the concept of the mean shift attractors (which will be described later on) very attractive and thought, whether it can be also applied in the stereo matching algorithms. In this chapter, we will show that the answer is yes and, moreover, the algorithm based on the attractor matching can provide very promising results.

Our main idea is to take advantage of the transformation of the input images into the mean shift attractor space and then perform the matching in the attractor domain instead of using the original pixel domain. The attractors can be seen as a form of compressed information and thus can be matched more quickly and easily. We believe that the proposed attractor matching is less susceptible to errors caused by the noise and other distortions. By using this approach, we were able to perform fast and precise matching, especially on image parts containing large homogeneous areas. The results of the experiments support our hypothesis.

It is interesting that the proposed method differs from the processes taking place in our brain. The neuropsychological observations show that the human stereopsis is a bottom-up process, occurring even before a shape perception. While this approach might be suitable for humans (who try to mimic this behaviour with computer programs), it is essential to note that the similar machine processing is not always so efficient. Therefore, some researchers suggested an opposite top-down approach – at first identify the shapes and structures and then refine the results on the pixel to pixel basis. The following algorithm presents an example of such method.

We compared our algorithm with other state of the art approaches. The algorithm performs well on images with a lot of homogeneous areas (images of man-made structures, buildings, etc.) The results and the description of this algorithm were published in ISVC proceedings [81] in 2010.

8.1 MEAN SHIFT ALGORITHM

The mean shift is a non-parametric feature-space analysis technique, founded on a density gradient estimation using a generalized kernel approach. It is based on an observa-
8.1 MEAN SHIFT ALGORITHM

tion that the value of a density function can be estimated using the sample observations falling into a certain area around the point. The feature space can be considered as the empirical probability density function of the represented parameters [29]. The algorithm seeks for the local maxima of the probability density function (given by the samples). The dense regions in the sample (feature) space correspond to the local maxima. Once the local maxima are located, the segments can be recognized as the clusters associated with it.

In the majority of the mean shift implementations, the kernel density estimator is defined as follows: Assume we have \( n \) data samples \( x_i, \ldots, x_n \) in the \( d \)-dimensional space \( \mathbb{R}^d \). The kernel density estimator is defined as function

\[
\tilde{f} = \frac{1}{nh^d} \sum_{i=0}^{n} K \left( \frac{x - x_i}{h} \right), \tag{8.1}
\]

where \( h > 0 \) is a single bandwidth parameter and \( K \) is a radially symmetric kernel satisfying

\[
K(x) = c_k k(||x||^2), \tag{8.2}
\]

with a normalization constant \( c_k \) and a function \( k(x), x > 0 \) called the kernel profile. Several types of profile functions \( k(x) \) are commonly used, e.g. uniform, triangle, Epanechnikov, quartic, Gaussian, and cosine. For our purposes, we used the Epanechnikov kernel defined as

\[
k(x) = \begin{cases} 
1 - x & 0 \leq x \leq 1 \\
0 & x > 1
\end{cases} \tag{8.3}
\]

The density maxima are found when the gradient of density estimator \( \nabla \tilde{f} \) is equal to zero. The density gradient estimator can be rewritten as follows:

\[
\nabla \tilde{f} = \frac{2c_k}{nh^{d+2}} \left( \sum_{i=0}^{n} g_i \right) \left( \frac{\sum_{i=0}^{n} x_i g_i}{\sum_{i=0}^{n} g_i} - x \right), \tag{8.4}
\]

where \( g(x) = k'(x) \) and \( g_i = g(||\frac{x - x_i}{h}||^2) \). The third term of Eq. (8.4) is the mean shift vector \( m(x) \). The mean shift vector points to the region of a maximum point density. For each sample we compute the successive locations, given by the shifts of the mean shift vector. Thus we define a path to a convergence point, referred as an attractor. For the segmentation purposes, we have to group all attractors closer than a given threshold in the spatial and range domain.

The mean shift procedure was firstly described in 1975 by Fukunaga and Hostetler [50] and reintroduced in 1995 by Cheng [26]. The readers who are interested in a detailed description of the mean shift algorithm and its applications in low-level vision tasks and image segmentation can also read a comprehensive study written by Comaniciu and Meer entitled “Mean Shift: A Robust Approach Toward Feature Space Analysis” [29]. In the next section, we will describe the process of exploiting the known attractors of both images for performing the stereo matching.
8.2 Matching in the Attractor Space

The basic idea underlying our approach is quite simple. As we have mentioned in the state of the art (Section 4.7), some of the stereo matching algorithms apply the mean shift segmentation on the pair of input images to find the pixel aggregations. In our proposed matching process, we use only certain part of the mean shift segmentation process.

Our algorithm can be divided into three successive steps: (i) Attractor clustering; grouping the convergence points in order to present more coherent objects. It is necessary as the mean shift tends to over segment the input images. (ii) Attractor matching; establishing the matches between the attractor clusters. The goal is to match as many pairs of attractor clusters as possible. We assume that we are matching the segments representing the same object as seen from the different points of view (i.e., the left and right image). (iii) Calculating the attractor disparity and utilizing the value in, e.g., window-based algorithms or plane fitting techniques to improve the overall performance. By having the correct matches, we can also calculate the disparity directly.

The next paragraphs describe the utilization of the mean shift clustering and its attractor domain in more details.

8.2.1 Attractor Clustering

Let \( x^L_i, x^R_i, i = 1, \ldots, n \) be the pixels of the input images, and \( z^L_i, z^R_i \) are the mean shift convergence points of the left and right image, calculated as described in Section 8.1.

For each image, we need to create a set of attractor clusters \( \{ A^L_k \}_{k=1}^{p} \) and \( \{ A^R_l \}_{l=1}^{q} \) (for the left and right image). The clusters are formed by grouping those convergence points \( z^L_i \) (\( z^R_i \), respectively) that are mutually closer to each other in the spatial domain than the given threshold \( d_A \). In other words, the cluster can be seen as a group of points in the attraction basin (the region of maximum point density, where the convergence paths lead). Unfortunately, we do not usually get a single attractor for the specific image area and therefore, we are forced to group the neighbouring attractors into a single “super” attractor to achieve better performance in the next steps. The grouping can be performed in several ways. The attractor grouping using a simple agglomerative method is not very effective, the more convenient way is to map the convergence points into the fixed grid and connect them using the component labelling algorithm, e.g., the flood fill. The flood fill algorithm starts with the first attractor in the grid, relabels it as the first “super” attractor and recursively looks for all other attractors in the given grid which are connected to it and relabels them as well. Then, the algorithm continues with the next not relabelled attractor until all attractors are parts of some “super” attractor.

We recommend this approach as it is faster, and clusters can be easily accessed since the grid is directly mapped to the original image grid.
8.2 Matching in the Attractor Space

For each attractor cluster \( A \) (in the previous paragraph denoted also as the “super” attractor), we calculate its centroid \( \mathbf{C} = \{ C_x, C_y, C_c, C_w \} \), representing the \( x, y \)-coordinates, colour and number of its convergence points). The value \( C_w \) is the number of the attractors that were merged into the “super” attractor. These centroids create an auxiliary structure we will use to improve the implementation performance of the next steps.

The primary motivation of this step is to provide clustering that will separate the image into the coherent objects. We assume that the same object will be decomposed into almost the same group of segments on both images (left and right), despite the perspective projection. The idea is that we do not have to do pixel-to-pixel matching, as far as we are able to disassemble the objects into the similar matching segments.

8.2.2 Attractor Matching

Attractor matching is done between the sets of \( A^L_k \) and \( A^R_l \). We are looking for the most similar pair of attractor clusters of the left and right image. The similarity is evaluated on the basis of the cluster size and its average colour. In our opinion, the attractor clusters are more suitable for matching than the other segment features or moments. The attractors can be seen as a form of compressed information and thus can be matched more quickly and easily. We believe that the proposed attractor matching is less susceptible to errors caused by the noise and remote observations (outer points shifting the moments into the false values). The attractor matching is performed as follows.

Initially, we do not expect the right matching cluster to appear outside the horizontal stripe given by the range of vertical positions of the left cluster samples. In this region, we will look for the pairs sharing the similar colour and cardinality (number of attractors creating the cluster). The search range is limited to the rectangular area bounded by \([d_{\text{min}}, d_{\text{max}}]\), the minimum and maximum allowable disparity in the horizontal direction and \([\min(z_{y,L}^L), \max(z_{y,L}^L)]\), \( z_{y,L}^L \in A^L_{m} \), the minimum and maximum \( y \)-coordinates of the left image attractors connected with the given cluster (“super” attractor) in the vertical direction.

The matching factor is computed as a weighted sum of squared differences of the average attractor colour and size (the equal results can be obtained using the already calculated values \( C_c, C_w \) of each attractor). The assumption is that if the attractor clusters represent the same object on both images, the attractors itself should be very similar (since they were formed by the density gradient of the samples depicting the same object, only from a slightly different position). The cost of the matching is defined as follows

\[
M_c(k, l) = w_1 \left( \frac{\sum_{z^L_c \in A^L_k} z^L_c}{|A^L_k|} - \frac{\sum_{z^R_c \in A^R_l} z^R_c}{|A^R_l|} \right)^2 + w_2 \left( |A^L_k| - |A^R_l| \right)^2, \tag{8.5}
\]
8.2 Matching in the Attractor Space

which can be equally rewritten using the centroid values described in Section 8.2.1 to

\[ M_c(k,l) = w_1 (C_c(k) - C_c(l))^2 + w_2 (C_w(k) - C_w(l))^2, \]  
(8.6)

where the parameter \( w_1 \) controls the weight of the colour constraint, presuming that the matching attractors will have almost the same average colour. The parameter \( w_2 \) controls the weight of the size constraint (count of the attractors in the cluster), practically meaning that the number of convergent points should be approximately the same. Despite the fact that these limited (as to the number) constraints can induce certain matching errors, they were chosen as the compromise leading to the acceptable running times. We can think of more advanced pattern matching techniques for the cluster comparison, but they were not used as they provided only small improvements in our experiments.

In the implementation, we sort the attractor clusters according the cardinality (size) and start the matching process with the largest ones. The best match \( M_b(k) \) between the attractor clusters is the one that minimizes the matching cost

\[ M_b(k) = \arg\min_l M_c(k,l), \]  
(8.7)

where the possible matches are located in the searched area as described before.

The method, in which each cluster is matched against all other clusters in turn, is not very efficient. Therefore, we suggest using a heuristic approach, starting with the largest clusters and then moving to the smaller ones. Clusters are matched using the sorted list to successively limit the number of possible candidates due to the size constraint (by matching the largest one, we significantly reduce the search space of the next match).

Figure 8.1 shows an example output of the matching, produced by the proposed approach. Matching segments are labelled and coloured for better readability. As you can see, matching the large segments is quite simple, but the smaller ones are more challenging since they are often ambiguous for a unique match (this will be discussed in the following section).

8.2.3 Disparity Calculation

As for the large segments, we calculate their disparities using the horizontal difference of the matched attractor centres given by

\[ d_k = C_x(M_b(k)) - C_x(k), \]  
(8.8)

and consider it as the disparity of the large segments. The first term is the horizontal position of the best matching cluster centre (shifted by the disparity), the second one is the position of the original cluster (in the reference position).

While the large segments can be matched easily, small clusters are hard to match, mainly due to the ambiguities. Therefore, the areas containing a large set of small segments have to be solved differently than by the proposed attractor matching.
8.3 Tests and Results

The decision, whether to use the cluster disparity or the other approach (discussed in the next paragraph), is based on the difference of $M_c$ between the best match and the second best candidate. If the difference is small (i.e. the both clusters are good candidates), we switch to the disparity calculation by the other means than by the Eq. (8.8).

As for the other method (performing the matching of the small segments), we have implemented a local method based on the matching windows, shaped according to the image segments obtained from the mean shift segmentation. The cost function is computed using the standard SAD method (the implementation was done according to [35]).

The final disparity map is composed of the disparities $d_k$ (associated with the matched attractors) and disparities obtained from the window matching.

8.3 Tests and Results

The success rate of our algorithm (and the quality of the disparity map) is strongly dependent on the success of the attractor matching. The images containing easily segmentable parts are more suitable for our approach than the highly detailed scenes with a lot of textures. As the idea of matching the segments, represented by a mean shift attractor, is theoretically very tempting, the practical results show that the application has some limits.

At first, even if we find the correct match, we do not know the shape of disparity (or the disparity gradient) of the segment. We are able to match the object (segment)

Figure 8.1: The visualization of matched image segments (Venus pair) as found by the proposed algorithm. Matched segments are labelled according to the size and coloured for better identification.
8.3 Tests and Results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Tsukuba nocc</th>
<th>Venus nocc</th>
<th>Teddy nocc</th>
<th>Cones nocc</th>
<th>Map nocc</th>
<th>Map all</th>
<th>avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>agcp [127]</td>
<td>1.03</td>
<td>1.29</td>
<td>0.10</td>
<td>4.63</td>
<td>6.47</td>
<td>1.81</td>
<td>5.70</td>
</tr>
<tr>
<td>hh [61]</td>
<td>2.61</td>
<td>3.29</td>
<td>0.25</td>
<td>5.14</td>
<td>11.8</td>
<td>2.77</td>
<td>8.35</td>
</tr>
<tr>
<td>bp [76]</td>
<td>0.97</td>
<td>1.75</td>
<td>0.16</td>
<td>6.47</td>
<td>10.7</td>
<td>4.79</td>
<td>10.7</td>
</tr>
<tr>
<td>sgbm [60]</td>
<td>3.26</td>
<td>3.96</td>
<td>1.00</td>
<td>6.02</td>
<td>12.2</td>
<td>3.06</td>
<td>9.75</td>
</tr>
<tr>
<td>gc [74]</td>
<td>1.94</td>
<td>4.12</td>
<td>1.79</td>
<td>16.5</td>
<td>25.0</td>
<td>7.70</td>
<td>18.2</td>
</tr>
<tr>
<td></td>
<td>msc</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5.12</td>
<td>6.88</td>
<td>2.68</td>
<td>16.26</td>
<td>24.4</td>
<td>10.1</td>
<td>18.8</td>
</tr>
<tr>
<td>dp [32]</td>
<td>4.12</td>
<td>5.04</td>
<td>1.01</td>
<td>14.0</td>
<td>21.6</td>
<td>10.5</td>
<td>19.1</td>
</tr>
<tr>
<td>ssd</td>
<td>5.23</td>
<td>7.07</td>
<td>3.74</td>
<td>16.5</td>
<td>24.8</td>
<td>10.6</td>
<td>19.8</td>
</tr>
</tbody>
</table>

Table 8.1: The performance of our algorithm according the Middlebury stereo evaluation [125]. The overall performance is measured by the percentage of bad pixels (PER) in the non-occluded areas (nocc). The performance measured on the whole image (all) is provided as well. Our algorithm is denoted as msc. In order to give a better idea of the performance of our method, we provide a comparison to the other popular stereo matching algorithms (the full list of more than 150 algorithms is available on the [125] website). To sum up, our algorithm provides promising results on image pairs, where the attractor matching is unambiguous (e.g., the Venus pair) but the overall results on all images are, mainly due to a missing optimization, average.

and establish its average disparity, but our approach gives no additional information about the disparity of its surface. This disadvantage brings certain specific errors into the final outputs. We use the plane approximation, but as we have discovered, it does not provide optimal results, especially when compared to the recently added algorithms (in the Middlebury dataset [125]) based on the global optimization.

The experimental results of our approach are shown in Table 8.1. For the performance evaluation, we have followed the methodology described by Scharstein and Szeliski [125]. The complete list of the Middlebury stereo evaluation has more than 150 entries. Our table contains only a short selection of the algorithms that provided better results, and were discussed in the state of the art section (Chapter 4).

The best results can be expected for the images that can be easily segmented. An example of such image is the Venus pair. The result for this pair is quite good. On the contrary, the Teddy image creates a difficult configuration with a lot of details and smooth surfaces. This decreases the number of successfully matched attractors and makes the overall performance worse.

We have to point out that during the period between now and the publication of our paper [81], literally tens of new algorithms appeared in the Middlebury evaluation and, regrettably, our algorithm can be nowadays considered as outdated. However, it
is still the only algorithm (as far as we know) based on the matching of the mean shift attractors and we consider this idea as our main contribution of this chapter.

8.4 Summary

The algorithm described herein shows the possibility of using the mean shift attractors as the matching features (i.e., the objects of matching). Our main idea is to consider the mean shift attractor as a sort of aggregated value reflecting the segment properties. We assume that the same object will be decomposed into almost the same group of segments on both images, despite the perspective projection. We do not have to do pixel-to-pixel matching, as far as we are able to disassemble the objects into the similar segments (we are able to match).

The implementation showed that such approach is feasible and provides good results for the large segments on the images; nevertheless the remaining (smaller) parts must be matched in the other way. The experiments presented in this chapter were based on the combination of the proposed approach and the classical window method for matching the remaining parts of the image.

Although the idea of attractor matching did not provide extraordinary results for all set of images, it produces (for the certain types of input images) results that are above the average. The algorithm performs well on images with a lot of homogeneous areas (images of man-made structures, buildings, etc.) The best results can be expected when the image can be easily segmented. The advantage of this method is that it can handle the untextured regions very naturally. Critically, we have to admit that the presented algorithm does not always guarantee the significant improvements for all possible inputs. The overall score can be, in the light of recent advances in the stereo matching algorithms, ranked as average (the algorithm would appear in the second third of the ranking table). However, we are convinced that there is a further potential in exploring the mean shift attractor space.
This chapter presents a new stereo segmentation algorithm. Our algorithm extends the popular fuzzy c-means clustering method by introducing an additional disparity cue. In our opinion, the proposed approach presents an interesting theoretical concept. In the following sections, we will provide the necessary theoretical background as well as practical results in our applications. For our specific purposes (the moss segmentation), the algorithm provides very good results.

The next sections describe the adaptation of the fuzzy c-means algorithm to perform the clustering in the disparity space. The creation of the clusters will be driven by a degree of the stereo match (this measure will be described later on). An attractive aspect of this strategy is that we are able to take advantage of known number of depth levels or objects (if this information is available). The following fuzzy c-means based algorithm evaluating a stereo pair is, as far as we know, a novel concept. We consider the proposed depth extension of the fuzzy c-means as a main contribution of this chapter.

The motivation for our work was to provide an algorithm that can separate objects based on their different colour and spatial depth. We regard this method as more suitable in specific cases (will be described later on) than the segmentation of the final disparity maps. The distance, based on both dissimilarities (spatial and colour), provide more sensitive segmentation (especially on segment borders) than the segmentation performed on the filtered disparity maps which contain only the best matches, and do not take into account segment properties. The algorithm was originally developed for segmenting the moss clusters. Therefore, we have tested and evaluated the algorithm mainly on the "Map" dataset (see Figure 9.1) as it strongly resembles the stone structures which are frequently covered by the moss layers.

In this chapter, we will actually describe two algorithms. The main algorithm (and also the main contribution of the chapter) is the extension of the fuzzy c-means clustering. We will describe this method in Section 9.2. The algorithm is more designed for the segmentation purposes than calculating the dense disparity maps (but is capable of that). The second algorithm, described in Section 9.2.1, is the application of the previous method in the window based stereo matching. We use the information of the pixel membership in the clusters to create more accurate aggregation windows. As to the second algorithm, we do not consider it as such important as the main idea presented in the first one.
9.1 FUZZY C-MEANS

Figure 9.1: The “Map” dataset – input images and the ground truth. The texture strongly resembles the stones (which are in our case covered by the moss).

Disparity Clustering

For the completeness, we shall clarify the term clustering. Generally, clustering is described as a process of forming partitions from a data set on the basis of a performance function, also known as an objective function; the details will be described later. Partitions are defined as mutually non-overlapping groups, where the elements within a group are close or similar to each other, but dissimilar from elements in other groups.

The underlying idea of our algorithm is to consider the disparity map as a specific type of the data set, consisting of clusters representing the three dimensional objects of the scene. The idea of using the clustering techniques inside the stereo matching process is not new. This approach was introduced in [133, 11] and further developed in [152, 134, 132, 95]. In contrast to these, our approach differs in several aspects. The clustering is not performed on the individual input images, but on both stereo images, taking into account the matching properties. In each step, the clusters are adjusted to minimize the matching costs.

In the following sections, we will introduce and describe the significance of the particular steps used for modifying the fuzzy c-means algorithm and the results obtained using this novel approach. We have to notice that at the time of writing, this part of the thesis has not yet been published, but has already been submitted for publishing.

9.1 FUZZY C-MEANS

Let us briefly introduce the original method. Fuzzy c-means is a widely used clustering technique, developed by Dunn [42] and improved by Bezdek [10]. It is based on a standard least squared error model that generalizes an earlier and popular non-fuzzy c-means mode [101]. Fuzzy c-means can be generalized in many ways to include, e.g., Minkowski, Hamming, Canberra or hybrid distances.
The fuzzy c-means algorithm attempts to partition a collection of \( n \) data points \( \{x_k\}_{k=1}^n \) into a collection of \( c \) fuzzy clusters (represented by the cluster centres) on the basis of a distance \( d \) between the cluster centre and the data point. The algorithm is minimizing the objective function \( J(U, V) \), where \( V = (v_1, \ldots, v_c) \) is the set of cluster centres and \( U = [u_{ki}] \) is the \( n \times c \) membership matrix. The space of all possible values of \( U \) is denoted as \( U_f \). The elements of the matrix \( U \) are organized as follows. The column \( i \) gives the membership of all \( n \) input data points (rows) in the cluster \( i \) for \( i = 1 \ldots c \). The \( u_{ki} \) stands for the membership of the \( k \)-th point of the \( i \)-th cluster. The idea is that the closer the data point is to the cluster centre, the larger is its membership value towards that specific cluster. Consequently, the sum of all memberships of the data point across all clusters is equal to one. The fuzzy membership is formally given by the following constraint

\[
U_f = \{U = (u_{ki}) : \sum_{j=1}^c u_{kj} = 1, 1 \leq k \leq n; \\
u_{ki} \in [0, 1], 1 \leq k \leq n, 1 \leq i \leq c \}. \tag{9.1}
\]

The minimized objective function \( J(U, V) \) is defined as [10]

\[
J(U, V) = \sum_{i=1}^c \sum_{k=1}^n (u_{ki})^m d(x_k, v_i), (1 \leq m \leq \infty), \tag{9.2}
\]

where \( u_{ki} \) is a degree of membership of \( x_k \) in the cluster \( i \), and \( v_i \) represents the centre of the cluster. The parameter \( m \) is called the weighting exponent of the model. For \( m = 1 \), the memberships converge to 0 or 1, producing a crisp partitioning. The best choice for \( m \) is probably in the interval \([1.5, 2.5]\), where \( m = 2 \) is the most common choice [117]. The distance \( d(x_k, v_i) \) represents (usually) Euclidean distance between the \( k \)-th data point and the \( i \)-th cluster centre.

We should notice that the minimization of the objective function \( J(U, V) \) is not an exact minimization but an iteration procedure of so called “alternate minimization”. In essence, the algorithm is searching for a local optimal solution, which we will denote with stripe (e.g., \( \bar{U} \)). The overall iterative process may be summarised as follows.

**Algorithm Steps**

1. Initialize the matrix \( U \) by randomly generated \( u_{ki} \) membership coefficients for all cluster centres \( \bar{V} = (\bar{v}_1, \ldots, \bar{v}_c) \).
2. Find the optimal \( U \) by iteratively calculating \( \bar{U} = \arg \min_{U \in U_f} J(U, \bar{V}) \). The following solution can be derived using the Lagrange multiplier method [101]

\[
\bar{u}_{ki} = \left[ \sum_{j=1}^c \left( \frac{d(x_k, \bar{v}_j)}{d(x_k, \bar{v}_i)} \right) \pi_{ij} \right]^{-1}, (x_k \neq v_i). \tag{9.3}
\]
The solution for \( x_k = v_i \) is obviously \( \bar{u}_{ki} = 1 \).

3. Find the optimal \( V \) by calculating \( \bar{V} = \arg \min_V J(\bar{U}, V) \). The solution is computed by differentiating \( J \) with respect to \( V \) [101]:

\[
\bar{v}_i = \frac{\sum_{k=1}^{n} (\bar{u}_{ki})^m x_k}{\sum_{k=1}^{n} (\bar{u}_{ki})^m}.
\]

(9.4)

4. Repeat from step 2 until \( \bar{U} \) and \( \bar{V} \) is convergent.

The convergence is achieved when \( \max_{k,i} |\bar{u}_{ki} - u_{ki}| < \epsilon \), where \( \bar{u} \) is the new solution, \( u \) is the value from the previous iteration and \( \epsilon \) is a small positive number, the threshold. Alternatively, we can use \( \max \| \bar{v}_i - v_i \| < \epsilon \) as a convergence condition.

### 9.2 Introducing the Matching Constraint to Fuzzy C-means

In a simplified way, we can say that the original fuzzy c-means algorithm is usually based only on the pixel intensities. In our approach, we have extended this algorithm to include the matching constraints. First, by expanding the dimension of the data vector to include the disparity, and then, by evaluating the dissimilarity of the stereo pair (which will be explained later).

As stated in Section 9.1, the algorithm attempts to partition the elements with respect to a given criterion, defined as a degree of belonging that is related inversely to the distance. However, for the depth segmentation, we need to add additional components measuring the intensity (colour) difference between the point and its supposed projection and the distance between the point disparity and the disparity of its supposed cluster. In that way, we associate the clusters with the disparity space. Therefore, we have to define the vector of the cluster centre as

\[
\bar{v} = (\bar{v}_X, \bar{v}_Y, \bar{v}_I, \bar{v}_D),
\]

(9.5)

where \( \bar{v}_X, \bar{v}_Y \) stand for the spatial position, \( \bar{v}_I \) for the brightness and \( \bar{v}_D \) for the disparity value. For the clarity, the capitalized subscripts, \( X, Y, I \) and \( D \), are used to indicate the vector elements. The small subscripts will later be used to specify a particular vector from the set.

Our new membership function takes into account the dissimilarity of the left image pixel and the right image pixel shifted by the average cluster disparity. We use the disparity in the similar fashion as the intensity, grouping the pixels sharing the same, or almost the same disparity value (see Figure 9.2). For this, we need to adapt the
9.2 Introducing the Matching Constraint to Fuzzy C-Means

Figure 9.2: Illustration of the rationale behind the algorithm. The left figure shows the coloured disparity levels of the Tsukuba dataset, while the right one depicts the original pixel colours. The algorithm is based on the observation that the objects share the similar disparity, as well as similar colour. This can be clearly seen on the red lamp in the foreground or the white statue on the left. Therefore we try to form the clusters grouping the neighbourhood data points in the disparity and colour (intensity) space.

membership function to penalize the pixels having the incorrect match (not similar to their projections on the other image) and provide the way of measuring the distance between the cluster centres and pixels with associated disparity value.

We propose the use of the extended vector space model with the additional dimensions reflecting the disparity and pixel dissimilarity in the stereo image pair. The distance in the proposed vector space is, for clarity, separated into the two components \( d \) and \( d_s \), described later on. The proposed fuzzy stereo partitioning is carried out using the following membership function (the subscripts \( k, i, j \) are the indexes)

\[
\tilde{u}_{ki} = \left[ \sum_{j=1}^{c} \left( \frac{d^2(x_k, \tilde{v}_j) + d_s^2(x_k, \tilde{v}_j)}{d^2(x_k, \tilde{v}_j) + d_s^2(x_k, \tilde{v}_j)} \right) \right]^{-1}, (x_k \neq v_i). \tag{9.6}
\]

The membership \( \tilde{u}_{ki} \) is related inversely to the distance between the processed point and the cluster centre (as calculated in the previous iteration). The new term \( d_s \) reflects the correctness of the stereo match between the pixel of the left (\( \phi_L \)) and its projection on the right (\( \phi_R \)) image (the subscripts \( X, Y, D \) denotes the vector elements):

\[
d_s^2(x, v) = \lambda_m(\phi_L(x_X, x_Y) - \phi_R(x_X + v_D, x_Y))^2, \tag{9.7}
\]

where \( x \) is the data point (vector) and \( v \) is the cluster centroid. The uppercase subscript of the vector denotes its component. The constant \( \lambda_m \) stands for the weight of the
9.2 Introducing the Matching Constraint to Fuzzy C-Means

Figure 9.3: Visualisation of the data points and their clusters taken from our experiments. The points on the left figure are coloured according to the disparity levels associated with them. The right figure shows their real colour. The both figures shows the depth levels as obtained from the calculations of the modified fuzzy c-means algorithm.

The iteration steps remain the same as in Section 9.1. The outline of the algorithm can be summarized as follows: (i) choose the proper parameters, especially the number of clusters (discussed in Section 9.2.2), (ii) to each point assign random cluster membership coefficients, (iii) in each iteration compute the centroid for each cluster (Eq. 9.4), followed by the computation of the membership coefficients for all points (Eq. 9.6). Repeat this step until the algorithm has converged. Finally, create the output disparity map based on the cluster disparities (iv).

The algorithm was tested on several types of real images depicting the processed botanical samples and also on the standard dataset (described in Chapter 5). While our approach is not intended to be used as the general purpose stereo matching algorithm,
we would like to give the reader an opportunity to examine the results in standard benchmark tests (see Section 9.3) as well.

9.2.1 Stereo Matching Algorithm with Modified Fuzzy C-Means as the Cost Aggregation

The output segmentation from the previous section can be either considered as a final product or used as a support for the window based method. Because the previous algorithm is more suitable as the segmentation technique and is not intended as a general purpose stereo matching algorithm (this will be discussed with the results), we have been motivated to create a general stereo matching algorithm that uses the obtained segmentation as a form of the pixel aggregation. This is the second algorithm presented in the chapter and shows the application of the previous method. We consider this as a rather minor contribution.

The idea is simple: the advanced window based methods do not handle the pixels within the matched region with the same weight. The neighbourhood pixels, belonging to the same cluster (as the matched one) have usually higher weights for the matching than the remaining parts. To be more specific, if we think about the situation of having two different points in the window, both very similar to the matched pixel, we consider the one, which shares the segment with the matched pixel as more important. The continuation of this section shows how the pixel attributes from the modified fuzzy c-means (introduced in the previous section) can be used as the weights.

First, let us define a two-dimensional support region \( \Omega \) around the matched pixel \((x, y)\). We have to note that in the description of the fuzzy c-means algorithm, we used only one single index to access the whole set of the image pixels. In the following text, we need to maintain the pixel properties as the elements of the image grid. Therefore, we need to define an auxiliary function that will map the indexes between these two forms of notation. We assume that we already have some sort of mapping (address) function \( M(x, y) = k \), used in the fuzzy c-means algorithm, as it is necessary for the practical implementation. The function is mapping the pixel coordinates \((x, y)\) into an unique data point index \(k\) (e.g., \( M(x, y) = yw + x \), where \(w\) is the image width).

Instead of computing the matching cost using the correlation window (e.g., SAD, SSD), we compute the number of well matched pixels (the reason will be described later on). The well matched pixels are those pixels that have their intensity differences less than a given threshold \(T\) and, in the same time, lie in the same segment. The similar approach was adopted in [55], but in our method we use the weights from the fuzzy membership matrix instead of the binary weights used in the cited paper. The matching cost for the pixel \((x, y)\) and the disparity \(d\) is defined as

\[
C(x, y, d) = \sum_{\delta, \epsilon \in \Omega} P(M(x + \delta, y + \epsilon), M(x + \delta + d, y + \epsilon)), \tag{9.9}
\]
where \( \delta \in (-w/2,w/2) \) and \( \epsilon \in (-h/2,h/2) \) represents the rectangular support window. The function \( P(.) \) is analogous to the probability function. We perceive the membership function as a form of the probability function.

We will illustrate the idea on the following example. Let’s assume that the pixel \( p_1 \) is classified as the member of the cluster \( c_1 \) with the membership value close to one, meaning we are almost certain its classification is correct. Now, when comparing it with the pixel \( p_2 \), we need to find the membership value of the pixel \( p_2 \) in the cluster \( c_1 \) (to find out if it shares the same segment). If its membership value is also high, it means, that pixels \( p_1 \) and \( p_2 \) are probably sharing the same segment and the function \( P(p_1, p_2) \) returns the value comparable to the probability that \( p_1 \) and \( p_2 \) are on the same segment. In contrast, if the membership value (to the specific cluster) of one of the compared pixels is low, then the resulting value of \( P(p_1, p_2) \) is low too. The function \( P(a, b) \) can be formally expressed as

\[
P(a, b) = \begin{cases} 
(\max_i \bar{u}_{ai}) \cdot (\bar{u}_{b, \arg\max_i u_{ai}}), & \text{if } |a_i - b_i| < T \\
0, & \text{otherwise.}
\end{cases}
\]  

(9.10)

The first term \( (\max_i \bar{u}_{ai}) \) is the membership value of the first point \( a \) in its dominant cluster. The second term \( (\bar{u}_{b, \arg\max_i u_{ai}}) \) depicts the membership value of the second point \( b \) in the dominant cluster of the point \( a \).

Unlike the SAD or SSD methods, based on minimising the matching error, we are looking for the maximum value of the processed window (the cost function as defined in Eq. 9.9). The disparity \( d_c \) of the pixel \((x, y)\) can be established as

\[
d_c(x, y) = \arg\max_d C(x, y, d),
\]  

(9.11)

for \( d \in (d_{\text{min}}, d_{\text{max}}) \), the minimum and maximum allowable disparity. As we can see, the algorithm takes into account only the pixels sharing the same segment and similar colour (intensity). The intensity difference is not used as a similarity measure; therefore, the image noise has much less negative impact on the disparity estimations. This is the main reasoning for using this type of cost.

Finally, the disparity map refinement is achieved using the plane fitting algorithm. This is a very common approach in the most of the published algorithms. The reason for this step is to smoother the disparity in the segment. Because we used our modified fuzzy c-means algorithm only as a source of information of the cluster shapes (disregarding their disparities), and the algorithm described in this section obtains the disparity values \( (d_c) \) from the window area, the obtained values can be ambiguous and contain false matches. Therefore, we have decided to use the plane fitting method to limit the unwanted noise in the disparity map.

\footnote{We are aware, that the membership function and probability function are conceptually distinct, since the fuzzy truth represents membership in vaguely defined sets, but in this text we use it more as an analogy than a strict mathematical term.}
Each plane is assigned to the specific cluster of the modified fuzzy c-means algorithm (as described in Section 9.2). The assignment is obviously one-to-one. The fitted plane is defined as

\[ d_c(x, y) = a + bx + cy, \]  

(9.12)

where \( a, b, c \) represent the three plane parameters. For simplicity, we will omit the subscript \( c \) of the \( d_c \) in the following equations. For the obtained set of the cluster points \((x_1, y_1, d_1), (x_2, y_2, d_2), \ldots, (x_m, y_m, z_m)\), where \( m \geq 3 \) is the number of points of the particular cluster, we establish the best fitting plane. The plane parameters can be estimated using the least squares method, i.e.

\[ \sum_{i=1}^{m} \left| d_i - (a + bx_i + cy_i) \right|^2 = \min. \]  

(9.13)

The unknown parameters \( a, b \) and \( c \) can be obtained by solving the following linear equations:

\[ \sum_{i=1}^{m} d_i = a \sum_{i=1}^{m} 1 + b \sum_{i=1}^{m} x_i + c \sum_{i=1}^{m} y_i, \]  

(9.14)

\[ \sum_{i=1}^{m} x_id_i = a \sum_{i=1}^{m} x_i + b \sum_{i=1}^{m} x_i^2 + c \sum_{i=1}^{m} x_i y_i, \]  

(9.15)

\[ \sum_{i=1}^{m} y_id_i = a \sum_{i=1}^{m} y_i + b \sum_{i=1}^{m} x_i y_i + c \sum_{i=1}^{m} y_i^2. \]  

(9.16)

By solving these equations, we have reached the last step of the proposed approach. As you may have noticed, the last step is based on the assumption that the real space can be modelled by a set of planes; this is a common idea of many stereo matching approaches.

Summarizing, we have decided to include this section as we feel strongly obliged to the reader to present at least one general purpose algorithm in this thesis and not only the special purpose approaches. The results are promising but for us not fully satisfactory, as our algorithm falls into the last third of the Middlebury evaluation. However, we are still thinking about this concept and hope to achieve better results. We consider the idea presented in the beginning (Section 9.2) as the main contribution and this part only as a minor improvement.

The detailed results of the modified fuzzy c-means (Section 9.2) and the approach based on the aggregation windows derived from the fuzzy c-means clusters (Section 9.2.1) are described in Section 9.3.

### 9.2.2 Cluster Count Problem

The disadvantage of the fuzzy c-means (as well as k-means) is the result dependency on the initial choice of weights. This is also true for our method. Despite the algorithm
9.2 Introducing the Matching Constraint to Fuzzy C-Means

minimizes the intra-cluster variance, calculated minimum is still only a local minimum. But more serious problem of the fuzzy c-means algorithm is that it requires the number of clusters to be known in advance.

The correct choice of the cluster count is ambiguous, with interpretations depending on the shape and scale of the data point distribution in the input data set and the desired resolution. This may seem as a disadvantage for general settings, but may be an advantage for special cases, where the number of segments or number of disparity planes is already known. For example, in Figure 9.4 the box is the only object in the foreground, and can be easily represented by only a small number of segments. As you can see, with only a few clusters, we are able to acquire very precise disparity map and by increasing the number of the segments, we are able to capture even smaller changes in the disparity gradient (the box in the example is slightly tilted). We can say that by choosing the number of clusters, we can set, whether we are more interested in large segments covering the whole objects, or small fine-grained parts.

The results of the first algorithm (Section 9.2) surpass (but only for these specific types of scenes) the performance of the majority of the standard state-of-the-arts algorithms (see Table 9.1, ”Map” column). However, due to its specialization, it is less suitable for the other types of scenes.

Figure 9.4: The influence of cluster count on the output disparity map. For better reading, the segments are coloured, numbered (number in brackets) and marked with their disparity values (the value below the number in brackets).
This section describes the experiments and shows the results confirming the anticipated segmentation features and proper depth discrimination.

First, we have performed the tests of the first algorithm described in Section 9.2 (denoted as $f_z$) on the images fulfilling the assumptions, we made in the beginning – the scenes with only a few objects (each having almost the flat depth). The “Map” dataset (Figure 9.1) complies with these specifications. The results for this specific dataset are very satisfactory (Table 9.1, column “Map”); however, for the other types of image pairs they are not very encouraging. We do not consider this as a disadvantage, since the intentions of this algorithm are different and we are not going to use it in other applications than the ones presented in this thesis. The explanation for the results on the other samples is that these pairs violate the initial presumptions of our algorithm; the scenes contain a lot of objects with fine-grained disparity. The limits of our algorithm – the number of clusters and plane disparities – do not offer many opportunities for improvements in such general scenes.

To be rigorous, we have performed the tests also on the images that are not very suitable for our approach. The absolute results with the comparison of the other algorithms are shown in Table 9.1. The evaluation has been performed on the Middlebury dataset (as described in Chapter 5). While our algorithm is not the typical stereo matching algorithm, due to the lack of more suitable, generally accepted dataset, we decided to perform the tests on these images. The parameters were maintained the same for all images – cluster count $n = 200$, $\lambda_d = 1.0$, $\lambda_s = 0.1$, $\lambda_i = 0.05$, and $\lambda_m = 0.1$.

The algorithm $f_z$ converges approximately after 15 iterations on all images of the given dataset. The outputs with 100 segments are displayed in Figure 9.5 (evaluated outputs with 200 segments were not used for the illustrations, due to the hard readability of the small clusters). The images in the upper row show the segments. The disparity maps obtained from the segment properties are displayed below. As you can see, the proposed algorithm is capable of obtaining the disparity maps of more sophisticated scenes, but not at the level of detail as the generally used approaches.

To increase the overall performance, it is possible to increase the number of clusters, which in result leads to a more grained segmentation, where each segment can have different disparity. The drawback of a huge number of clusters is the increasing computational time. At the certain level, the additional increasing starts to be inefficient.

During the development, we have also performed several experiments to investigate the effects of the algorithm parameters on the segmentation performance. The parameter settings may vary from scenario to scenario, but generally, only two parameters appear to be particularly influential - the spatial and disparity weight ($\lambda_s$ and $\lambda_d$). Figure 9.6 shows the influence of these weights on the output segmentation. The experiment showed the significant effect of the disparity weight ($\lambda_d$) mainly on the images
9.3 Tests and Results

Figure 9.5: The images show the disparity and cluster maps obtained for the default Middlebury dataset using our expanded fuzzy c-means algorithm (fz). The ground truth data are provided in the last row. As you can see, the output disparity maps are not as good as the results from the "Map" dataset (Figure 9.4). The reason for that is the Middlebury dataset contains images with lot of details and a set of various objects, which contradicts the initial algorithm assumptions. In order to improve the performance it is necessary to significantly increase the number of the clusters, which consequently leads to a much longer processing time. Unfortunately, this still does not guarantee for all inputs the results comparable to the best algorithms.
containing the planar objects (e.g., the “Venus” pair). This is a predicted behaviour as our algorithm favours planar disparities. On an example of “Venus” pair, you can see that the increasing disparity weight forces the algorithm to create segments with less disparity deviations from the cluster centroid, leading to the better results. However, for images not containing such objects (e.g., “Teddy” or “Tsukuba”) the change in these parameters has only a small impact on the results. We have not studied all possible parameter configurations for all types of images, but empirically, we can say that the best results were achieved with $\lambda_s = 0.1$. Increasing this value forced the algorithm to create too compact clusters and, vice versa, decreasing caused merging too distant pixels into one cluster.

For the biological applications we have in mind, it was important to separate the layer of the base (usually the stone) and the layer above, formed by the moss structures. An example is illustrated in Figure 9.7. As you can see, the resulting segmentation strongly benefits from the inherit features of our algorithm. The design of the algorithm was strongly driven by the expected look of the captured samples, as we had a lot of samples from the whole beginning.

As for the second algorithm (described in Section 9.2.1 and denoted as $f_{zbm}$), we have performed only a limited set of experiments as we do not consider this algorithm as such important contribution of this chapter. The results are summarized in Table 9.1. In our opinion, these results are promising but not fully satisfactory, as our algorithm falls into the last third of the Middlebury evaluation. However, the aim of this algorithm was mainly to show the application of the clustering using the modified fuzzy c-means. As we do not consider this second algorithm as the principal part of this chapter we did not concentrate our efforts on the further optimizations. We consider the idea presented in the beginning Section 9.2 as the main contribution of this chapter.

In this chapter, we have analysed the recently proposed and so far not published modification of the fuzzy c-means algorithm. The fuzzy c-means algorithm is one of the most popular clustering techniques in image processing. In the past, it has been modified in many ways to take into account different constraints. In our case, we have added an additional disparity constraint and examined its impact on the segmentation performance and depth discrimination.

In the context of the image segmentation, we see the advantage of the proposed joint analysis using brightness and depth constraints. We believe, such combination improves the segmentation by creating edges not only in places where brightness changes abruptly but also in places of the depth discontinuities. Objects of the similar colour in different depths may be connected by the classical algorithm but with an additional depth constraint they are separated correctly.
Table 9.1: The performance of the modified fuzzy c-means algorithms according the Middlebury stereo test bed [125]. The overall performance is measured by the percentage of bad pixels in the non-occluded areas (nocc). The performance measured on the whole image (all) is provided as well. Our algorithms are denoted as \textit{fz} and \textit{fzbm}. In order to give a better idea of the performance of our methods compared to the state-of-the-art algorithms, we have included the results of the other algorithms from the Middlebury evaluation, including the currently best approach – agcp [127].

Figure 9.6: The \textit{fz} algorithm results achieved with different disparity and spatial weights ($\lambda_d$ and $\lambda_s$). The different disparity weights ($\lambda_d$) are represented by the different line colours. The significant effects of the disparity weight ($\lambda_d$) can be seen only on the images containing the planar objects (e.g., the “Venus” pair).
Figure 9.7: The segmentation results of the moss sample using the modified fuzzy c-means algorithm (Section 9.2). The figure shows (from left to right): the left image of the input pair depicting the moss layers on the stone base, the segments coloured according the average colour, the segments coloured according the disparity and the visualization of the clusters itself. As you can see, our modification of the fuzzy c-means still retains the properties of the original algorithm and in addition provides the disparity values.

The motivation for the main algorithm (Section 9.2) was to develop a segmentation technique that can be used for the images similar to the moss structures we have studied. Therefore, we have mainly tested and evaluated the algorithm on the datasets that resemble stone structures (e.g., the standard “Map” dataset). As for these types of images we have achieved very promising results. However, the strong a-priori assumptions limit the algorithm performance on the other types of input images. If we consider the algorithm as a general purpose algorithm, we have to say that the results are not very satisfactory. However, if we think about it as a special purpose algorithm, we can judge the algorithm as successful. The results of the segmentations of the moss layers are, according to the feedback of the biologists, most satisfactory. To sum up, we consider this segmentation algorithm (providing also the disparity maps) as a major contribution of this chapter.

Finally, the algorithm presented in the second part of this chapter, should be considered just as a small contribution. The algorithm shows, how to use the obtained segmentation as a form of the pixel aggregation. Actually, it is an application of our first method and not a completely new approach.
SUMMARY

The stereo matching has traditionally been, and continues to be, a method used mainly in the field of robotics and computer vision. But apart from this, we have shown that there are other domains that can benefit from the advances in the stereo vision as well, for example biology. We have contributed to the both areas: by the newly developed algorithms to the area of computer vision and, moreover, enriched the biological research of bryophytes by inventing the three-dimensional scanning device used for studying the surface structural changes.

In the first part of this thesis we introduced the main concepts of the stereo reconstruction and its fundamental problem – the stereo matching. The algorithm review covered the principles and methods of the state-of-the-art approaches. We tried to give a comprehensive description of their advantages, problems, and limitations. We have carefully selected those algorithms that are, in our opinion, the most important and fundamental.

Our own contributions and novel approaches were described in the second half of the thesis. The second part starts with the application of the stereo vision in the bioscience. More specifically, Chapter 6 is devoted to the use of stereo vision in the field of biology, in an environmental study of the bryophytes. Based on the requests and requirements of the phytologists, we developed a specialized scanning device for the three-dimensional reconstruction of the biological samples. The proposed construction is used for exploring the formation, structure and changes of small bryophytic plants. The capabilities and operational ease of the newly constructed device surpassed all previous methods.

In addition to this, we published a method for increasing the positioning accuracy of the step motor used in the construction. An improved version of the driver’s microcontroller firmware is able to perform the position corrections without the necessity of a feedback loop. While it may be argued that this topic is out of scope, it is our opinion that it should be mentioned as well, as it is a crucial part of the developed device.

Those, more technical sections, were followed by the description of three novel algorithms. Part of our inspiration came from the study of several segmentation techniques we were working on. In contrast to the most stereo matching approaches, we decided to use a top-down approach rather than the more common bottom-up approach. In such way, we obtain the objects first, while their spatial depth is determined later. The presented algorithms either try to use the outputs from the segmentations as the matching features or separate the objects based on the different spatial depth during the segmentation process.

The first introduced algorithm separates the planar objects of arbitrary shapes lying in a certain height above a background (base plane) using the data from the stereo image...
pair. The segmentation is solved as a minimisation problem using the Euler-Lagrange equation and the level-set function. We had to modify the original functional and add completely new terms. The new terms are based on evaluating the mismatches between the images. We have found out that the mismatches are stronger clues for driving the propagation of the segments borders than the standard matching criteria. While the segmentation into two planes may seem very limiting for the practical use, we have discovered that there are many applications meeting this constraint, e.g., the layers of the bryophyte canopies or, more common, cars on parking places. Using this approach, we have achieved outstanding results especially when processing extremely noisy images. Where other methods failed due to noise, our algorithm was still able to detect some parts of the objects.

The second presented algorithm is based on the mean shift attractor matching. The novelty of our approach lies in considering the attractor as the matching feature, reflecting some of the segment properties. We showed that the matching of attractors can be done in the similar way as the pixel matching. In other words, in our algorithm we use the attractors as a form of aggregated information rather than depending on the original image pixels. Our experiments showed that the proposed attractor matching provides fair results on general images and is especially advantageous for scenes containing large homogeneous areas (e.g., the blocks of buildings).

The last contribution was the adaptation and extension of the popular fuzzy c-means clustering algorithm. The new modification adds a support for the disparity space. The algorithm was adjusted, so that the creation of clusters is not only driven by a spatial or colour proximity of the samples, but also by a degree of stereo match. An attractive aspect of our method is that we are able to take advantage of known number of depth levels or objects (if this information is available).

Besides the achievements in computer science and information technology, we hope we have made a small contribution to the area of environmental protection and bryophyte research. We are very pleased that the scanning device is still in use since its first operation in 2008.

The list of all publications related to the topics covered in this thesis can be found in the Appendix.

10.1 Future Work

This work introduced several new features and enhancements to the traditional methods used in the image processing. We tried to get the maximum out of our methods, but still there is a lot of things to do.

From our current perspective, we see a lot of potential in the exploitation of the mean shift attractor space. The algorithm proposed in Chapter 8 should be further investigated in order to ensure the better attractor stability in the presence of noise and in
finding more suitable metrics for the attractor matching. The proposed matching criterion in the modified fuzzy c-means algorithm (in Chapter 9) also leaves some space for improvements. It should be possible to incorporate additional constraints, e.g. smoothness term, to improve the final outputs.

Since some of our algorithms are computationally expensive, a part of our future work should be focused on the optimizations. The graphic processing units (GPGPUs) are gaining in popularity in high performance computing and we see them as a natural target for the algorithm optimizations.
APPENDIX

11.1 AUTHORS BIBLIOGRAPHY

The following list of publications summarizes the author’s activities from September 2006 to July 2013. The publications in bold are directly related to the thesis topic.

Journal articles listed in ISI Web of Knowledge


Other journals articles (not listed in ISI Web of Knowledge)


Conference papers listed in ISI Web of Knowledge and Scopus


11.1 Authors Bibliography


Other publications


11.1 Authors Bibliography


*Technical Reports*


BIBLIOGRAPHY


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


[129] P. Strandmark and F. Kahl. Parallel and distributed graph cuts by dual decomposi-


[136] O. Veksler. Stereo correspondence by dynamic programming on a tree. In Pro-

[137] V. Vineet and P. J. Narayanan. CUDA cuts: Fast graph cuts on the GPU. In Pro-
cedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, volume 0, pages 1–8, 2008.


Bibliography