A comprehensive Approach for Evaluation of Stereo Correspondence Solutions in Outdoor Augmented Reality

by

© Baharehsadat Pourazar

A thesis submitted to the

School of Graduate Studies

in partial fulfilment of the

requirements for the degree of

Master of Sciences

Department of Computer Science

Memorial University of Newfoundland

April 2014

St. John's Newfoundland

Abstract

For many years, researchers have made great contributions in the fields of augmented reality (AR) and stereo vision. One of the most studied aspects of stereo vision since the 1980s has been *Stereo Correspondence*, which is the problem of finding the corresponding pixels in stereo images, and therefore, building a disparity map. As a result, many methods have been proposed and implemented to properly address this problem. Due to the emergence of different techniques to solve the problem of stereo correspondence, having an evaluation scheme to assess these solutions is essential. Over the past few years, different evaluation schemes have been proposed by researchers in the field to provide a testbed for assessment of the solutions based on specific criteria. Middlebury Stereo and Kitti Stereo benchmarks are two of the most popular and widely used evaluation systems through which a solution can be evaluated and compared to others. However, both of these models take a general approach towards evaluating the methods, that is, they have not been designed with an eye to the particular target application. In our proposed approach, steps are taken towards an evaluation design based on the potential applications of stereo methods, which enables us to better define the criteria for *efficiency*, that is, the processing time, and the required *accuracy* of the disparity results. Since AR has attracted more attention in the past few years, the evaluation scheme proposed in this research is designed based on outdoor AR applications which can take advantage of stereo vision techniques to obtain a depth map of the surrounding environment. This map can then be used to integrate virtual objects in the scene that respect the occlusion effects that are expected to occur based on the depth of the real objects.

Acknowledgements

Foremost, I would like to express my sincere gratitude to my supervisor, Dr. Oscar Meruvia-Pastor, for his continuous support, motivation, and kind guidance throughout my research program. This thesis would have not been possible without the generous resources provided by him, the Department of Computer Science, and Memorial University.

I would like to express my heartfelt appreciation to Dr. Sam Bromley for his patience, constant encouragement, and insightful, challenging comments which compelled me to always do better.

I am deeply grateful to my family who never stopped supporting me throughout all my studies, even when we were far from each other. I would have never been the person I am today if it was not for their unconditional support.

Last but not the least, I want to thank my fellow labmates in Computer Vision Lab for the interesting ideas we shared, for all the sleepless nights we spent together before deadlines, and for all the fun we had together in the last two years. Studying at Memorial University along with my peers in the department has been a great experience for me and is absolutely an unforgettable memory.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Augmented reality (AR) systems combine standard video inputs with computer-generated objects and usually provide real-time interaction for the users. In general, an augmented reality system can be defined with the following properties [1, 34] :

- Combination of real and virtual environment
- Registration (alignment) of real and virtual objects
- Real-time interaction

This concept was pioneered in the 1960s by an American computer scientist named Ivan Sutherland who created the first head-mounted augmented reality system with the help of one of his students [1].

Combining virtual objects and annotations with real world scenes has proved to be an effective way of conveying information about the surrounding environment to the user and can be useful in many applications such as gaming, medical surgeries, tourism, and other entertaining, informative or instructional tasks.

Many mobile augmented reality systems have been built over the past decades, from the Touring Machine in 1997 by Feiner et al. [11] to Google AR glasses which was announced in 2013 [24]; however, most of these prototypes have remained experimental due to certain difficulties and constraints of using them in practical applications [10, 29]. To name two of the most important constraints we can refer to:

1. Human factors in augmented reality

2. High demand of computational resources in order to provide a real-time interaction between the user and the system

1.1 Image Registration in Augmented Reality

AR systems overlay 2D or 3D virtual objects on real scenes. Therefore, depending on the application, certain accuracy is required for registration of the virtual and real objects in the scene, for which certain knowledge of the location of the user and different objects is essential [1, 34]. In an AR system, different techniques can be used to obtain the user's location and the position of other objects in the environment. In many AR systems, fiducial markers are used in the environment with computer vision tracking methods to find the actual position of the objects within the scene. This method, however, is more useful in prepared, indoor AR environments. Tracking sensors such as gyroscope and accelerometer along with video sensors can also be used as complementary techniques to provide information on the user's position and viewing orientation [1]. However, for unprepared, outdoor environments, especially in mobile AR applications, it is not practical to use markers in various locations in the scene

and, therefore, a markerless technique, such as obtaining a dense depth map of the surrounding environment, must be considered as an alternative to find the position of the objects in the scene. To obtain the depth of the surrounding environment several depth sensing technologies can be used such as 3D laser scanner, depth cameras or regular cameras. However, in order to have a mobile AR system that is easy to carry around, the weight of the whole system will be more of a concern, hence, 3D laser scanners and depth cameras are not proper choices for such systems. Depth cameras, such as Kinect, or DS325 have a strong limitation in the viewing range; Kinect, 0.8m to 4m [33]; and DS325, 0.15m to 1m [43]. On the other hand, 3D laser scanners can generate very accurate depth maps; however, they are normally expensive and their price ranges from $3,000 to $300,000 or more, depending on their accuracy and range. Therefore, among all these technologies, using several cameras to generate a depth map of the surrounding environment seems to be a more practical approach for outdoor mobile augmented reality systems.

However, using several cameras to get the depth map of the scene requires certain conditions to be met, geometrically and computationally. Many researchers have already looked into this particular problem, i.e, finding the 3D position of the points in the scene from two or multiple views using regular cameras [46]. Attempts of these researchers have resulted in certain techniques in computer vision to find the depth of different points in an environment using one or more stereo pairs taken from slightly different points of view of the same scene. These techniques are known as *Stereo Correspondence* or *Stereo Matching* in computer vision [46]. Stereo matching has been one of the most studied subjects in computer vision for many years now and there are many solutions proposed by researchers to address this problem using different

techniques; however, finding the corresponding pixels in stereo pairs with certain level of accuracy and in real-time for practical applications still remains a challenging task.

1.2 Motivation

Due to the emergence of different techniques to solve the problem of stereo correspondence, having an evaluation scheme to assess these solutions is essential. Over the past few years, different evaluation schemes have been proposed by researchers in the field to provide a testbed for assessment of the solutions based on specific criteria. For instance, the Middlebury Stereo [40] and the Kitti Stereo benchmarks [15] are two of the most popular and widely used evaluation systems through which a solution can be evaluated and compared to others. However, both of these models take a general approach towards evaluating the methods, that is, they have not been designed with an eye to the particular target application. In other words, they mainly focus on the fundamental aspects of designing a stereo algorithm as a solution per se to generally find the *best matches* of corresponding pixels in stereo pairs.

In this study, we take steps towards an evaluation design which is based on the potential applications of stereo vision methods. This enables us to better define and adjust the criteria for *efficiency* and *the best correspondence matches* while doing the evaluation. Since AR has attracted more attention in the past few years, the evaluation scheme proposed in this study is designed based on outdoor AR applications which take advantage of stereo vision techniques to obtain a depth map of the surrounding environment. This map will then be used to integrate virtual objects in

the scene that respect the occlusion property and the depth of the real objects in the scene.

In other words, our motivation in this research is to study the possibility of combining stereo vision approaches with AR systems considering the most important constraints that AR systems normally encounter in outdoor environments. In fact, our fundamental research question is:

> **"Can the combination of stereo matching techniques with augmented reality meet the requirements of an AR system in outdoor environments?"**

To provide an informed answer to the previous question, we believe that the following more fundamental questions need to be answered first:

> **"How does the human visual system (HVS) perceive depth?"**
>
> **"What is the standard angular disparity for the human visual system and how would it affect an AR system?"**
>
> **"How can we evaluate stereo vision in an augmented reality framework and what are the important factors we need to consider for this type of evaluation?"**
>
> **"In a combination of augmented reality with stereo vision, what is considered an *accurate* depth result?"**
>
> **"How can a three dimensional model be built from stereo images using computer vision techniques?"**

"What are the requirements to maintain an interactive augmented reality application for the user?"

To answer these questions, we have designed and implemented a testbed for evaluation of the stereo matching solutions based on specific criteria which will be thoroughly described in the following chapters.

As a starting point for our AR system, the depth map generated from two or multiple camera views will be used as the depth source to determine the position of the objects in the scene when overlaying virtual objects at different locations and depth levels in the real environment. For our research, we decided to narrow down our study to the effect of using stereo vision techniques on two of the most important constraints of an AR system mentioned earlier in this chapter: *human factors* and *real-time interaction*.

Human perception of depth can vary depending on the environment and under different circumstances. Many studies have focused on the evaluation of human perception of depth within different frameworks and in different applications, such as virtual reality and augmented reality, which have recently attracted more attention [47, 10, 29, 26, 45, 28]. These studies show that the viewer perception of depth is inversely proportional to his/her distance from the object [28, 45, 26, 29]. For instance, in [45] some experiments are designed to study and evaluate human perception of distance, which is the absolute depth of the objects from the observer, for an outdoor augmented reality application in urban settings. However, in this research we are more interested in the human perception of relative depth in stereo vision, which is

the ability to perceive and distinguish the depth of different objects relative to each other. In binocular vision, the minimum depth difference between two points that can be detected in the visual system is known as *Stereoscopic Acuity* or *Stereoacuity* [36]. More detail about this metric will be provided in the following chapters. We have investigated the standard stereoacuity in the human visual system and applied it to our evaluation in order to obtain the smallest detectable depth of objects in human binocular vision based on their distance from the observer.

Providing real-time interaction in an AR system for the user requires the processing time and update rate of the whole system to keep up ideally with the standard video frame rate, between 24fps and 30fps, or higher. However, studies show that in practice to build a reasonable interactive augmented world the processing rate should not be less than half of the video frame rate [17]. There are different approaches to speed up a system:

1. Using more advanced technology and hardware

2. Achieving a more sophisticated and efficient software design

However, having access to advanced technology and hardware is not always feasible and even the most advanced technologies have some limitation in their memory space and computational capability which may not meet the requirement for some real-time applications. Therefore, we have decided to focus more on the second approach while designing our evaluation system which also looks into one of the key properties of an AR system mentioned earlier, that is, the real-time user interaction.

One of the most important features that makes our evaluation unique and different from the others is that we have designed the evaluation process of the stereo matching

solutions with an eye to augmented reality applications in outdoor environments. In order to address the speed factor, we evaluate the results based on the requirements of providing an interactive AR system for the user. In addition, to address the constraint of computational resource, we have integrated a module in our design that focus on the evaluation of particular regions in the scene rather than the whole image. It is known that distinctive features such as *edges*, either in RGB or depth images from a scene, play an important role in many computer vision applications, such as object detection and tracking, determination of a set of reliable correspondences to build a 3D model that helps with better perception of object locations in 3D space [30, 46]. Therefore, in an augmented reality application, wrong depth results, especially in those regions, which will lead to erroneous registration of virtual and real objects, can be perceived easier by the human visual system. This can lead to poor performance of the system and possibly faulty interaction between the user and the augmented world. Figure 1.1 shows an accurate and faulty registration of objects (the lamp, the head and the synthetic triangle) in a 3D environment that results from an accurate and wrong disparity map in a specific area of depth discontinuity.

(a) Accurate registration of objects (b) Erroneous registration of objects

Figure 1.1: Integration of objects in a 3D environment

Therefore, we focus the evaluation in our model on the depth edges in the scene and their surrounding regions [29, 28]. Our hypothesis is that salient edges caused by depth discontinuities, which can also represent the object boundaries and occlusion, and their surroundings are one of the most important depth cues for the observer to perceive the depth of different objects in the scene [46]. Furthermore, the regions of depth discontinuity and occlusion are known as two of the most challenging parts in the image for the stereo correspondence algorithms [41]. Finding correct depth values in these regions can lead to a higher quality combination of the virtual and real objects in the scene, thus providing a more reasonable augmented world for the user to interact with.

1.3 Methodology

To achieve our objectives in this research, we have surveyed some of the existing approaches to solve the problem of stereo correspondence and the geometrical principles of the 3D reconstruction from stereo pairs, which will be explained more in the next chapters. In order to investigate the benefits of our proposed model, we have evaluated two sample stereo matching algorithms in our system. These algorithms are:

1. Semi-global block matching, also known as SGBM, which is a modified version of the semi-global matching by Hirschmuller [18].

2. Our implementation of the solution proposed by Mei et al., "On building an accurate stereo matching system on graphics hardware" [32], also known as ADCensus.

SGBM is now integrated in the Open Source Computer Vision Library (OpenCV) [25] and, therefore, we have used this implementation in our evaluation. On the other hand, since no implementation of ADCensus is available in the public vision libraries, we have used our own implementation of it which we refer to as ADCensusB in this thesis. Although ADCensus is originally proposed as a GPU-based solution, we have used the CPU implementation of it in our evaluation, as no public GPU-implementation was also available.

SGBM is selected as it has shown to generate acceptable results within 1-2 seconds on the typical test images [18]; moreover, its integration within the OpenCV library has made its usage more common in different applications. ADCensus is also currently

ranked as one of the best solutions for solving the problem of stereo correspondence in terms of general accuracy, regardless of the running time, according to the Middlebury evaluation table [40]. In addition, AdCensus does not currently exist in the KITTI evaluation table which motivates us to evaluate our implementation of it under real world circumstances with outdoor stereo images.

1.4 Organization of Thesis

This report is organized in the following structure. We discuss a background of the related work and concepts in Chapter 2 where the geometry of stereo vision, stereo correspondence problem, and a survey of the stereo solutions will be reviewed. Moreover, we will introduce some of the key computer vision techniques employed during this research work. Chapter 3 introduces some of the most relevant concepts in binocular vision to this study. In Chapter 4, we will explain our system and its design and components in detail. Chapter 5 discusses the experiments conducted for the evaluation of the proposed system in the framework of an outdoor AR application. In Chapter 6, we discuss the shortcomings and benefits of our system based on the results from Chapter 5. Consequently, a discussion of the potential aspects for improvement and future research will also be provided.

Chapter 2

Background and Related Work

In this chapter, we introduce the relevant concepts and review the techniques used in this research.

2.1 Stereo Vision

Stereo vision is the concept of viewing a scene (object) in the real world from slightly different viewpoints at the same time which results in stereo image pairs that are used by the human visual system or computer vision techniques to convey depth in a scene.

Using computer vision techniques, it is possible to extract depth information from stereo images. This process is called *Stereo Matching* or *Stereo Correspondence* in computer vision, which in fact leads to the construction of a 3D model of a scene from two or multiple views by finding corresponding pixels and, therefore, their spatial displacement within various views of the same scene [46].

Corresponding pixels in stereo images are the ones that are originated from the

same point in the real world. In most cases, the corresponding pixels are represented by the same color emanating from the same point. However, some pixels may not have a corresponding pixel in the other image for a number of reasons. For example, pixels may have become occluded or may have appeared as the result of the change in the position of the viewpoint. As it will be seen shortly in more detail, the amount of horizontal motion of such pixels in stereo pairs, which is referred to as *disparity*, is inversely proportional to the distance from the observer, i.e., depth; however, estimation of the exact depth of the points requires some other information as well, such as the position, and the calibration data of the cameras that were used to take the pictures. While the physical and geometrical approaches to this problem are well understood by researchers in the field, the process of finding the corresponding pixels correctly, yet efficiently, and measuring the disparity to generate a dense depth map still remains a challenging task.

2.2 Epipolar Geometry

Understanding the fundamentals of the underlying geometry of stereo matching helps to better understand the principal idea behind all the methods designed to address this problem, thus facilitating the comprehension of 3D model reconstruction from stereo image pairs. Therefore, we will thoroughly describe the basic geometry of stereo matching in this section.

If we consider two cameras that are looking at a particular scene from slightly different view points, a back projection of any point in the 3D space via rays passing through each camera centre, $C^{'}$ and $C^{"}$, would result in two distinct points on each image

plane. For simplicity, we will refer to the point in space as P, and its projection on the first (left) and second (right) image planes, as $P^{'}$ and $P^{"}$ respectively.

As a result of P's back projection on the image planes, an important property will emerge between the points and the camera centres, which is coplanarity of all these points. This plane, also referred to as *epipolar plane*, passes through $P^{'}$, $P^{"}$ and the camera centres, thus intersecting each image plane. It should be noted that this property, from which the consequent properties are derived, is the building block of stereo matching methods. Let us denote the specified plane by S for further reference. Since S passes through the camera centres, it clearly traverses the line that connects two camera centres. This line, which is known as the *baseline*, intersects each image plane at a point called the *epipole*; denoted by $e^{'}$ and $e^{"}$ in Figure 2.1. Consequently, the intersection of the plane S and each of the image planes, creates a line called *epipolar line* [16]. The *epipolar line* always passes through the *epipole* in the image plane. These concepts, illustrated in Figure 2.1, constitute the important components of the stereo correspondence geometry.

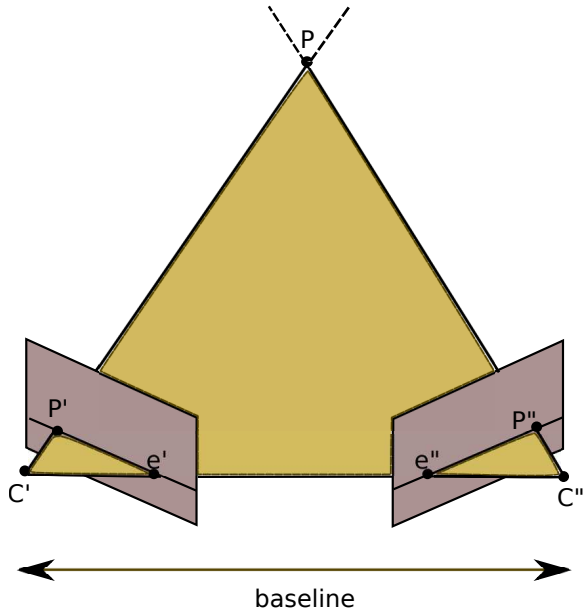


Figure 2.1: Epipolar geometry

Now, we can define the problem of stereo correspondence as a case in which the location of $P^{'}$ in the image plane is known, while the corresponding point $P^{"}$ is unknown; therefore, the problem can be stated as an attempt to find the correspondence of $P^{'}$ in the second image plane. Based on the aforementioned properties, we know that $P^{"}$ is located somewhere on the line, the epipolar line, created by the intersection of the plane traversing the ray that goes through $P^{'}$ and the first camera centre and the *baseline*. This line is in fact, the projection of the ray going through $P^{'}$ and the first camera centre, on the right image plane. Therefore, the search for the corresponding point, $P^{"}$, will be limited to merely scanning the corresponding epipolar line on the second image plane rather than the whole image.

It is now apparent that in order to find the correspondence of a particular point $P^{'}$, in the second image plane the corresponding epipolar line, must first be sought. The

projection from a point to its corresponding epipolar line can be obtained through certain transformations in space; normally a rotation and translation, Figure 2.2. For further geometrical calculations, these transformations can be represented with a matrix that is only dependent on the camera's properties, not the scene [16]. However, dealing with these transformations while looking for the corresponding points, can increase the complexity of stereo matching algorithms to certain levels [46]; therefore, in order to avoid this issue, many stereo matching approaches are proposed based on the assumption that image pairs are first warped [46]. This process is known as *image rectification* which is basically achieved by first having the cameras rotated in a way that their optical axis, the line passing through the camera centre which is perpendicular to the image plane, are parallel to each other, that is, their optical axis is perpendicular to the baseline. As a result of this transformation, the epipoles are sent to infinity. Furthermore, it might be necessary to have the cameras tilted so that their y axis also becomes perpendicular to the optical axis. After these two steps, corresponding epipolar lines actually become horizontal scanlines; Figure 2.2. This pre-processing step significantly constrains the process of searching for corresponding points and eliminates certain complications in stereo matching algorithms [46].

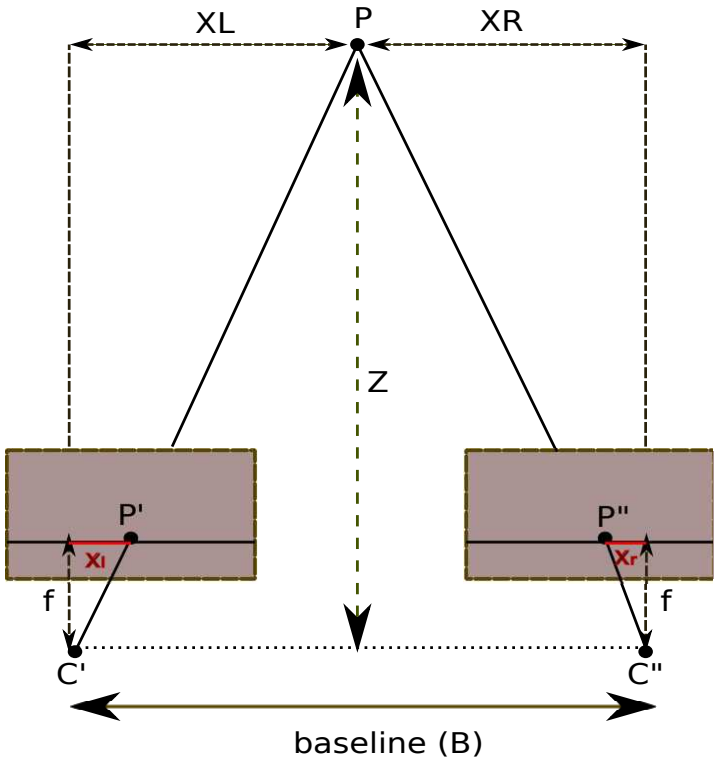
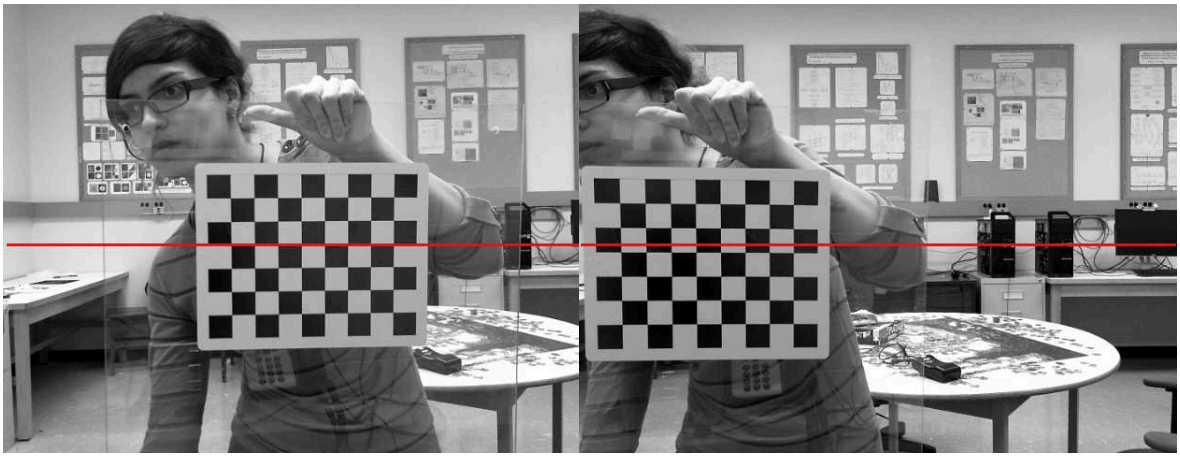


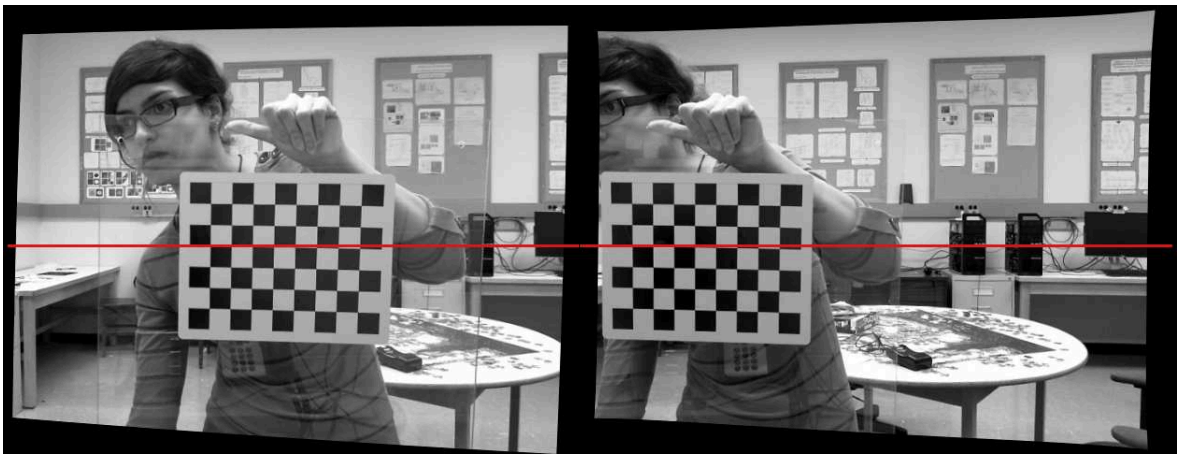
Figure 2.2: Rectified image pairs and disparity geometry

Sample stereo images taken using two regular webcams are displayed in Figures 2.3 and 2.4, before and after rectification. The red line shows how features get aligned in the left and right image after the rectification process.



(a) Left image unrectified (b) Right image unrectified

Figure 2.3: Sample stereo image before rectification



(a) Left image rectified (b) Right image rectified

Figure 2.4: Sample stereo image after rectification

Using the rectification model and epipolar geometry described earlier, derivation of the geometrical relation through which the depth of a certain point in 3D space can be obtained, will be straightforward [46]. This relation is presented as follows:

$$\frac{X_L}{Z} = \frac{x_l}{f} \tag{2.1}$$

$$\frac{-X_R}{Z} = -\frac{x_r}{f} \Rightarrow \frac{B - X_L}{Z} = -\frac{x_r}{f} \tag{2.2}$$

$$(2.1) + (2.2) \Longrightarrow \frac{B}{Z} = \frac{x_l - x_r}{f} \tag{2.3}$$

if $x_l - x_r = d$, we will have:

$$d = \frac{Bf}{Z} \tag{2.4}$$

where f is the *focal length* measured in pixels, B is the *baseline*, Z is the *3D depth*, and d is the *disparity*. The relationship between corresponding pixels in the left and right images according to disparity d is also as follows:

$$P_x^{"} = P_x^{'} + d(x, y) \tag{2.5}$$

$$P_y^{"} = P_y^{'} \tag{2.6}$$

where $d(x, y)$ is the disparity function dependent on variables x and y that can be chosen based on the coordinate of the pixel from which the displacement is calculated. This function, in fact, indicates the transformation between the corresponding pixels in the stereo images and if stereo images are aligned, then the function only indicates horizontal displacement of corresponding pixels between the two images. Therefore, based on the aforementioned formulas, the depth of points in 3D space can be easily calculated after finding the corresponding pixels in multiple views and consequently their disparities [3, 35, 41].

2.3 Stereo Correspondence Algorithms

A survey of the field shows that the algorithms which address stereo correspondence problem can be roughly divided into two main classes [41]. These classifications are commonly known as:

1. Sparse Correspondence Algorithms
2. Dense Correspondence Algorithms

Regardless of the category, stereo matching algorithms normally include specific steps in the process of finding the corresponding pixels in the stereo images. According to the taxonomy by Scharstein et al. these steps are as follows:

1. Calculation of matching cost
2. Aggregating the costs
3. Disparity computation
4. Disparity refinement

Depending on each algorithm, these steps and their sequence may change.
In this section, we are going to briefly describe the important specifications of the algorithms belonging to each of these two categories.

2.3.1 Sparse Correspondence Algorithms

Sparse correspondence algorithms, also known as feature-based algorithms, are the early stereo matching methods. In the 1980s, this class of algorithms received considerable attention by many researchers in computer vision [9]. In this type of methods,

particular features in an image, such as edges, points, line segments, or other distinctive features are extracted; therefore, the search for corresponding pixels is only applied to these regions. Consequently, algorithms of this sort result in a sparse disparity map [31, 23, 46]. The introduction of feature-based algorithms has mainly been motivated by three important factors [7, 46]:

- Lack of advanced hardware and technology for exhaustive computational tasks.
- Constraint of the search area in order to find more reliable matches.
- Stability of particular features to look for correspondences under certain circumstances when the image pairs are affected by external factors, such as illumination variations; in other words, when there is a considerable difference in photometric properties between the images, particular features, such as the edges, may be more reliable to start the correspondence search.

However, the requirement of having dense depth maps for many applications and also the emergence of efficient *dense correspondence algorithms*, have diverted the attention away from this class of algorithms in the last 20 years.

2.3.2 Dense Correspondence Algorithms

Unlike feature-based methods, dense correspondence algorithms try to find the correspondences for all the pixels in the image and, therefore, result in a dense disparity map. Most recent algorithms and studies have focused on this class of algorithms since many applications nowadays, such as graphical rendering, 3D model construction, or augmented reality require a dense depth map of the scene. However, these

algorithms face many challenges that need to be properly addressed, such as finding the depth values in occluded regions, depth discontinuities, and textureless areas [41, 7].

Dense correspondence algorithms are usually classified in two groups based on how they assign disparities to pixels [46]:

1. Local approaches
2. Global approaches

2.3.2.1 Local Approaches

Local methods tend to find the disparity of each pixel based on its neighboring pixels. In other words, the disparity of a pixel is calculated in a finite window containing its neighboring pixels, based on a particular metric, e.g. the intensity values [41].

These methods make an implicit smoothness assumption for the pixels in the search window and, therefore, assign the same disparity to all the pixels belonging to the same window which could result in incorrect disparity values in slanted surfaces or depth discontinuities [19]. This assumption can be considered as one of the major drawbacks of local methods. Another drawback of local methods is their dependency on the window size [41]. A fixed window size can raise certain problems in these algorithms:

1. If too large of a window size is considered, due to aforementioned smoothness assumption, the algorithm may result in blurry object boundaries and inaccuracy near depth discontinuities.

2. If the selected window size is too small, the disparity values will be less accurate and harder to find since little information has been considered for finding the correspondences of pixels in the image.

However, a significant advantage of using local approaches is their high speed in finding disparity results.

2.3.2.2 Global Approaches

Unlike local approaches, in global methods the disparity of a pixel depends on the information in the whole image. Global methods usually include an optimization step of a global energy function[38, 2, 6, 21]. In this class of algorithms, an optimal disparity value for each pixel is sought that leads to minimization of a global cost function that normally combines a data term with an explicit smoothness assumption.

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d) \tag{2.7}$$

The term E_{data} is normally defined as the difference of a common metric, e.g. the photometric property, between the corresponding pixels and is denoted as follows:

$$E_{data}(d) = \sum_{(x,y)} C(x, y, d(x, y)) \tag{2.8}$$

where C is a matching cost. The matching cost function can have various definitions depending on the algorithm; however, as mentioned above, it is normally defined as sum of absolute difference between the intensity of the corresponding pixels in two images [41].

The term, E_{smooth}, is the smoothness assumption based on which the disparity values

in different regions are refined. The definition of this term can also vary in different solutions. λ is also a weighting factor, by which the effect of the smoothness assumption in the global function can be controlled in the algorithm [46]. In order to find the minimum of the global function, certain approaches in computer science have proved to be particularly useful. To name some of these approaches, we can refer to dynamic programming [27], graph cut [5, 6, 4], and belief propagation [44]. Many researchers have studied and addressed the problem of stereo matching by applying one of these approaches.

The major drawback of global approaches is normally their high usage of computational resources and low speed. However, they usually result in more accurate disparity values [19, 46].

It is also worthwhile to mention that in the past twelve years, another group of algorithms have emerged which cannot be explicitly classified in any of the previously mentioned groups. These methods, which are known as *Segmentation-based techniques*, first segment the image into regions and then, rather than searching for correspondences per pixel, they attempt to find the corresponding disparity for each region. A more detailed review of these methods can be found in chapter 11 of [46].

2.4 Edge Detection

As mentioned earlier in the "Introduction", salient *edges* in the scene are one of the important features that can be used in many applications, such as object detection, image stitching, or 3D model reconstruction. Due to their importance and application in this research, we review some of the relevant concepts and techniques to edge

detection in computer vision in the following section.

2.4.1 Edges

When looking at a scene, an edge is defined whenever the visual system can perceive a distinguishable variation in color, intensity or texture between different regions [46]. Therefore, a reasonable mathematical approach to detect the edges in an image would be calculating the gradient of the intensity image and then looking for the maximum values. Another alternative would be getting the second derivative of the image and then looking for zero values. However, since an image is normally affected by a certain amount of noise which intensifies at higher frequencies, taking the derivatives of the image can lead to significant noise amplification, as it makes high frequency signals more prominent to others. Therefore, it is better to attenuate high frequencies prior to applying any edge detection approach. There are a variety of filters for image smoothing (blurring); however, since we want to attenuate high frequencies, it is better to use a low-pass filter, which only passes low frequencies. A widely known class of image blurring filters in computer vision are called *linear filters*, whose output is a linear function of their input. In linear filtering operators, for each pixel a weighted summation of its neighboring pixels is used in order to estimate its final value [46]. In mathematics, this process can be modelled by convolution of the input signal with a particular function, known as kernel.

$$g(i,j) = \sum_{k,l} f(i+k, j+l)h(k,l) \tag{2.9}$$

which is denoted as:

$$g = f \bigotimes h \tag{2.10}$$

where f and g are the input and output signals respectively, and h is the kernel function which varies depending on the type of filter.

Therefore, each filter can modify the input signal differently based on its corresponding kernel function. The Gaussian filter is a filter commonly used for attenuating higher frequencies in an image and filtering out the noise [48]. Since edges in an image may be oriented along any arbitrary direction, applying a filter which is biased towards a particular direction in filtering out the noise, would not be a prudent decision. Instead, a better choice would be choosing a filter with a circularly symmetric, i.e. isotropic, 2D kernel function such as the Gaussian filter, which is normally used in most edge detection algorithms as a pre-processing step. The Gaussian kernel has the following form [46]:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.11}$$

where σ is the standard deviation. Figure 2.5 shows the Gaussian filter that has been applied to the right half of a sample image.

Figure 2.5: Gaussian filter with kernel size of 5px (on the right side)

A more thorough description of the Gaussian and some other types of filters can be found in chapter 3 of [46].

After smoothing the image with the Gaussian filter, the gradient of the smoothed image should be taken in order to detect the edges. This can be done by convolving the signal with a pair of convolution masks in each direction in order to detect the edges, both horizontally and vertically. An edge extracting operator called *Sobel* is normally used for this purpose [42]. Sobel convolution kernels for both x and y directions are defined as follows [42]:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} \tag{2.12}$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} \tag{2.13}$$

Following the estimation of image gradients in each direction, the magnitude and the direction of an edge element can then be found by [46]:

$$|G| = \sqrt{G_x^2 + G_y^2} \tag{2.14}$$

$$\theta = \arctan(\frac{G_y}{G_x}) \tag{2.15}$$

The process of applying the Sobel operator mask to the smoothed image, is in fact equivalent to getting the first or second order directional derivative of the smoothed image and then looking for the maximum values or the zero values, respectively [46]. As a result of this process, edge elements are detected throughout the image. After finding the edge points, the next step would involve moving along the edge direction and suppressing, setting to zero, any point which is not an edge; a pixel with the gradient less than a specified threshold is a non-edge element.

The *Canny* edge detector, proposed by John F. Canny in 1986 [8], is one of the most commonly used edge detection approaches. In addition to the process described above for detecting the edges in the image, two different thresholds are defined in the Canny edge detection. This additional property makes Canny preferable to other techniques since it reduces the error in edge detection, that is, it attempts not to miss any edges in the image and not to mistakenly label any non-edge elements as edges. The purpose of having the two specified thresholds is, in fact, the elimination of streaks, which are the breakage along an edge contour. If an image is affected by certain amount of noise, the operator outputs values that fluctuate around the single defined threshold, thus causing many streaks along an edge. However, by using

two thresholds in the process of edge detection, any value above the higher threshold will be output as an edge element and while inspecting other pixels along the edge direction, only those values above the lower threshold will be accepted as neighboring edge elements. This process has shown to reduce the streaking effect to a significant amount [8].

Figure 2.7 shows the detected edges obtained by applying the canny operation on the image in Figure 2.6.

Figure 2.6: Disparity image

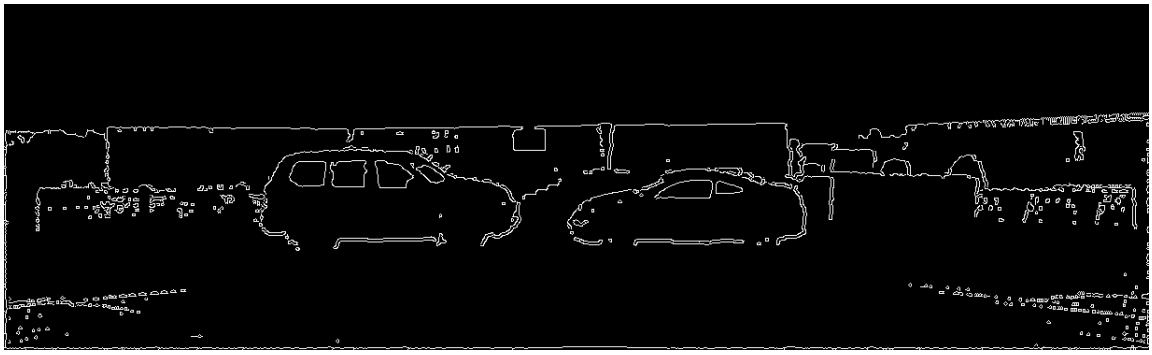


Figure 2.7: Canny edge detection on Figure 2.6

2.5 Morphological Operations

In addition to linear filters, there is another type of filters known as *non-linear filters*, whose output is a non-linear function of their input. In this type of filtering, unlike linear filters, the final value of a pixel is not necessarily a weighted combination of its neighboring pixels [46]. *Median filter, Bilateral filter*, and *anisotropic diffusion* are all different types of non-linear filters. Non-linear filters are used for certain image manipulation and enhancement tasks, and are commonly used with a particular type of image called *binary image* [46]. Binary images, as their name indicates, consist of merely two pixel values, 0 or 1. These images are usually the outcome of filtering the values in an image by a certain threshold, thus changing each value to 0 or 1 based on the comparison against the threshold. Binary images are widely used for *masking* operations in image processing [46]. Due to extensive application of binary images, certain operations are usually employed to manipulate them. These operations are known as *morphological operations* [49]. In morphological operations, the original image is convolved with a *structuring element*, also known as kernel. The structuring element is a mask (a binary image), normally smaller than the original image, with which different structures can be defined for later modification of the image. *Dilation* and *Erosion* are two of the most basic and widely used morphological operations in binary image processing. These two operations are normally used for expansion and erosion of the shapes in the original image. In dilation, the structure element which is usually in form of a circle or square with the origin located at its centre, is superimposed on top of the original binary image. By moving the structure element over the background pixels, each pixel belonging to the background, that is overlain

by the centre of the structuring element, is replaced by foreground value if at least one of the pixels of the structuring element coincide with any pixel marked as foreground. Erosion, which can be considered as the complementary operation of dilation, follows a similar process, with only the difference that the structuring element is moved over foreground pixels and any foreground pixels will be replaced by the background value if at least one pixel of the structuring element overlaps with a pixel marked as background. Hence, we can state that dilation of the foreground is equivalent to erosion of the background [49]. The dilation and erosion operations are illustrated in Figures 2.8 and 2.9, respectively.

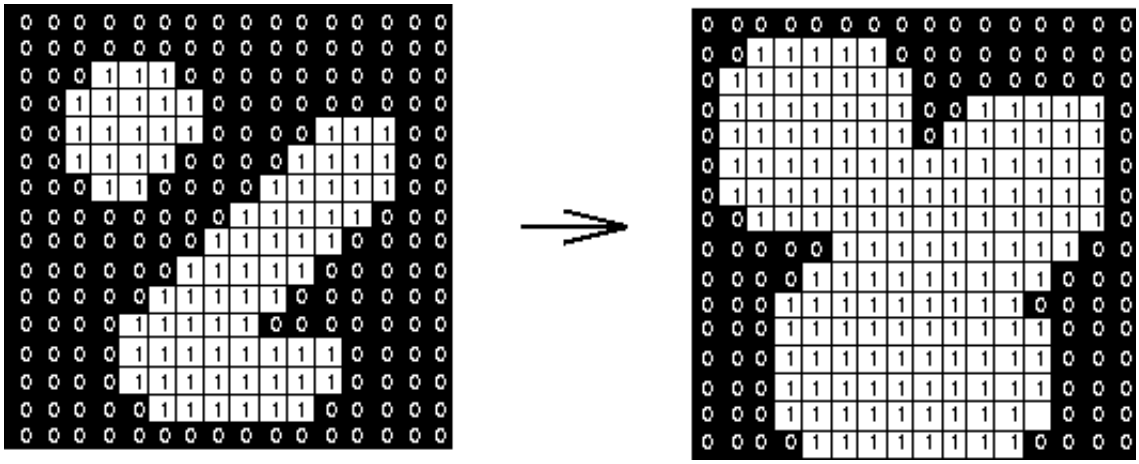


Figure 2.8: Dilation operation with a 3x3 structuring element [12]

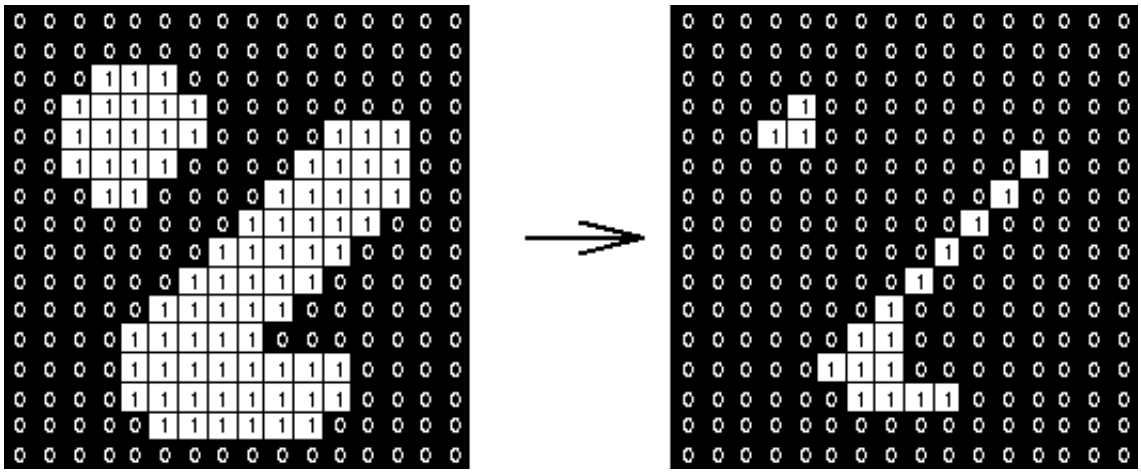


Figure 2.9: Erosion operation with a 3x3 structuring element [13]

The sample image shown in Figure 2.7 is presented below after applying dilation on the detected edges in the image to expand the detected regions.

Figure 2.10: Dilation of the detected edges in Figure 2.7 with a 10x10 structuring element

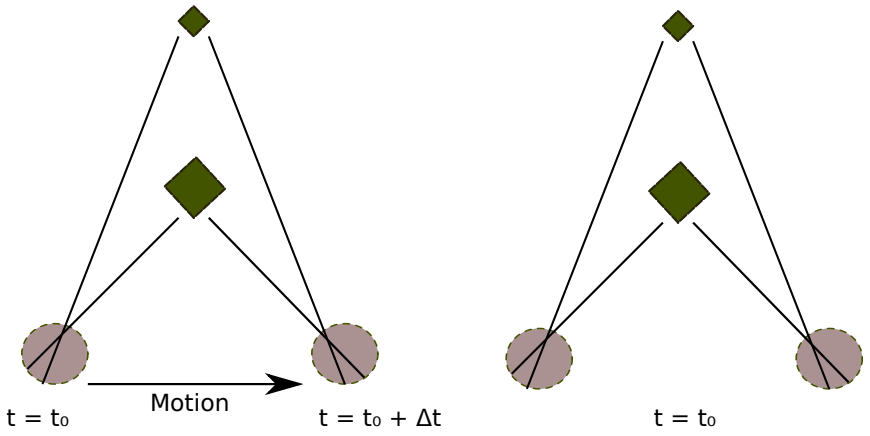
In the next chapter, we will describe the key concepts in the human visual system that are related to the perception of the depth.

Chapter 3

Binocular Vision and Stereopsis

Binocular vision is a term used for the visual system of animals with two eyes [22] and, therefore, applies to the human visual system as well. Possessing binocular vision not only leads to a better perception of depth of the surrounding environment, but also helps to better perform many visual tasks such as reading, object detection, interaction with surrounding objects such as grabbing and other manipulative tasks [22]. The most significant advantage of possessing binocular vision is its influence on how the 3D environment, that is, the depth of surrounding objects relative to each other, is perceived by the visual system. This visual perception of depth in binocular vision is referred to as *Stereoscopic Vision*. In the visual system, depth perception is a phenomenon that normally occurs though different types of cues and information existing in the surrounding environment. These pieces of information, known as *depth cues* in stereo vision, can be either monocular or binocular depth cues [22]. To name a few instances of monocular depth cues, we can refer to motion parallax, lighting and shading, and apparent size. However, as previously mentioned, binocular cues

which can only be perceived by stereo vision, play a major role in the perception of depth. One of the most important binocular cues is *binocular disparity*, or *binocular parallax*. It should be noted that the effect of binocular parallax and motion parallax on depth perception are very similar to each other. In motion parallax, which is a monocular depth cue, the scene is viewed at different times by the observer moving from one side to the other, whereas in binocular parallax, the scene is viewed from slightly different viewpoints at the same time by the visual system, while the observer is standing at a fixed position [22].



(a) Motion Parallax (b) Binocular Parallax

Figure 3.1: Motion parallax and binocular parallax difference

Binocular disparity, which in fact arises from the spatial difference between the images of the same scene in the visual system, provides a relative perception of depth from the surrounding environment. This perception is known as *binocular stereopsis* [22]. Another important binocular depth cue is the eyes *vergence*, which is the simultaneous movement of the pupils in opposite directions in order to obtain a unified

view of an object in the visual system. When focusing on an object, the optical axes of the eyes intersect on the object of interest resulting in an angle called vergence angle. The human visual system is capable of adjusting this angle based on the distance from the object [22]. In stereo vision, the locus of the points that yield a unified view of an object in the visual system is known as the *horopter*, and any point located on the horopter is usually called a *fixation point* [37, 22]. An important property of an object on the horopter is that no spatial difference exists between the images of the fixated object between the two eyes, that is, the binocular disparity is zero [22]. Exploiting this property, the disparity of any other object in the scene can be estimated relative to the fixated object by inspecting two important factors: whether the object of interest is closer or further than the fixated object and then how much closer or further it is relative to the fixated object. As a result, the binocular disparity provides a relative perception of depth of the surrounding environment. In the geometry of stereopsis, the relative disparity between two objects is usually presented as angular disparity in degrees, radians, minutes of arc (arcminute), or seconds of arc (arcseconds). The relation between these measurements is as follows:

$$1arcmin = \frac{1}{60}degree = \frac{\pi}{10800}radians \tag{3.1}$$

$$1arcsecond = \frac{1}{60}arcmin = \frac{1}{3600}degree = \frac{\pi}{648000}radians \tag{3.2}$$

3.1 Stereopsis Geometry and Angular Disparity

In the following section, we will describe how the angular disparity can be calculated utilizing the geometry of stereopsis [37].

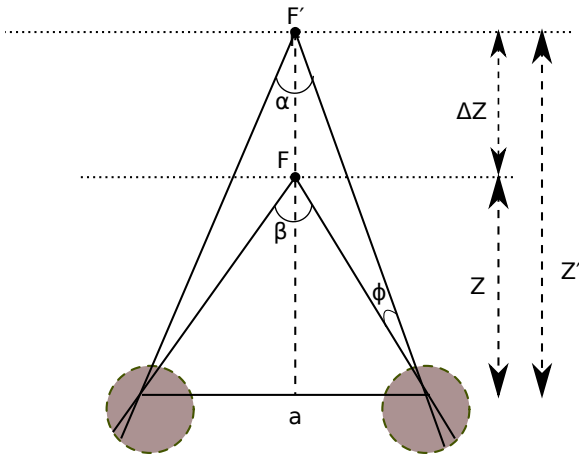


Figure 3.2: Binocular disparity

According to Figure 3.2, we have:

$$\tan\frac{\alpha}{2} = \frac{a}{2Z'} \tag{3.3}$$

$$\tan\frac{\beta}{2} = \frac{a}{2Z} \tag{3.4}$$

It is known that for small angles, when the angle approaches zero, the tangent of an angle is approximately equal to the angle in radians. Therefore, we will have the following relations:

$$\theta = 2\phi = \beta - \alpha \tag{3.5}$$

$$\frac{\alpha}{2} \approx \frac{a}{2Z'} \tag{3.6}$$

$$\frac{\beta}{2} \approx \frac{a}{2Z} \tag{3.7}$$

$$Z^{'} = Z + \Delta Z \tag{3.8}$$

$$\Rightarrow \theta \approx \frac{a}{Z} - \frac{a}{Z'} = \frac{a}{Z} - \frac{a}{Z + \Delta Z} \tag{3.9}$$

$$\Rightarrow \theta \approx \frac{aZ + a\Delta Z - aZ}{Z(Z + \Delta Z)} = \frac{a\Delta Z}{Z(Z + \Delta Z)} \tag{3.10}$$

When ΔZ is a small value compared to Z, the term ΔZ in the denominator can be neglected without significant loss of accuracy. This results in the approximate formula as follows:

$$\theta \approx \frac{a\Delta Z}{Z^2} \tag{3.11}$$

Here, a is the distance between the center of the pupils of the two eyes, which is known as interpupillary distance. It should be noted that a, Z and ΔZ must all have the same units in this formula. This equation estimates the angular disparity in radians; in order to convert θ to arcseconds, according to the conversion rules presented in Equation 3.2, it should be multiplied by:

$$\frac{648000}{\pi} = 206,265 \frac{arcsecs}{radians} \tag{3.12}$$

Studies show that the visual system capability to distinguish two objects at different depths relative to each other is limited to certain thresholds [37, 22]. This threshold, which is defined as the minimum detectable depth between two objects at difference distances, is known as *stereoacuity* which varies in different visual systems [37, 22]. According to standard stereo tests [37], the finest detectable disparity in the human visual system is approximately 10-15 arcseconds. However, a more recent study on 60 subjects [14] at different age groups, from 17 to 83 using standard stereotests, shows that the average stereoacuity for different age groups is as follows:

Table 3.1: Average stereoacuity for subjects of age 17 to 83

Age Range	**Stereoacuity (arcsecs)**
17-29	32
30-49	33.75
50-69	38.75
70-83	112.5

As can be seen, the stereoacuity for the the human visual system increases with age, that is, the amount of error in the depth results is less perceptible in the visual system of the elders than the youths. Using these values in Equation 3.11 along with the average interpupillary distance in the human visual system that is reported to be approximately 64mm [22], we can estimate the threshold for minimum detectable depth between two objects based on their distance from the observer.

We have employed the concepts introduced in this chapter in the design of our evaluation model for an augmented reality system in outdoor environments. In the next chapter, we will describe the design of our system and its components in more detail.

Chapter 4

Design of the Evaluation Scheme

This chapter walks through the steps taken in order to build our evaluation system and describes its keys components in detail.

4.1 Design Criteria

Since outdoor AR applications are the focus of this research, we have designed our evaluation model within this framework. Moreover, all the proposed evaluation metrics are measured based on the relevant factors described earlier in the previous chapters.

4.2 A Comprehensive Evaluation Scheme

In an augmented reality system, there are certain factors that would affect the functionality and effectiveness of the system [29, 28] and, therefore, should be carefully considered when designing and evaluating the system. These factors, which cor-

respond to different components of an AR system, are related to the surrounding environment, human factors in AR, or technology and hardware constraints. Figure 4.1 illustrates the key components of a high level design we propose for a stereoscopic AR system.

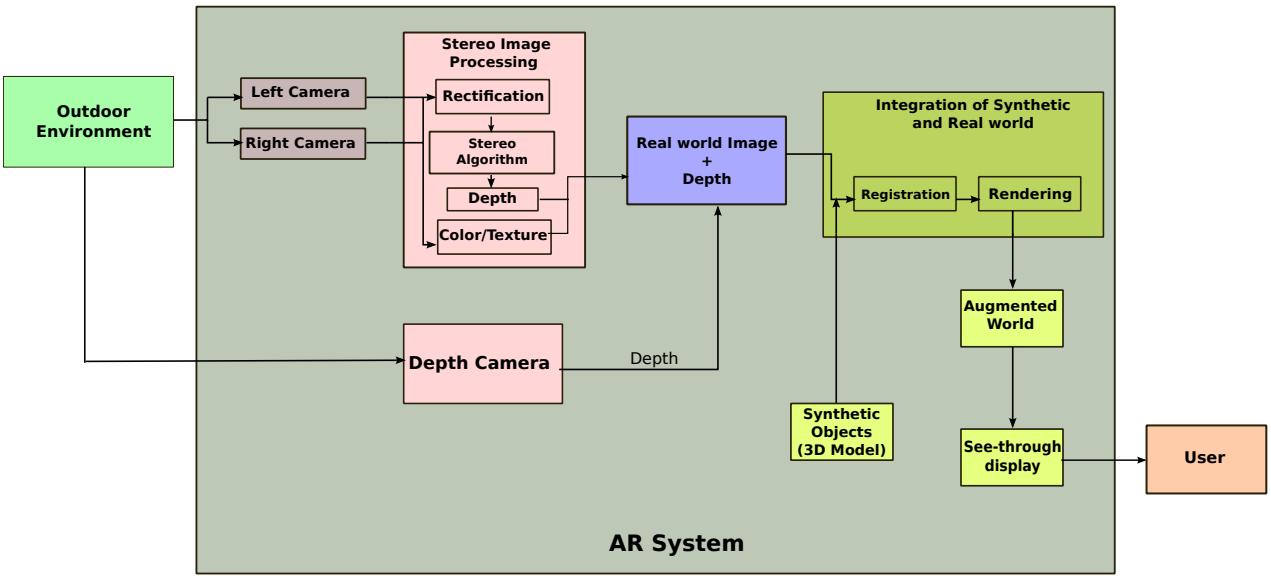


Figure 4.1: Proposed design of the high-level architecture of a stereoscopic AR system

In our design, unlike the Middlebury or Kitti benchmarks, we label a pixel in the disparity results as an *outlier* if the angular measurement, that is in form of stereoacuity, corresponding to the depth error between the ground truth and the estimated depth value by the algorithm is more than the minimum perceptible stereoacuities for the human visual system as determined by standard stereo tests [37, 14]. Moreover, we use the average stereoacuity for different age groups [14] in our design to evaluate the performance of the algorithm for users at different ages; this makes the evaluation results more reliable and applicable to practical applications of AR.

In order to evaluate the efficiency of an algorithm and investigate whether it meets the requirements for being part of a real-time AR application, we have integrated a module in the evaluation process that reports on the average execution time of the algorithm for the input data. The average outliers based on the specified stereoacuity thresholds and the average disparity error are also estimated during the evaluation process.

In addition, our model employs a particular approach which can be of specific value to practical AR applications. In this approach, we suggest that it is prudent to focus the evaluation process on the particular regions of the disparity map rather than the whole image. The main hypothesis is that salient edges caused by depth discontinuities, which also represent object boundaries and occlusion, are important depth cues for the human visual system to better perceive the location of different objects in the 3D environment [46]. Based on this hypothesis, we argue that more accurate depth results in these regions allows for a higher quality combination of the depth map of the real world with the virtual depth of the synthetic objects that are part of the AR scene.

4.3 Design Overview

Our evaluation model consists of the following key components:

- Stereo pairs, calibration data, and ground truth disparity (occluded or non-occluded) as inputs
- Edge region masks generated from the ground truth disparity maps

- Masked ground truth disparity
- Full and masked disparity maps generated by the stereo algorithm
- Main evaluation module
- Evaluation metrics output as data files and plots

It should be noted that some of these components, such as the masked ground truth, or the masked disparity maps can be optionally built during the process depending on the specific parameters set at the run time of each step. Figure 4.2 shows the high level block diagram of our design.

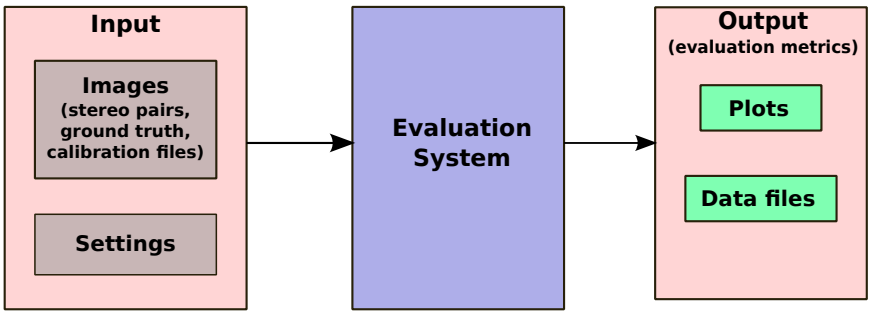


Figure 4.2: High-level block diagram of the evaluation system

A lower level architecture of our evaluation system is shown in Figure 4.3. This figure illustrates the sequence of the operations during the whole process.

As can be seen in Figure 4.3, first the input data consisting of the stereo images, the ground truth disparity, and the calibration data are passed to the system. Afterwards, the specified masks are created using a *Canny* edge detector and a *Dilation* operation with the appropriate parameters selected separately for each image. After the corresponding disparity maps have been generated by the stereo algorithm and

stored on the disk, they are passed to the evaluation module with the specified arguments. Finally, the evaluation metrics are estimated and output as data files and plots to facilitate the evaluation of the stereo algorithm in the application of interest, outdoor AR systems.

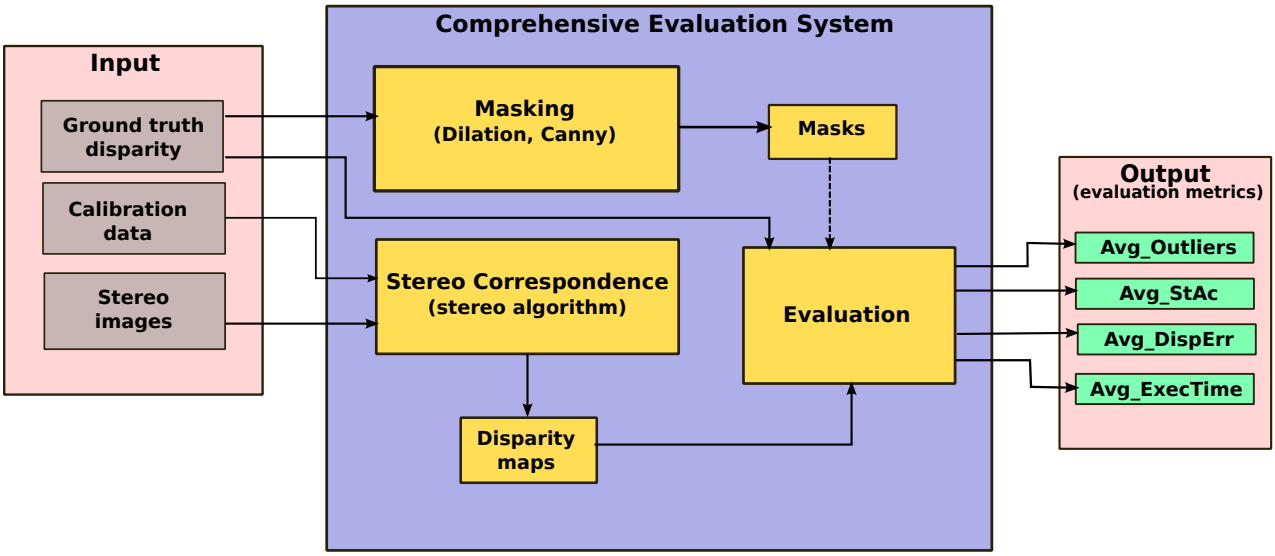


Figure 4.3: Low-level architecture of the evaluation system

4.4 Evaluation

In this section, we break down the main evaluation component to its underlying modules. We will then look at the functionality of each module in more detail.

As previously mentioned in this chapter, the results of the evaluation are presented through specific metrics which are as follows:

- The average execution time
- The average disparity error

- The average outliers
- The average stereoacuity

The analysis of these metrics in the framework of an outdoor AR application will then allow for a practical evaluation of the stereo algorithm performance. We will now explain how each of these metrics is measured in each module that builds up together the evaluation component in the system.

4.4.1 Average Execution Time

For each image pair, the time spent on generating the disparity results is estimated using the C++ function, *clock()*. This function returns the number of clock ticks elapsed since a program starts running. A division by the system-specific value *CLOCK_PER_SECOND*, the number of clock ticks in a second, converts the value returned by the *clock* function into the time consumed by the CPU in seconds. Getting the difference between the *clock* values before and after a function call results in the execution time of the particular function. We have applied this method in our implementation to estimate the execution time of the algorithm for each image pair. In the end, the mean of all the values corresponding to different image pairs is taken to obtain the average execution time of the algorithm for the input dataset.

4.4.2 Average Disparity Error

Two average disparity errors are calculated in our evaluation. One corresponds to the valid pixels in the ground truth, depending on what value is considered valid in the ground truth disparity, and the other to the valid pixels in the generated disparity

which depends on the implementation of the stereo algorithm. The valid ground truth disparity for the Kitti disparity maps, is a value greater than 0 and in the selected algorithms, SGBM and ADCensusB, values equal to or greater than 0 are considered valid. To this end, for each validity criteria, the mean difference between the ground truth disparity and the one found by the algorithm is estimated for all the pixels in the image or merely the masked pixels depending on the availability of a mask. The pseudocode for this operation is as follows:

```
ADE Estimation; START
   DispErr_total = 0;
   for all pixels p in the image:
      if (masked)
         if(!mask[p])
            continue;
         end if
      end if
      pix_count += 1;
      Disp_err = |disp_gt − disp_gen|;
      DispErr_total += Disp_err;
  end for
   Avg_DispErr = (DispErr_total)/(pix_count);
ADE Estimation; END
```

4.4.3 Average Outliers

Similar to the average disparity error, based on the validity criteria for disparity, two values are reported for this metric as a result of the evaluation. For this measurement, the relative depth error is first calculated by finding the corresponding depth values for the ground truth disparity and the disparity generated by the algorithm in Equation 2.4. This value is then compared to the relative detectable depth threshold for the human visual system that is estimated using equation 3.11. If the relative depth error is equal to or more than the detectable threshold in the human visual system, the corresponding pixel is labelled as an outlier. Since we are using four different thresholds of stereoacuity corresponding to different age groups in our evaluation, the estimated error is compared against each of these thresholds and, therefore, four different values are eventually calculated. This process is repeated for all the pixels in the image or merely the pixels in the masked regions depending on the availability of a mask. Considering the two validity criteria of pixels and the four identified age groups, eight values are reported at the end of the evaluation for the average outliers.

```
AO Estimation; START
   Total_Outliers = 0;
   for all pixels p in the image:
         if (masked)
            if(!mask[p])
               continue;
            end if
         end if
```

$pix\_count \; += \; 1;$

$depth_{gt} \; = \; \frac{focal\_length \; * \; baseline}{disp_{gt}};$

$depth_{gen} \; = \; \frac{focal\_length \; * \; baseline}{disp_{gen}};$

$depth_{err} \; = \; |depth_{gt} - depth_{gen}|;$

$stAc_{err} \; = \; \frac{pupil\_distance * depth_{err}}{depth_{gt}^2};$

`if (`$stAc_{err} \; \geq \; stAc_{threshold}$`)`

$Total\_Outliers \; += \; 1;$

`end if`

`end for`

$Avg\_Outliers \; = \; (Total\_Outliers)/(pix\_count);$

`AO Estimation; END`

4.4.4 Average Stereoacuity

The estimation of the average stereoacuity can be broken down into 3 steps:

1. Stereoacuity estimation based on the generated disparity for each image pair and the ground truth

2. Averaging the stereoacuity results over certain depth ranges in each image

3. Averaging the results from the previous step over all the images

Corresponding plots are generated after the third step based on the final results.

According to the specific age ranges, different values are reported for the average stereoacuity at the end of the evaluation. In order to estimate this metric, the depth

values corresponding to both ground truth and the generated disparity by the algorithm are first calculated using Equation 2.4. Subsequently, the difference between these values is used in Equation 3.11 to calculate the corresponding stereoacuity, as mentioned in the estimation of the average outliers. This process is done for all the pixels in the image; or if a mask has been provided, it will be only applied to the pixels in the masked areas. Finally the results are output and stored in a separate data file for each image. After conducting the first step on all the disparity maps corresponding to input image pairs, the second step starts by building a histogram of the stereoacuity values over specific depth ranges. Using the output file containing the stereoacuity values from the first step for each disparity image, the corresponding histogram is constructed by defining the number of bins and their width. In our design, the width of each bin determines the aforementioned depth range and is kept constant for all the bins. Moreover, the number of bins along with their corresponding width determine the total distance over which the results are estimated and subsequently examined, Equation 4.1.

$$Total\_distance = Number\_of\_bins * Width \tag{4.1}$$

For outdoor applications of AR, these parameters are normally set to certain values so that the total distance can cover the medium to far depth fields; extending from 1.5 meters to more than 30 meters [45]. The results of the previous step, which are all stored in a single data file, are then passed to the last step. At this point, a histogram is built over the data from all the disparity images, which results in the average stereoacuity values within each specified depth range over all the images. It should be noted that the number of bins and their corresponding width at this point,

are similar to the histogram constructed in the the previous step.

```
ASA Estimation; START
   // STEP1:
   for all images:
      for all pixels p in the image:
            if (masked)
               if(!mask[p])
                  continue;
               end if
            end if
            pix_count += 1;
            depth_gt = (focal_length * baseline)/disp_gt;
            depth_gen = (focal_length * baseline)/disp_gen;
            depth_err = |depth_gt − depth_gen|;
            stAc_err = (pupil_distance*depth_err)/depth_gt^2;
            Append(stActFile,stAc_err);
      end for
   end for
   // STEP2:
   /**Histogram over depth ranges for each image**/
   width = depth_range;
   for each stActFile:
      Avg_StAc_img = histogram.build(bins,width);
```

```
      Append(img_histFile,Avg_StAc_img);
   end for
   // STEP3:
   /**Histogram for final stereoacuity over all the images**/
   Avg_StAc = histogram.build(bins,width);
   Write(Avg_StAcFile,Avg_StAc);
   plot(Avg_StAcFile);
ASA Estimation; END
```

4.5 Platform

The evaluation system was implemented on a Linux platform, Ubuntu 12.04 distribution, with 12GB RAM and Intel Core(TM) i7 960 3.20GHz CPU. No optimizations have been used in the evaluation. We have used g++ as the compiler and C++ as the high level language for implementing the core functions within the system, such as the main evaluation function, the masking process, and the other fundamental operations that are the building blocks of the system. Furthermore, the Tool Command Language (TCL) has been used for all the scripts that wrap around the C++ functions, to facilitate and accelerate the execution of each step in the process.

In this chapter, we described the main components of our proposed model. Next, we will discuss the functionality of our system through different experiments.

Chapter 5

Evaluation

In this chapter, we will go through our experimental hypotheses, testing scenarios, experiments conducted on two sample stereo matching algorithms, SGBM and AD-CensusB, and the results with our proposed evaluation system to assess the benefits of using our evaluation model for outdoor AR applications over the general-purpose evaluation models; the Middlebury and Kitti Stereo Evaluation.

5.1 Stereo Dataset

It should be noted that the stereo images we have used to conduct the experiments on stereo algorithms in our system, are selected from the Kitti Stereo Dataset. In contrary to the Middlebury dataset, the Kitti Stereo Project provides stereo images and ground truth disparity maps that are taken from outdoor scenes under real circumstances. These properties make them more appropriate for evaluating the performance of the algorithms in outdoor AR applications, thus better meeting the objectives of this study. We have selected fifty-two image pairs from the Kitti Stereo

dataset based on different photometric and visual properties that are important in stereo vision and an AR application, as observed by the human visual system. Some of these properties are listed as follows:

- Variation in light and shading, that is, the scenes including bright, dim, and dark regions.
- Various depth ranges, that is, including near field, medium field and far field objects.
- Various degrees of depth discontinuity and occlusion, as observed in the images.
- Well textured and not properly textured regions.

5.2 Methodology

Before going through the explanation of the experiments to assess our evaluation model, we restate our main research question in this study to better justify our hypotheses and the experiments defined for their validation. As mentioned earlier in Chapter 1, our main objective is to investigate whether using stereo matching techniques to generate the depth map of the surrounding environment in an outdoor AR application can meet the requirements of the AR system. Therefore, our experiments focus on assessing those aspects of our evaluation model that assist to better answer this question. As a result, our first attempt towards evaluating our model is to investigate and demonstrate whether the results of the evaluation process are properly measured and presented in the framework of the important factors in an outdoor AR application. After confirming this property, which is the key property of our model,

we investigate the effect of our proposed masking approach on the evaluation results. Moreover, we present how the methods are evaluated in the framework of real-time interactive AR systems. We also explain how the evaluation and comparison of the methods is done in our model with some experiments on the sample stereo matching algorithms.

5.3 Hypotheses

We have defined a set of hypotheses to evaluate our proposed design. These hypotheses are as follows:

- **Hypothesis 1**: *Our model is more suitable than other approaches to evaluate and demonstrate the performance of the stereo matching algorithm in the framework of outdoor augmented reality applications.* Unlike the Middlebury and Kitti benchmarks which are considered general-purpose evaluation models, our system can particularly evaluate the algorithms in the framework of an outdoor augmented reality application to facilitate the process of determining the proper method for using in the AR system for a high quality real-time generation of the depth map of the surrounding environment from the user's point of view.

- **Hypothesis 2:** *Observing, evaluating, and consequently refining the areas near the depth edges in an image are more important in an AR application.*

  Salient edges caused by depth discontinuities, which can also represent the object boundaries and occlusion, are one of the most important depth cues that

helps the observer to better perceive the depth of different objects in the scene. In other words, the areas near the edges corresponding to depth discontinuities in a scene are more important to the human visual system for perception of depth in an AR application and, therefore, the disparity errors in these regions can be detected easier by the HVS. Therefore, we argue that in our model, which has the property of masking and evaluating the results for these particular regions, the evaluation results can be of great value to an outdoor AR application.

- **Hypothesis 3:** *Our system is better than other evaluation models for assessing the performance of the algorithm in real-time AR applications.*

  Other evaluation models, the Kitti and Middlebury benchmarks, do not evaluate and report on the efficiency of the algorithms with respect to their execution time. On the other hand, our system is capable of examining and evaluating an algorithm based on its execution time and, therefore, can report on its efficiency for real-time AR applications.

- **Hypothesis 4:** *The trade-off between the accuracy and the running time of the stereo algorithms can be effectively evaluated in the framework of an outdoor AR application through our system.*

  Nearly all the solutions to the problem of stereo correspondence have been dealing with the trade-off between the accuracy of the results and the running time. Therefore, most of the solutions focus only on improving one of these aspects in the final results. Some methods use certain post processing techniques to refine the disparity results in the end, thus improving the accuracy, whereas

the others propose particular approaches that can be implemented on the GPU to reduce the processing time. Due to the importance of both metrics in an outdoor AR application, we argue that the trade-off between these metrics can be effectively analyzed in our evaluation system.

- **Hypothesis 5:** *The ability to detect the difference in depth in stereo correspondence methods not only depends on their accuracy in estimation of the disparity values, but is also affected by other factors, such as the environmental noise, the resolution of the capturing device and its robustness to noise.*

  According to different studies [10, 28, 1], some other factors such as issues associated with the environment, display devices, or capturing devices can also affect the perception of depth in the visual system. As a result, we hypothesize that the ability to detect the difference in depth in an outdoor AR system, does not merely depend on the accuracy of the stereo correspondence algorithm, and other factors should also be taken into account.

The experiments designed to validate these hypotheses are explained in the following sections.

5.4 Experimental Environment and Settings

Experiments were carried out on a Linux machine with Intel Core(TM) i7 3.20GHz CPU. We have evaluated two sample stereo matching algorithms in our system: First, Semi-global block matching, also known as SGBM, which is a modified version of the semi-global matching by Hirschmuller [18]. Second, our implementation of the

solution proposed by Mei et al. [32], known as ADCensus.

SGBM is now integrated in the Open Source Computer Vision Library (OpenCV) [25] and, therefore, we have used this implementation in our evaluation. On the other hand, since no implementation of ADCensus was available, we have used our own implementation of it which we refer to as ADCensusB. Although the GPU-based approach to both algorithms are proposed in the literature, we have used their CPU implementations in this research, since none of the GPU implementations is publicly available. Before moving forward with other experimental settings, we provide a brief description of our implementation of the ADCensus algorithm in the following section.

5.4.1 ADCensusB Implementation

ADcensus proposed by Mei. et al. is one of the top ranked algorithms in the Middlebury benchmark [40] which is proved to efficiently generate highly accurate disparity results for Middlebury dataset. The main reason for its superior performance, in terms of both accuracy and processing time, is the combination of various computer vision techniques that can be properly mapped to GPU for acceleration and result in accurate matches of pixels in stereo images [32]. Various cost functions are estimated through these techniques which, in the end, are used for finding the corresponding disparity values. As suggested in the paper [32], our implementation of ADCensus also includes various cost estimations at different steps of the algorithm that are computed by separate functions in the code; as a result of each function call, specific arrays of type *float* corresponding to various costs are filled in and used by the subsequent functions in the algorithm. The main cost estimations are as follows:

1. **AD-Census Cost:** The initial matching cost which consists of the average intensity of pixel values in the left and right images, and the census cost that is formulated based on the relative ordering of the pixel intensities within a specific window size [20], defined as a 9×7 window in the implementation. This step is accomplished by three functions in our implementation which are named *costAD()*, *c_census()* and *initCost()*.

2. **Aggregated Cost:** The aggregated matching cost of each pixel over a specified support region, which is defined as a cross-based region in [32] and originates from the method proposed by Zhang et al. [50]. This cost is computed by a function called *aggregatecost()* in our implementation

3. **Path Cost:** The path cost that is estimated from the aggregated cost of each pixel by scanline optimization from different directions. The idea of multi-direction cost optimization originates from Hirshmuller's semi-global matching solution [18]. This step is accomplished by a function called *scanline()* in our implementation.

We have followed the approach in each of the referenced papers for the implementation of each cost estimation. As a result of the aforementioned steps a three-dimensional cost volume with the size of $image\_width \times image\_height \times disparity\_range$ is generated as the final cost of type *float*. After this step, disparity values are selected by following the "winner takes all" approach, that is, the disparity with the minimum cost is selected as the final disparity value of each pixel [41]. After the main body of the method is implemented, the algorithm proceeds with a multi-step refining process. This step, which is one of the unique features of ADCensus method,

attempts to detect the outliers, that is the wrong matches of pixels, and refines the disparity results based on the detected outliers. For the detection of outliers, we follow a common approach known as left-right (L-R) consistency check. In this check, the disparity map for both the left and right images is first calculated. Then, if a pixel in the left image, based on its disparity value, corresponds to a pixel in the right image that does not map back to it, it will be labeled as an outlier [32]. This description can be formulated as follows:

$$D_L(p) \neq D_R(p - (D_L, 0)) \tag{5.1}$$

Where $D_L(p)$ is the disparity function for the left image and D_R is the disparity function for the right image.

In our implementation of ADCensus, we have the L-R check and its subsequent refinement steps triggered with a flag. Therefore, when the flag is not set, neither the check nor the refining steps are triggered in the algorithm. Following the proposed approach in the paper, we also have multiple steps of refinement in our implementation, which occurs through different function calls. These functions are as follows:

1. ***findOutliers()***, which tends to find the outliers by conducting the L-R check.

2. ***regionVoting()***, which iteratively updates the disparity value of each outlier pixel based on a histogram of its reliable neighboring pixel values in its cross-based support region; the disparity with the most votes in the neighboring region is chosen as the final disparity value of the outlier pixel.

3. ***interpolate()***, which follows an interpolation strategy to find the remaining

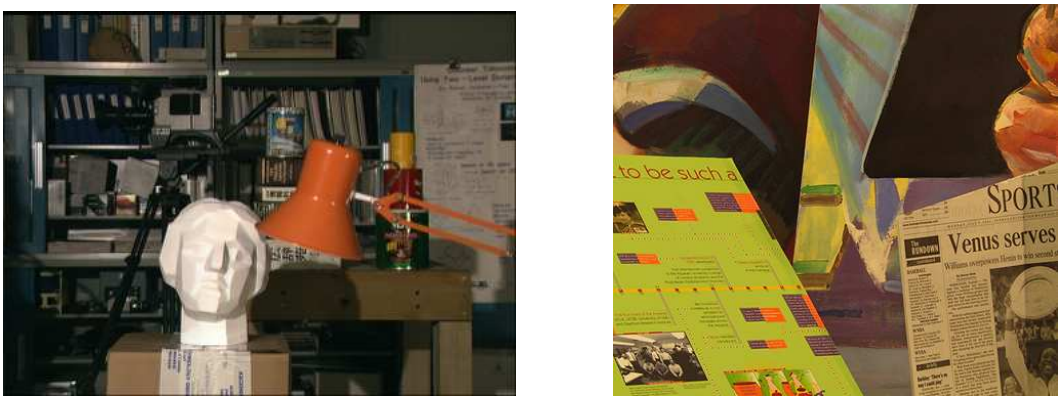
outliers disparity value based on their 16 reliable neighboring pixels. In this strategy, mismatch pixels are interpolated differently from occluded pixels. If a pixel is occluded, its disparity is taken from the background, that is, the minimum disparity of its neighboring pixels is selected as the disparity value of the outlier pixel. On the other hand, if the pixel is a mismatch point, the disparity of the pixel with the most similar color to the outlier pixel, is selected from the neighbors.

4. ***discAdjust()***, which first attempts to find the edges in the disparity map, that is, finding the depth discontinuities, and then, for each pixel on any detected edges, the disparity values of its neighbors on the both sides of the edge are sought. Next, the disparity value of the pixel of interest is updated with the disparity of any of the adjacent pixels with the least matching cost.

5. ***subpxEnhance()***, which applies a quadratic polynomial interpolation on the estimated disparity values to decrease the discretization error and is followed by a 3×3 median filter for smoothing the results.

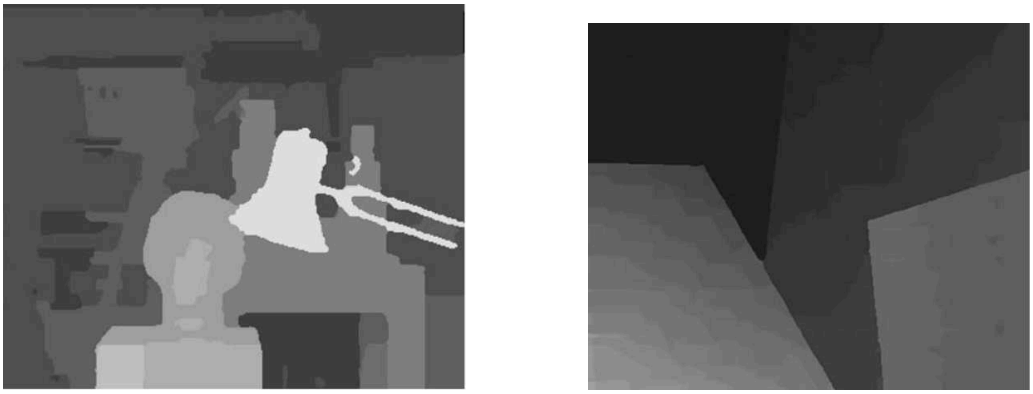
We have attempted to carefully follow and implement the strategies described in ADCensus solution as explained in [32]; however, measuring the extent to which our implementation has come close to the proposed method is a matter we could not truly investigate, due to unavailability of any actual disparity image to compare our results with or the original source code of ADcensus. It should also be noted that we could not proceed with porting our implementation to GPU due to time constraint and inadequate description about it in the published paper.

The disparity results from our implementation of ADCensus corresponding to Tsukuba and Venus images from Middlebury dataset, Figure 5.1, are shown in Figure 5.3, which nearly resemble the results published in the paper [32], presented in Figure 5.2. Since we could not find access to the original disparity images generated by ADCensus to make a solid comparison with our disparity results from ADCensusB, we leave it to the reader to judge the similarity between these images.



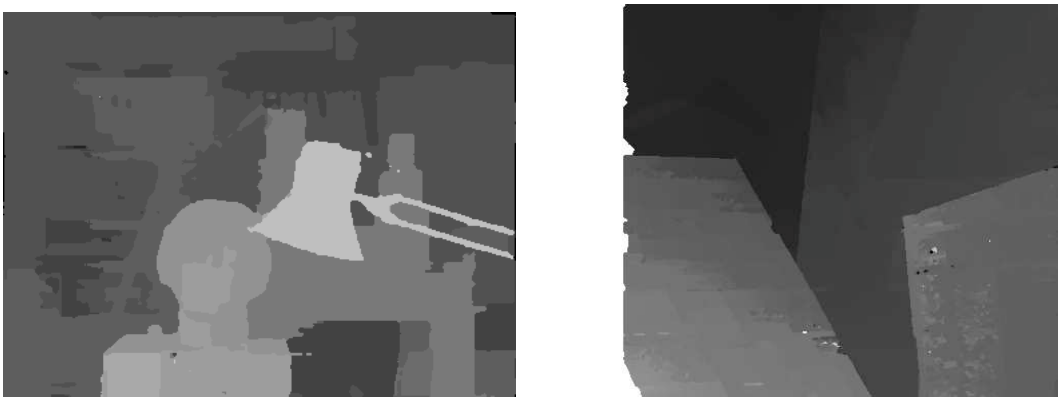
(a) Tsukuba image (b) Venus image

Figure 5.1: Sample images from Middlebury stereo dataset [39]



(a) Tsukuba (b) Venus

Figure 5.2: Disparity images by ADCensus for Middlebury images in Figure 5.1



(a) Tsukuba (b) Venus

Figure 5.3: Disparity images by ADCensusB for Middlebury images in Figure 5.1

Next, the set of parameters used at different steps of the evaluation are presented. It should be noted that these parameters were kept constant for all the images and experiments. However, if a parameter is changed during an experiment for specific reasons, it will be explicitly mentioned in the description of the experiment.

5.4.2 Masking

In order to test our hypothesis for the benefit of evaluating the disparity results in the areas of depth edges and their surroundings in AR applications, we build the corresponding masks using the OpenCV Canny edge detector and Dilation. Canny have been used to detect the depth edges in the ground truth disparity map, and the dilation operation for expanding the detected edge regions in the masking process. The extent to which the regions are expanded is determined by the number of iterations in the dilation operation. Table 5.1 shows the parameters used in the Dilation and the Canny edge detection. However, the *minimum threshold* in Canny is tuned and selected separately for each image since the threshold should change depending on the scene.

Table 5.1: Masking parameters

Dilation_iterations	10
Canny_apertureSize	3

The ground truth disparity maps in the Kitti stereo dataset are generated by a 3D laser scanner, thereby resembling a point cloud map of discrete disparity values. This property of the disparity images can be problematic for the masking process since it can result in many small streaks as the edges. Therefore, before applying any edge detection on the image, we need to first fill the gaps by interpolating the values and obtain a smoothed ground truth disparity. This can be achieved by applying a dilation operation. In our implementation, we have used the OpenCV dilation operation with different number of iterations for each image, that is set depending on the scene and the original ground truth disparity, to obtain a fully dense disparity

map. The new disparity images are then stored on the disk for further use. However, it should be noted that the dilated disparity images are only used in the construction of the edge masks while detecting the depth edges in the image.

5.4.3 Stereo Algorithms Settings

The parameters for each algorithm used in our experiments to generate the disparity maps are kept constant over all the images in the dataset. These parameters are presented in Tables 5.2 and 5.3 for SGBM and ADCensusB, respectively.

Table 5.2: SGBM parameters

SADWindowSize	9	disp12MaxDiff	2
uniquenessRatio	10	P2	3*9
speckleWindowSize	100	speckleRange	2

Other parameters not mentioned in the table are considered with their default values.

Table 5.3: ADCensusB parameters

λ_{AD}	10	λ_{Census}	30	L_1	34	L_2	17
τ_1	20	τ_2	6	π_1	1.0	π_2	3.0
τ_{SO}	15	τ_S	20	τ_H	0.4		

The minimum and maximum disparity values are also kept constant for each image pair in both algorithms; however, the maximum disparity differ for each image pair as the scenes are different and objects are located at different depth fields. The minimum disparity is set to 0 for both algorithms. The maximum disparity for each

image pair is selected based on the maximum value in their corresponding ground truth disparity. The only restriction to consider here is to choose a value greater than or equal to the maximum disparity of the ground truth that is a multiple of 16, which is a constraint in the available implementation of the SGBM algorithm in the OpenCV library.

5.4.4 Evaluation Parameters

In our evaluation model, due to the large amount of data which grows as more images are added to the input selection, plots are generated by taking the average of the results over all the images. As mentioned in the previous chapter, this average results from two steps; first, getting the average of the stereoacuity over specific depth ranges for each image and then calculating the average of the values from the previous step over all of the images. This operation finally results in a single plot that demonstrates the average stereoacuity within specific distances. The averaging operations at this step are implemented by building histogram over the resulting data. In our experiments, we set the number of bins to 100 and the total range is from 0 to $50m$. Therefore, the first averaging is conducted over distances of $0.5m$, the range of each bin, in each image and the maximum distance over which the results are examined is $50m$.

5.5 Experiments

In this section, we discuss the experiments conducted to evaluate the system and investigate the validity of our hypotheses.

5.5.1 Evaluation in Augmented Reality Framework

In this experiment, the disparity maps were generated for fifty-two image pairs with both SGBM and ADCensusB algorithms. After generating the corresponding disparity maps for all the images, the evaluation process was conducted on each map separately.

Sample plots for the estimated average disparity error, converted to effective stereoacuity, corresponding to one of the stereo pairs, shown in Figure 5.4, over the masked areas are displayed in Figures 5.5 and 5.6, respectively.

(a) Left image (b) Right image

Figure 5.4: Sample stereo image from the Kitti dataset

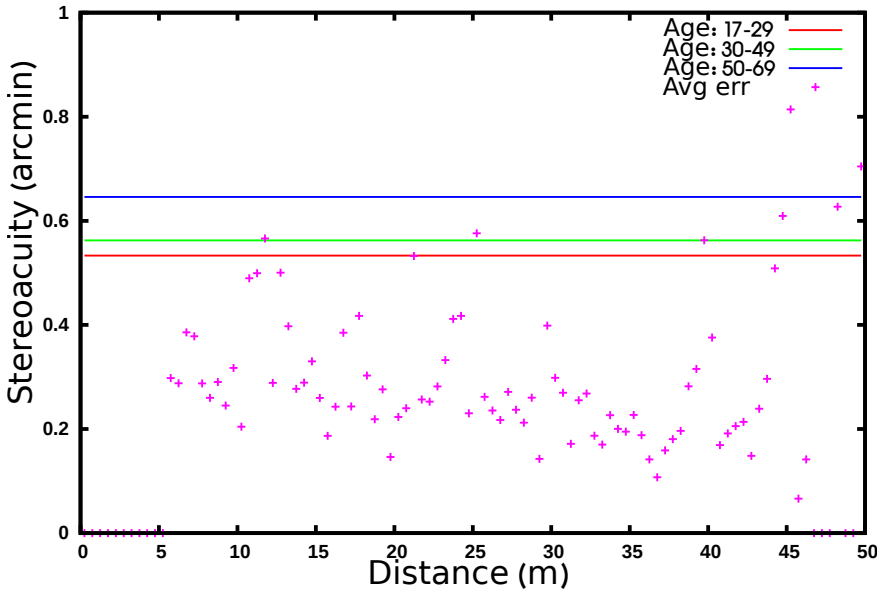


Figure 5.5: Average disparity error over distance by SGBM for Figure 5.4

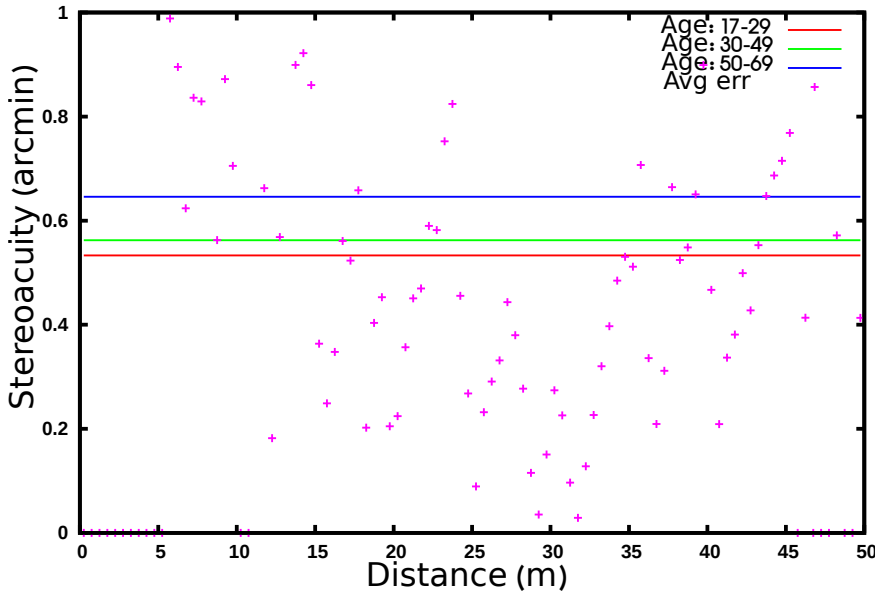


Figure 5.6: Average disparity error over distance by ADCensusB for Figure 5.4

The corresponding mask, Figure 5.7; masked ground truth, Figure 5.8; and the

masked disparity images generated by SGBM and ADCensusB, Figures 5.9 and 5.10 are shown below.

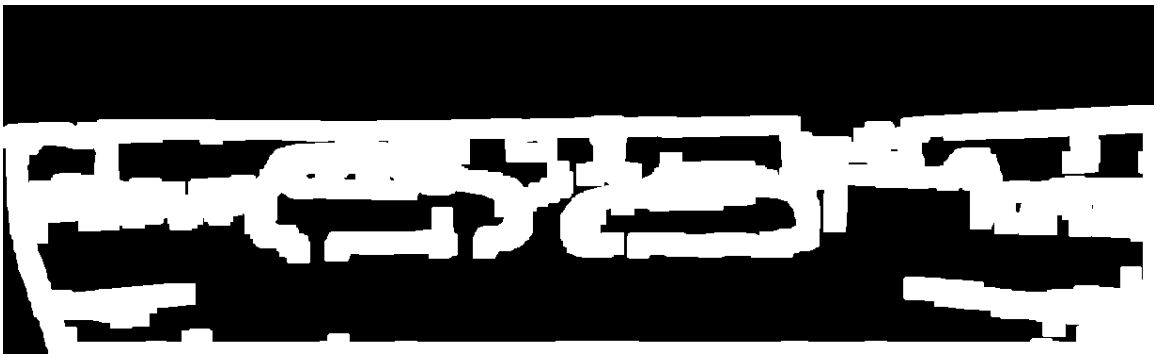


Figure 5.7: The mask of depth edges and their surrounding regions for Figure 5.4

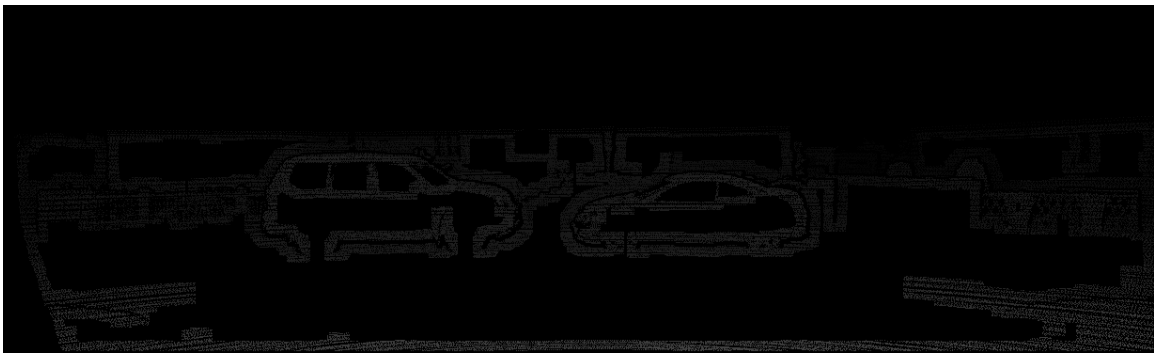


Figure 5.8: Masked ground truth for Figure 5.4

Figure 5.9: Masked disparity by SGBM for Figure 5.4

Figure 5.10: Masked disparity by ADCensusB for Figure 5.4

Figures 5.11 and 5.12 show the average results over all the disparity images for both SGBM and ADCensusB, respectively.

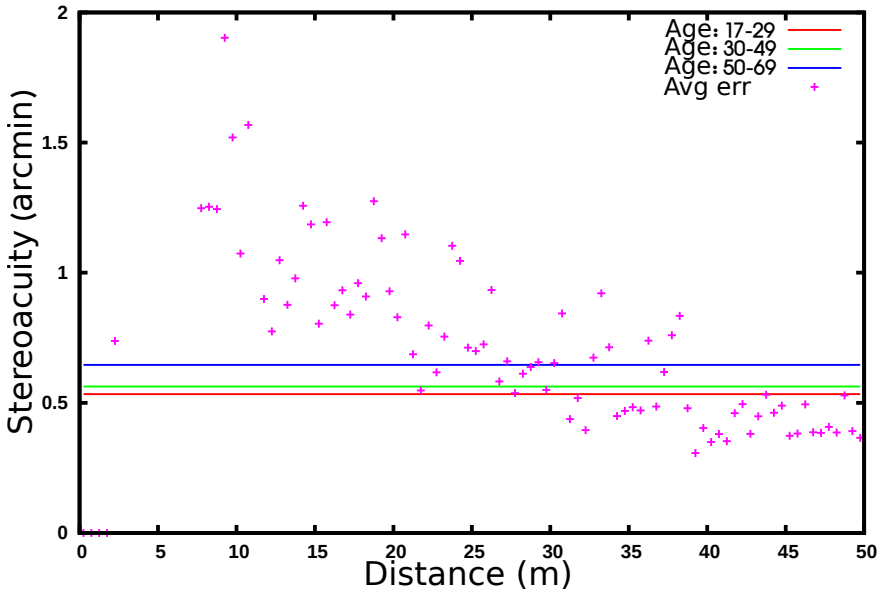


Figure 5.11: Average disparity error over all the images by SGBM

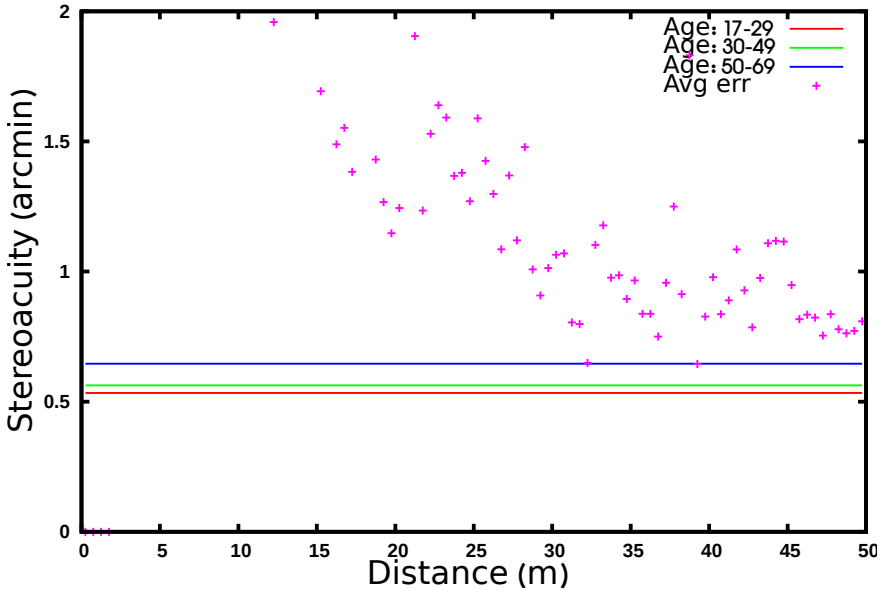


Figure 5.12: Average disparity error over all the images by ADCensusB

As can be seen, the average results displayed in the previous plots contain sparse

points and do not demonstrate any consistent pattern. When we investigated the cause of this large variation, we found that in the results of both algorithms, there are some disparity values which differ from the ground truth by a considerable amount and yet have not been invalidated by the algorithm. We assume that these types of outliers can be easily removed from the set by applying a post processing filter, or they will be eventually culled out by the 3D renderer in the AR system. Based on this assumption, we filter them out in our evaluation. In order to filter out the disparity values which largely differ from the ground truth disparity, we have integrated another step in our evaluation process. This step is similar to the strategy used in the Kitti and Middlebury evaluation models. In this step, the estimated disparity error is initially compared to a more generally defined threshold, for instance a threshold of 3 pixels. This comparison allows only for those values of disparity with an error less than or equal to the specified threshold to move on to the next steps of the evaluation. It should be noted in our design, the specified threshold is defined as a run-time variable.

The additional filtering had a significant impact on the evaluation results. In fact, a consistent pattern was observed in the final plots after filtering out the outliers with large differences. The results are displayed in figures 5.13 and 5.14.

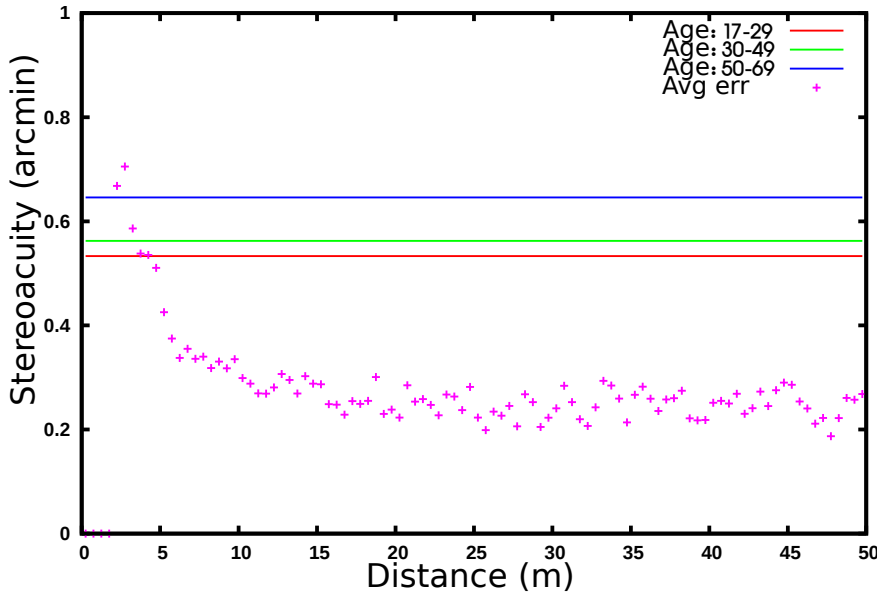


Figure 5.13: Average disparity error over all the images by SGBM after filtering large outliers

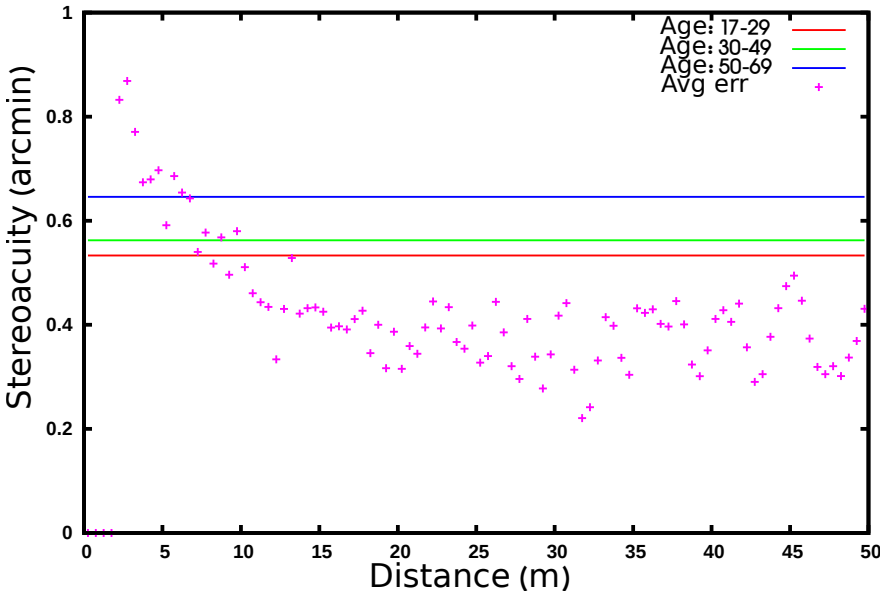


Figure 5.14: Average disparity error over all the images by ADCensusB after filtering large outliers

In these plots, a cross point below a stereoacuity threshold (straight lines) implies that the average error in the disparity values estimated by the stereo matching algorithm is imperceptible to the human visual system. However, a value higher than the threshold indicates that the error cannot be ignored by the human visual system and should be resolved to achieve a better alignment between the virtual and the real world in the AR application of interest. Moreover, most of the errors fall below the standard stereoacuity value corresponding to older ages; indicating that they are not perceptible to the visual system of the people at these particular ages.

The zero values in the plots imply that either there is no object within the corresponding range or the disparity value estimated by the algorithm is equal to the ground truth disparity; however, since the average of the results has been taken over all the images, it is more likely that the zero values indicate that no object was found within the particular range.

As can be seen in the results, SGBM performs better in finding more accurate corresponding matches compared to ADCensusB, as most of the error points fall below the standard stereoacuity lines. Moreover, the plots show that in both methods the significant amount of error corresponds to the near field objects, within the first 5 meters. This range of the depth field can be considerably important in some applications, such as the ones involving certain manipulative tasks.

5.5.2 Depth Edges and Occlusion

In order to examine the effect of evaluating certain regions of the disparity image instead of the whole image, we estimated the average error both for the masked areas

and the whole disparity map. Results of SGBM are shown in Figures 5.15 and 5.16.

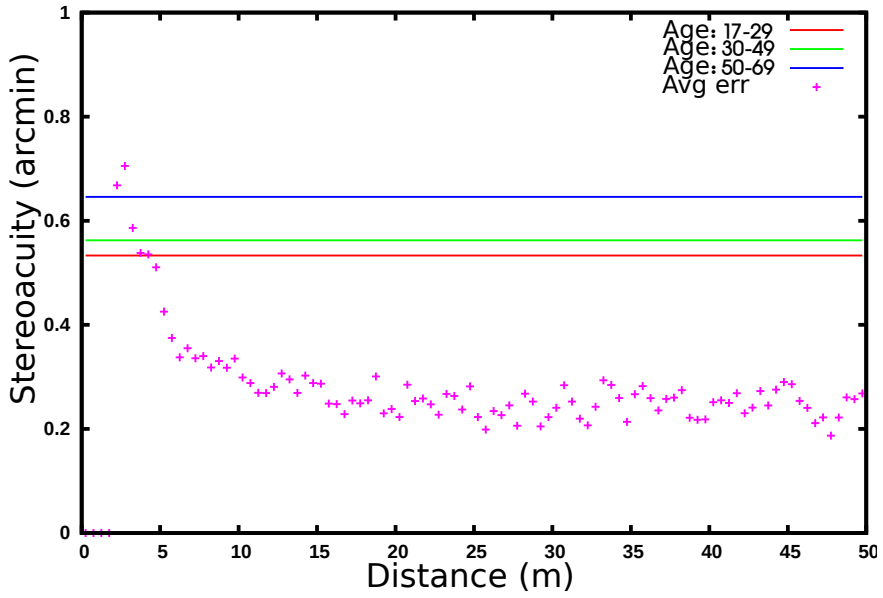


Figure 5.15: Average disparity error over masked areas by SGBM

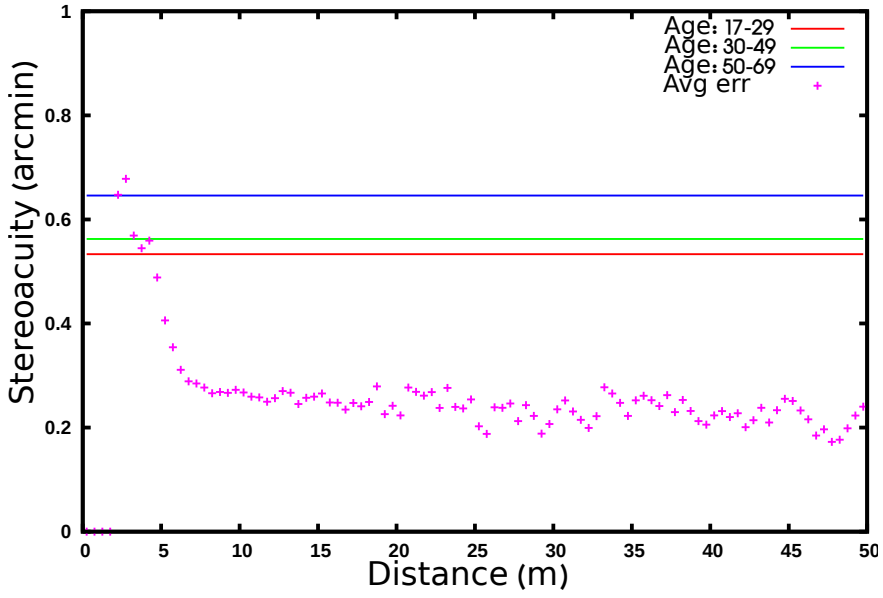


Figure 5.16: Average disparity error over the whole image by SGBM

The plots show that the average error over the masked regions, that is near the depth edges, is very similar to the results over the whole image. This may imply that there is no additional benefit in the inspection of these regions. However, this might be merely an indication of the performance of the selected algorithms and can be better analyzed by evaluating more algorithms within our model. In either case, we hypothesize that, due to the importance of occlusion and areas near depth discontinuities to the HVS [46], it is reasonable to focus more on the depth edges and their surroundings when designing or employing a stereo matching technique for an AR application.

5.5.3 Average Outliers

In this experiment the average outliers were measured for both algorithms. The values for both validity criteria mentioned in chapter 4, valid pixels in the ground truth and generated disparity, are presented in Tables 5.4 and 5.5 for each age group. For simplicity, we have labelled the valid pixels in the ground truth and the generated disparity with **valid_gtDisp** and **valid_genDisp**, respectively. Figure 5.17 presents a comparison between all the results for one of these validity criteria, when the ground truth disparity is valid.

Table 5.4: Average outliers for the masked regions

		Avg_Outliers	
Algorithm	Age	valid_gtDisp	valid_genDisp
SGBM	17-29	0.12	0.16
	30-49	0.11	0.15
	50-69	0.09	0.12
	70-83	0.0012	0.0016
ADCensusB	17-29	0.23	0.32
	30-49	0.22	0.31
	50-69	0.18	0.27
	70-83	0.002	0.003

Table 5.5: Average outliers for the whole image

		Avg_Outliers	
Algorithm	Age	valid_gtDisp	valid_genDisp
SGBM	17-29	0.11	0.14
	30-49	0.10	0.12
	50-69	0.08	0.09
	70-83	0.005	0.007
ADCensusB	17-29	0.27	0.39
	30-49	0.26	0.37
	50-69	0.22	0.32
	70-83	0.002	0.003

Results in Figure 5.17 show that in both cases, the masked regions and the whole image, SGBM has less outliers than ADCensusB, indicating that SGBM generates a more accurate disparity map as perceived by the human visual system. Another observation from Figure 5.17 is that in SGBM, the average outliers over the masked regions are more than the outliers over the whole image, whereas in ADCensusB the opposite behavior is observed. This implies that SGBM generates less accurate results near the depth discontinuities and occluded regions compared to the other areas in the image. On the other hand, ADCensusB generates more accurate disparity

values near the depth edges compared to the other regions in the image and tends to preserve the occluded regions. This indicates that, despite the better performance of SGBM over ADCensusB according to the experimental results, it is important to consider this behavior to employ the stereo correspondence method in the right application based on the requirement of the target system for the accuracy of the depth results in different regions; in other words, it is reasonable to first investigate which regions in the image are more important in the context of the target application. For instance, ADCensusB performs better in an application where the areas near the depth discontinuities and occlusion are more important than the rest of the image, such as image compositing for layering visual elements on the scene, compared to application scenarios where obtaining an accurate, dense disparity map for all the regions in an image is essential, such as constructing a 3D model of the scene or preparing a model for 3D printing.

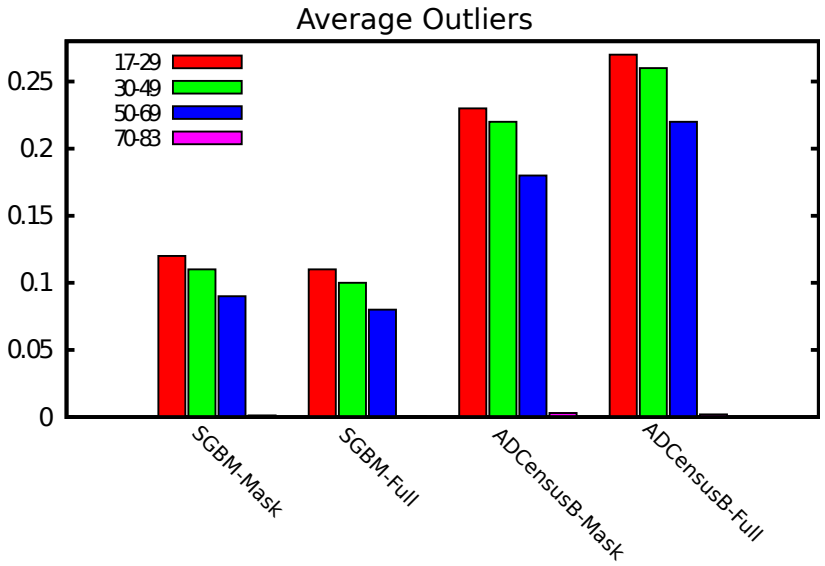


Figure 5.17: Average outliers for SGBM and ADCensusB over the masked and the whole image; each bin color corresponds to different age groups with specific stereoacuity threshold

We should note that the reason the number of outliers for the valid pixels in the generated disparity is more than the outliers for the valid pixels in the ground truth is that in our implementation, the counter for the number of pixels in the ground truth image is incremented whenever a disparity pixel in the ground truth is labelled as valid, regardless of other conditions in the process; however, the counter for the number of pixels in the generated disparity is only incremented whenever the disparity value is valid and the amount of disparity error is less than the specified pixel threshold in the evaluation process, thus resulting in a smaller denominator of the fraction in the estimation of the average outliers for the criteria of valid pixels in the generated disparity map and, therefore, a larger average value in the end.

5.5.4 Average Disparity Error

The average disparity error has also been estimated in the evaluation process for both pixel validity criteria. However, the resulting values were similar for both cases and, therefore, only one value is reported in the following table for this metric.

Table 5.6: Average disparity error

Algorithm	Region	Avg_DispErr
SGBM	Full	6.58
	Masked	7.81
ADCensusB	Full	4.49
	Masked	4.74

As can be seen, ADCensusB results in less average disparity error than SGBM. This difference is likely caused by the various refinement steps implemented in the ADCensusB algorithm which do not exist in SGBM. As a result, despite the larger number of outliers in ADCensusB than SGBM as measured in the previous experiments, ADCensusB attempts to decrease the difference between the resulting disparity value and the ground truth disparity, thus generating smoother disparity patches within different regions of the images.

5.5.5 Real-time Execution

In another experiment, we estimated the average execution time for both algorithms. Results show that the average execution time over all the images for SGBM and ADCensusB are 0.54 and 272.82 seconds, respectively. Considering the requirements of having an interactive real-time AR system [17], the processing time of each frame

should not be more than 0.06-0.08 seconds. Therefore, we need to get at least 6.75× and 3410.25× speedup for each algorithm, respectively. Although the current implementation of SGBM could be used when the real world scene remains stable for approximately one second, it can be safely concluded that none of these algorithms meet the requirements of a real-time interactive AR system. This suggests that GPU-based solutions along with using more advanced hardware are more suitable to achieve the processing speed required for the real-time interactive applications of AR. In [32], it is stated that for the Tsukuba image of Middlebury dataset, the CPU implementation takes 2.5 seconds whereas the GPU implementation takes only 0.016 seconds. Their evaluations on a PC with Core2Duo 2.20GHz CPU and NVIDIA GeForce GTX 480 graphics card show that their GPU implementation of ADCensus brings 140× speedup in the processing speed. More speedup may be achieved using modern graphics cards; however, we cannot make a numerical comment on the amount of speedup as it depends on different specifications of the hardware and also the nature of the algorithm. If we were to look at only the number of cores alone, current hardware has 5 times more cores than the hardware used by [32]. With a software implementation that took advantage of the increase in the number of cores, we could in principle expect an execution time of 0.0032 seconds.

5.5.6 Effect of Refinement

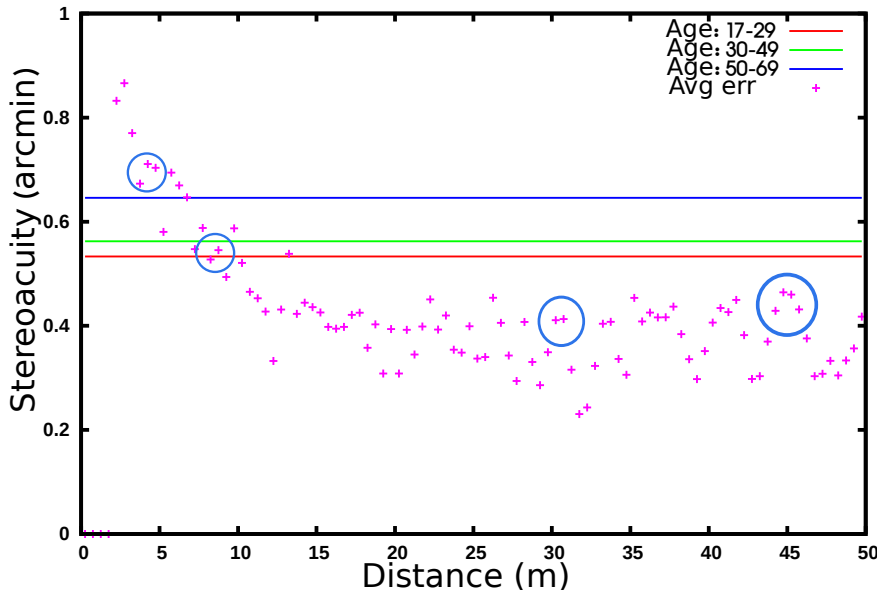
In this experiment, we studied the effect of the post processing steps, also referred to as the *refinement steps*, in the stereo algorithms on the accuracy of the results in our evaluation criteria.

Refinement is usually the last step in a stereo correspondence algorithm because it attempts to decrease the number of wrong matches or the error after the disparity results have been found [41]. Therefore, this step must be applied after the outliers, that is the wrong pixel matches, have been detected in the results. The detection of the outliers occurs through a check known as left-right consistency check in a stereo matching algorithm. In this check, the disparity map for both the left and right image is first calculated. Then, if a pixel in the left image, based on its disparity value, corresponds to a pixel in the right image that does not map back to it, it will be labeled as an outlier [41]. This description can be formulated as follows:

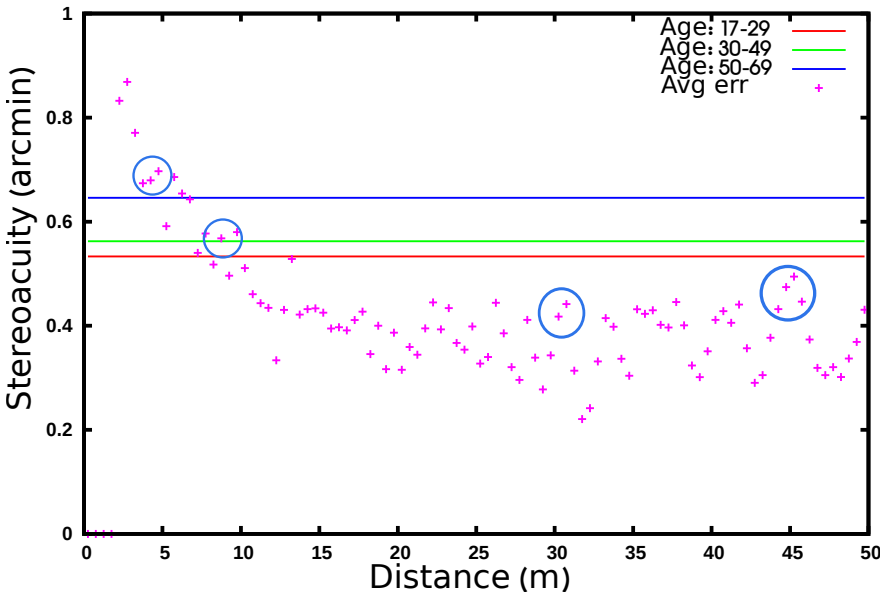
$$D_L(p) \neq D_R(p - (D_L, 0)) \tag{5.2}$$

Where $D_L(p)$ is the disparity function for the left image and D_R is the disparity function for the right image.

For simplicity, we will refer to this check as L-R check in this report. In our implementation of ADCensus, we have the L-R check and its subsequent refinement steps triggered with a flag. Therefore, when the flag is not set, neither the check nor the refining steps are triggered in the algorithm. To investigate the effect of the refinement on the final results, we used ADcensus in this experiment with the L-R flag set to zero, generating the disparity results for the image pairs, and evaluating the results. The results for both cases, not refined and refined, over the masked regions and the whole image are shown in Figures 5.18 and 5.19, respectively.

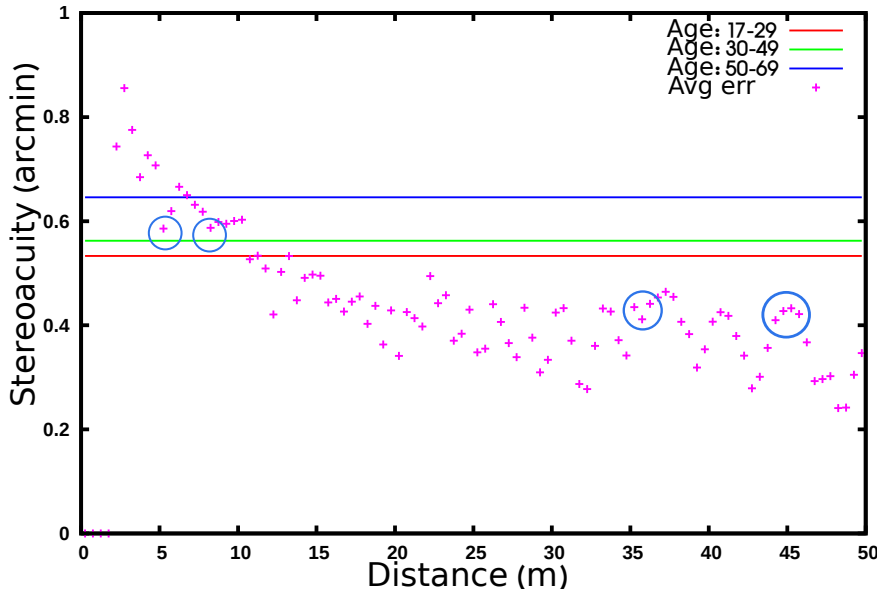


(a) No refinement on disparity results

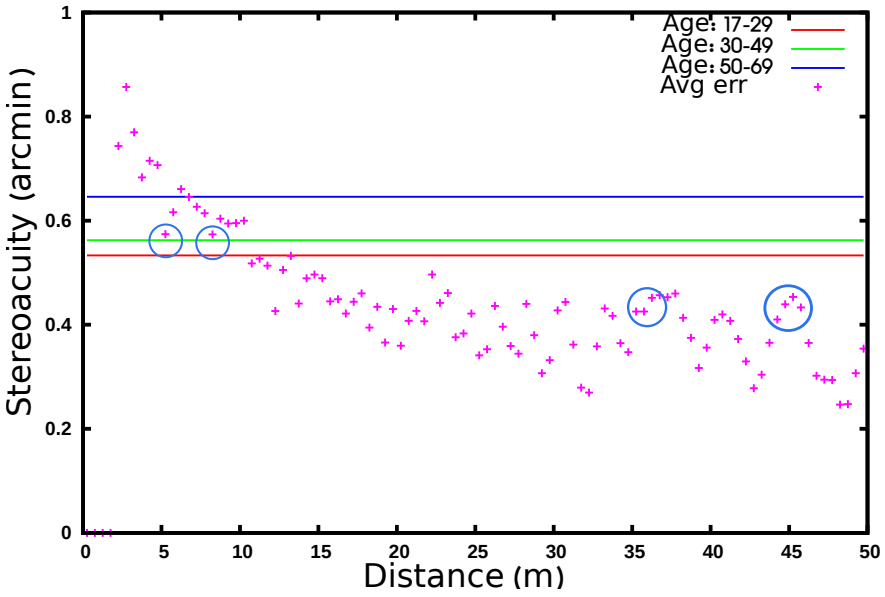


(b) Refinement on disparity results

Figure 5.18: Average disparity error by ADCensusB for the masked images; blue circles show some sample values that have slightly changed as a result of refinement



(a) No refinement on disparity results



(b) Refinement on disparity results

Figure 5.19: Average disparity error by ADCensusB for the whole images; blue circles show some sample values that have slightly changed as a result of refinement

As can be observed in the plots, the evaluation results in our specific criteria are not significantly different from the results of the algorithm when L-R check and refinement were triggered and only a few average values have slightly changed. We have marked a few of these values with blue circles in Figures 5.18, 5.19.

We also estimated the average execution time, the average disparity error, and the average outliers in this experiment. The results for the average error and outliers are shown in the tables below.

Table 5.7: ADCensusB average disparity error - unrefined

Region	Avg_DispErr
Masked	5.59
Full	5.29

Table 5.8: ADCensusB average outliers - unrefined

		Avg_Outliers	
Region	Age	valid_gtDisp	valid_genDisp
Masked	17-29	0.23	0.33
	30-49	0.22	0.31
	50-69	0.18	0.27
	70-83	0.002	0.003
Full	17-29	0.27	0.39
	30-49	0.26	0.37
	50-69	0.23	0.32
	70-83	0.001	0.002

Figure 5.20 shows a comparison between the average outliers by ADCensusB with the effect of refinement and without it for the masked and the whole images in one of the validity criteria, that is, when the ground truth disparity is valid. As can be seen, no significant decrease is obtained in the number of outliers.

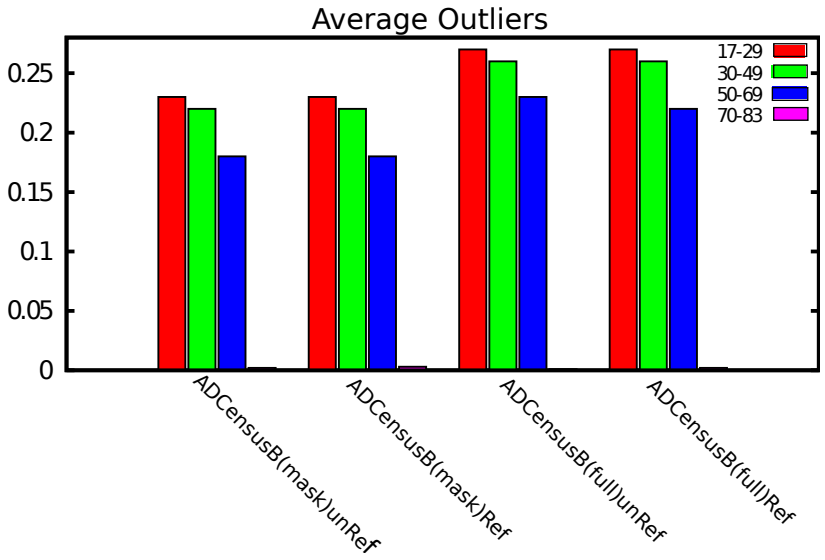


Figure 5.20: Average disparity error by ADCensusB in refined and unrefined cases for both masked and whole images; each bin color corresponds to different age groups with specific stereoacuity thresholds

The average execution time was approximately 147.84 seconds which is nearly half the running time of the algorithm with the L-R check and refinements triggered, Table 5.9. Comparing these results to the ones presented in Tables 5.4, 5.5, and 5.6 a slight decrease in the amount of errors and nearly no change in the number of outliers is observed. Analyzing the results in this experiment, we can conclude that despite the considerable rise in the execution time of the algorithm, no significant improvement in accuracy is achieved in our evaluation criteria through refinement of the disparity results; therefore, the execution of ADCensusB without any L-R check and refinement step is more beneficial to an AR application in outdoor environments, since it requires less processing time.

Table 5.9: ADCensusB average execution time - refined and unrefined

ADCensusB	Avg_ExecTime (secs)
refined	272.82
unrefined	147.84

5.5.7 Discretization Degree of Disparity Values

According to different studies [10, 28, 1], some other factors such as issues associated with the environment, display device, and capturing device can also affect the perception of depth in the visual system. As a result, the ability to detect the difference in depth and to accurately estimate the depth of different points, in practice, do not merely depend on the implemented discretization level of the disparity values in the stereo correspondence algorithm. In order to investigate the validity of this statement, we conducted the following experiment. In this experiment we defined some stereoacuity thresholds. In order to find the minimum threshold to start with, we attempted to find the minimum disparity change in the ground truth disparity images. To this end, we move along horizontal scanlines in each image and compute the difference between the values of consecutive pixels, which is, in fact, an indicator of the detectable threshold of the changes in depth between different pixels, Figure 5.21.

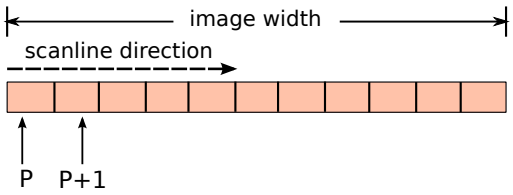


Figure 5.21: The scanline pixels difference process

After finding the minimum value in each image, a global minimum is sought between all the computed values from different images. The value we found for a group of twelve images selected from our dataset with the size of 1242×375 and the focal length of 721 pixels, as reported by Kitti stereo [15], was 0.0022 arcmins. After finding this minimum and defining our thresholds, we apply a nearly similar operation on the disparity results of the same group of images from each of the sample algorithms. In this process, while moving through each image, for those pixels whose generated disparity is close to the ground truth disparity, within a specific pixel threshold, we estimate their depth difference from their following pixel in the ground truth and compare the value to each of the specified thresholds; if this value is less than a threshold, then we check to see whether the stereo algorithm has also detected different values for the corresponding pixels. In case of detection, we increment a counter corresponding to each threshold that indicates the number of detected pixels. This process is repeated for different images and, finally, the average of detected pixels is estimated for each specified threshold.

```
ADP; START
      define StAc_thresh;
      for all images; do
         for all pixels p in the image:
            if ( |disp_gt − disp_gen| < pix_thresh)
               pix_count += 1;
               dispDiff = |disp_gen_p − disp_gen_p+1|;
               depth_gt_p = (focal_length × baseline) / disp_gt_p ;
```

$depth_{gt_{p+1}} = \frac{focal\_length \times baseline}{disp_{gt_{p+1}}}$;

$depthDiff = |depth_{gt_p} - depth_{gt_{p+1}}|$;

$stAc\_detected = \frac{pupil\_distance * depthDiff}{depth^2_{gt_p}}$;

```
for each threshold thr in StAc_thresh:
    if (stAc_detected < thr)
        if (dispDiff > 0)
            detected[thr]++;
        end if
    end if
end for
end if
end for
for each threshold thr in StAc_thresh:
    Avg_detected[thr] = detected[thr]/pix_count;
    Append(Avg_pixFile, Avg_detected[thr]);
end for
end for
/**Concatenate the files for all images into one **/
for each Avg_pixFile:
    Concat(AllimgFile,Avg_pixFile);
end for
/**Getting the average over all images for each threshold**/
Calc_Average(OutAvgFile, AllimgFile);
plot(OutAvgFile);
```

```
ADP; END
```

The results for both algorithms are shown in Figure 5.22. As can be seen in these plots, for both algorithms, the average detected pixels with detectable change in depth values starts to converge at the value of approximately 0.4 arcmins. We also observe that for the values below this threshold the average detected pixels are very small and for some values, such as the minimum detectable threshold in ground truth, both algorithms are not capable of detecting any change in depth values. This implies that, regardless of the accuracy resolution of the algorithms, which is 1/8th of a pixel for SGBM, approximately 0.6 arcmins, and 1/16th of a pixel for ADCensusB, approximately 0.3 arcmins, for Kitti images based on the camera parameters and the geometrical relation presented in Figure 5.23 and Equation 5.3, some changes in depth in the real world still cannot be detected by the algorithm. This effect might be due to the constraints of the sensor, that is, the errors associated with the capturing device and its resolution, or the environmental noise.

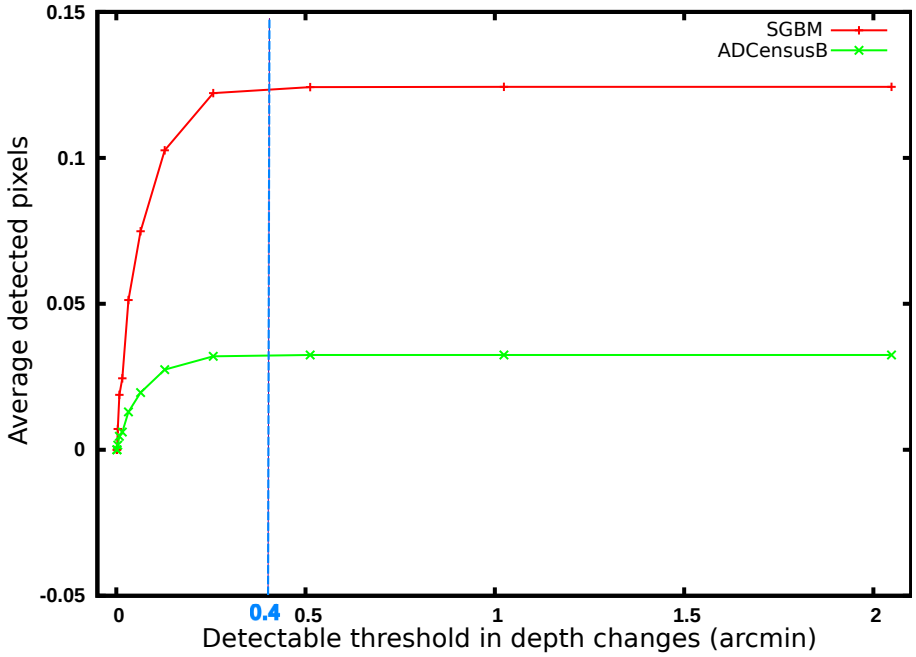


Figure 5.22: Average of detected pixels by SGBM and ADCensusB for specific stereoacuity thresholds marked on each curve for a group of 12 images; the vertical blue line shows the approximate threshold after which the average of detected pixels converge

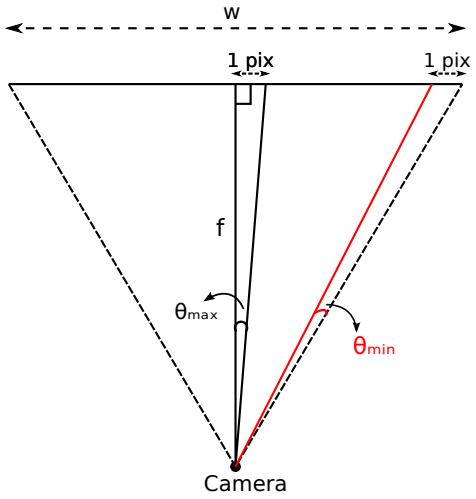


Figure 5.23: Resolution of image in angular disparity

In Figure 5.23, w is the image width and f is the focal length of the capturing device.

$$\theta = \arctan(\frac{pixel\_resolution}{focal\_length}) \tag{5.3}$$

For the image size of 1242×375 pixels and focal length of 721 pixels, and based on the resolution of SGBM and ADCensusB in the estimation of the disparity values, the minimum and maximum detectable disparity, that is, at the center and at the boundary of the image, respectively, in terms of effective stereoacuity are as follows:

$$\begin{aligned} SGBM : \theta_{max} &= \arctan(\frac{\frac{1}{8}}{721}) \\ &= 0.00993 \text{ degrees} \times 60\frac{arcmins}{degrees} = 0.596 \text{ arcmins} \end{aligned} \tag{5.4}$$

$$\begin{aligned} SGBM : \theta_{min} &= \arctan(\frac{\frac{1}{8} \times (\frac{1242}{2})}{721}) - \arctan(\frac{\frac{1}{8} \times (\frac{1242}{2} - 1)}{721}) \\ &= 0.00982 \text{ degrees} \times 60\frac{arcmins}{degrees} = 0.589 \text{ arcmins} \end{aligned} \tag{5.5}$$

$$\begin{aligned} ADCensusB : \theta_{max} &= \arctan(\frac{\frac{1}{16}}{721}) \\ &= 0.00496 \text{ degrees} \times 60\frac{arcmins}{degrees} = 0.298 \text{ arcmins} \end{aligned} \tag{5.6}$$

$$\begin{aligned} ADCensusB : \theta_{min} &= \arctan(\frac{\frac{1}{16} \times (\frac{1242}{2})}{721}) - \arctan(\frac{\frac{1}{16} \times (\frac{1242}{2} - 1)}{721}) \\ &= 0.00495 \text{ degrees} \times 60\frac{arcmins}{degrees} = 0.297 \text{ arcmins} \end{aligned} \tag{5.7}$$

However, as can be seen, the minimum and maximum angular resolution in the image are not considerably different.

As a result of this experiment, we can conclude that in order to achieve more accurate depth results in the stereo algorithms and correctly detect the difference

between depth values, that is, to obtain a lower threshold of depth changes closer to the actual resolution of the implemented algorithm, using higher resolution devices and considering their robustness to noise are also essential. Based on the information provided in Chapter 3 about the average stereoacuity in the HVS, we can say that the lower bound resolution of a capturing device with focal length of 721 pixels should be $\frac{1}{8}th$ of a pixel. In the end, we should note that the experimental results presented earlier in this chapter show that despite various types of error relevant to the capturing device, environmental noise, and the actual accuracy of the stereo correspondence algorithm in the estimation of disparity values, the effect of such errors on the results will still be imperceptible for most cases to the HVS in outdoor AR applications, especially where objects are distant from the observer.

5.6 Overview

Table 5.10 shows an overview of the difference between our proposed evaluation approach and the other evaluation models, Middlebury and Kitti, in terms of the estimated evaluation metrics.

It should be noted that although the average error and the average outliers exist in the other evaluation schemes as well, the major difference which makes our evaluation more appropriate than the other schemes for practical applications of AR, is the approach employed during the design of the metrics and the analysis of the results in the evaluation process. In fact, integrating the important factors related to the human visual system and its perception of depth in the design of the metrics and the insights they provide make the evaluation model more relevant and applicable to

outdoor AR systems.

Table 5.10: Comparison of different evaluation schemes

Metrics	**Evaluation Models**		
	Middlebury	*Kitti*	*Comprehensive_Evaluation*
Avg_StAc	✗	✗	✓
Avg_Outliers	✓	✓	✓
Avg_DispErr	✓	✓	✓
Avg_ExecTime	✗	✗	✓

5.7 Hypotheses Validation

Next, we will review our hypotheses mentioned earlier in this chapter and discuss their validity in light of our experiments and their results.

- **Hypothesis 1**: *Our model is more suitable than other approaches to evaluate and demonstrate the performance of the stereo matching algorithm in the framework of outdoor augmented reality applications.*

  As can be seen in the results of the experiments, our system employs a systematic approach for the measurement and demonstration of different evaluation metrics in the framework of an outdoor augmented reality application. In our system, the disparity error, which is the most important metric for the accuracy of the disparity results, is converted to a certain measurement, stereoacuity, that is relevant and applicable to the human visual system and its perception of depth. We evaluated two stereo matching methods in our system and analyzed their performance in terms of the accuracy of the depth map for an outdoor AR application. Due to the application-oriented design of the system, we could

comment on the suitability of each method for an AR application in outdoor environments. As a result, we can argue that our evaluation model is more appropriate for the evaluation of the solutions in an outdoor AR system than the conventional evaluation systems.

- **Hypothesis 2:** *Observing, evaluating, and consequently refining the areas near the depth edges in an image are more important in an AR application.*

  The results of our experiments on two sample stereo matching solutions showed no significant difference between the evaluation metrics corresponding to the whole image and the regions of the depth edges and their surroundings, thereby implying that there may be no specific benefit into the analysis of these particular regions in an outdoor AR application. However, due to the importance of these regions as depth cues to the human visual system for the perception of the 3D location of the surrounding objects in an environment [46], we argue that more solutions should be tested within our evaluation model to be able to certainly approve or disapprove this hypothesis.

- **Hypothesis 3:** *Our system performs better than the the conventional evaluation models for assessing the performance of a stereo algorithm in Real-time AR applications.*

  As explained in the design of our model and demonstrated in the experimental results, the execution time of the algorithms is estimated and evaluated in the system based on the requirements of having a real-time and interactive augmented reality application. In the experiments, we evaluated the running time of the two sample stereo matching algorithms which proved to be inefficient in

both cases for a real-time augmented reality application. Through this property, we can claim that the evaluation results through our system is more beneficial to AR applications than the conventional evaluation models which do not take this important aspect of the solutions into account.

- **Hypothesis 4:** *The trade-off between the accuracy and the running time of the stereo algorithms can be effectively evaluated in the framework of an outdoor AR application through our system.*

  In one of the experiments, we focused on the trade-off between the accuracy of the results and the running time of the algorithm by studying the effect of the post processing steps, also referred to as refinement steps, on the evaluation metrics. Results on ADCensusB showed that integrating these steps in the algorithm does not significantly improve the accuracy of the results in the framework of an outdoor AR system; on the other hand, it causes a considerable increase in the execution time of the algorithm which is detrimental to a real-time and interactive AR system. The results of this evaluation indicates that the trade-off between the accuracy of the results and the execution time of the algorithm, which normally exists in nearly all the stereo matching solutions, can be effectively analyzed through our evaluation system. The other available evaluation models lack this property which is of great importance to outdoor applications of AR.

- **Hypothesis 5:** *The ability to detect the difference in depth in stereo correspondence methods not only depends on their accuracy in estimation of the disparity values, but is also affected by other factors, such as the environmental noise,*

the resolution of the capturing device and its robustness to noise.

Our experimental results show that, regardless of the theoretical accuracy of the implemented algorithm in the estimation of depth values and its resolvability of depth changes, its effective stereoacuity threshold, that is, its ability to detect the changes in depth values, can be different from its ideal detectability threshold and is affected by other factors, such as the error associated with the capturing device or its resolution. Therefore, in order to achieve a higher quality depth map of the surrounding environment in outdoor AR applications, in addition to a wise choice of stereo correspondence algorithm, considering the resolution of the capturing device and its robustness to noise is also essential. It should be noted that this conclusion is apart from the observation in other experiments which show that in many cases the error in the estimated depth values will not be perceived by the HVS.

Chapter 6

Conclusion

In this chapter, we succinctly mention our contributions in this study and specify some interesting paths for future research and improvement to our proposed system.

6.1 Contributions

Due to the emergence of various applications which combine different techniques in computer vision to build a practical system, developing testbeds which are particularly designed for the evaluation of different components in the criteria of the important factors in the target application is essential. Nowadays, building practical AR applications is a challenging problem due to the various constraints that these systems normally face. We believe that addressing these constraints and attempting to find the efficient solutions are propitious research directions.

In this research, we suggest that stereo correspondence methods can be used in outdoor AR systems as a practical alternative to conventional and inefficient technologies, 3D laser scanner or depth cameras, for obtaining the depth map of the

surrounding environment, but only if a real-time implementation is used. This approach, that is, employing stereo correspondence solutions, requires an evaluation scheme which can effectively evaluate the stereo correspondence methods while taking the target application into consideration. As a result, the available evaluation models, Middlebury and Kitti, will not be sufficient for an effective evaluation of the solutions. Therefore, our main contribution in this study, is proposing an application-oriented evaluation scheme that is designed in the light of the most important factors in an outdoor AR system. Since humans are the ultimate users of an AR system, we have focused on the relevant factors in the human visual system that are important for the real-time interaction with the augmented world and the perception of depth of the surrounding environment. We have integrated specific metrics in our evaluation system which are measured and subsequently evaluated in the framework of an outdoor AR application, thus effectively analyzing the performance of the solutions in terms of their accuracy and efficiency, that is its execution time, for the target application. These metrics are the average stereoacuity over distance, the average number of outliers, the average disparity error and the the average execution time. We also suggest that some specific areas in a scene are of more importance to AR applications in outdoor environments. Due to the importance of depth discontinuities and occlusion as depth cues to the human visual system, we define these regions as the depth edges and their surroundings in the scene. Although our experiments did not prove to be sufficient to investigate the validity of this hypothesis, we would still hypothesize that these regions are worthy of being studied in AR applications. In addition, the trade-off between the accuracy and the running time of a stereo algorithm can be studied through our system in the framework of an outdoor AR system,

thus better determining the net benefit of certain post processing steps to the target application.

In conclusion, the experimental results in most cases showed the effectiveness of our approach for evaluation of the stereo solutions in outdoor AR applications, which encourages further research in this particular direction to improve this model in more useful aspects. Next, we will mention some aspects in which we believe the system can be improved.

6.2 Future Work

This evaluation model can be improved in a few aspects that we will discuss here. As seen in the experiments, we could not certainly determine the importance of the depth edges in the scene to the outdoor AR application and subsequently their consideration in the evaluation of the stereo algorithms. We believe that a solid conclusion can be made by evaluating more stereo matching solutions within our model and observing the results in the masked regions and the whole image. We had envisioned a user study in which the user estimates the distance of a particular synthetically generated object which is placed at different sections of the scene, we could ask the user to say whether the synthetic object was in front of behind one of the objects in the scene. We had also envisioned another user study in which we used Google Earth and the maps of the campus to validate the objects we found in the real environment that were added or removed from the 3D model of the campus.

Earlier at this research we thought to conduct a user study to observe the effect of different weather conditions. Eventually, we decided to discard this evaluation

since we thought that the only possible outcome would be deterioration on different weather and illumination conditions. However, an interesting question to address in later studies would be the degree of decrease in users performance and decrease of the algorithms success when estimating the depth of various objects in the scene.

Another interesting aspect of improvement is a rigorous study on the effect of other factors that can affect the effectiveness and usability of the outdoor AR system and, therefore, are important to be considered in the evaluation of the method that is used to obtain the depth map of the surrounding environment. To name some of these factors we can refer to the resolution of the display devices in the AR system and the effect of contrast and brightness. For this evaluation, different video streams or stereo images can be captured using devices with different resolutions and from different scenes where the effect of shadow and lighting is well depicted. A segmentation algorithm can then be used to segment specific regions with different effects of shadow and lighting and the depth results by the algorithm in these areas can then be estimated and evaluated. A user study, similar to the ones mentioned previously, can also be conducted here to observe and evaluate the effects of the depth results and their errors on the HVS in each case. A more complete list of the important factors in AR can be found in the survey on the perceptual issues in augmented reality by Kruijff et al. [28].

There are different post processing techniques in computer vision that can be used to refine the disparity results, such as color segmentation and plane fitting, anisotropic diffusion, and common smoothing filters as Gaussian filter. However, most of these techniques can considerably increase the execution time of the algorithm. A study of different refining methods and their effect on the accuracy and the running time of

the algorithm in the framework of a particular application, such as an outdoor AR system, is an interesting topic to investigate and can be a valuable asset to many industrial applications.

Another interesting subject for studying is focusing on the evaluation of the existing GPU-based stereo matching techniques in our system. Investigating their suitability for integration in an outdoor AR system based on their running time, which is expected to be considerably less than many CPU-based solutions, and the accuracy of their results in the light of the relevant factors to an outdoor AR system can be an interesting subject for research.

Furthermore, we believe that it will be of specific value to assess the benefits of our proposed model and its applicability to other applications of augmented reality, such as underwater environments, in which the environmental noise is more significant and is more challenging to address.

We certainly encourage the interested researchers to investigate these aspects as we believe the increasing development of the hybrid systems in the fields of augmented reality and stereo vision require a more systematic way of evaluation to effectively investigate the usability and effectiveness of the system in the target application.

Bibliography

[1] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent advances in augmented reality. *Computer Graphics and Applications, IEEE*, 21(6):34–47, 2001.

[2] A. F. Bobick and S. S. Intille. Large occlusion stereo. *International Journal of Computer Vision*, 33(3):181–200, 1999.

[3] R. C. Bolles, H. H. Baker, and D. H. Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.

[4] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(9):1124–1137, 2004.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, 2001.

[6] Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *Computer Vision, 2001. ICCV*

2001. Proceedings. Eighth IEEE International Conference on, volume 1, pages 105–112. IEEE, 2001.

[7] M. Z. Brown, D. Burschka, and G. D. Hager. Advances in computational stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(8):993–1008, 2003.

[8] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[9] U. R. Dhond and J. K. Aggarwal. Structure from stereo - a review. *Systems, Man and Cybernetics, IEEE Transactions on*, 19(6):1489–1510, 1989.

[10] D. Drascic and P. Milgram. Perceptual issues in augmented reality. In *Electronic Imaging: Science & Technology*, pages 123–134. International Society for Optics and Photonics, 1996.

[11] S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster. A touring machine: Prototyping 3d mobile augmented reality systems for exploring the urban environment. *Personal Technologies*, 1(4):208–217, 1997.

[12] R. Fisher, S. Perkins, A. Walker, and E. Wolfart. Image Processing Learning Resources. `http://homepages.inf.ed.ac.uk/rbf/HIPR2/dilate.htm`, 2013.

[13] R. Fisher, S. Perkins, A. Walker, and E. Wolfart. Image Processing Learning Resources. `http://homepages.inf.ed.ac.uk/rbf/HIPR2/erode.htm`, 2013.

[14] L. Garnham and J. Sloper. Effect of age on adult stereoacuity as measured by different types of stereotest. *British journal of ophthalmology*, 90(1):91–95, 2006.

[15] A. Geiger. KITTI Vision. `http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=stereo`, 2012.

[16] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*, volume 2. Cambridge Univ Press, 2000.

[17] A. Hertzmann and K. Perlin. Painterly rendering for video and interaction. In *Proceedings of the 1st International Symposium on Non-photorealistic Animation and Rendering*, NPAR '00, pages 7–12, New York, NY, USA, 2000. ACM.

[18] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):328–341, 2008.

[19] H. Hirschmüller, P. R. Innocent, and J. Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *International Journal of Computer Vision*, 47(1-3):229–246, 2002.

[20] H. Hirschmuller and D. Scharstein. Evaluation of stereo matching costs on images with radiometric differences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(9):1582–1599, 2009.

[21] W. Hong. *A study of fast, robust stereo-matching algorithms.* PhD thesis, Massachusetts Institute of Technology, 2010.

[22] I. P. Howard and B. J. Rogers. *Binocular vision and stereopsis.* Oxford University Press, 1995.

[23] Y. C. Hsieh, D. M. McKeown Jr, and F. P. Perlant. Performance evaluation of scene registration and stereo matching for artographic feature extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):214–238, 1992.

[24] G. Inc. Google AR Glasses. `http://www.google.com/glass/start/`, May 2013.

[25] I. Intel Corporation, Willow Garage. Opencv Doc. `http://docs.opencv.org/java/org/opencv/calib3d/StereoSGBM.html`, 2012.

[26] C. Jerome and B. Witmer. The perception and estimation of egocentric distance in real and augmented reality environments. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, volume 49, pages 2249–2252. SAGE Publications, 2005.

[27] J. C. Kim, K. M. Lee, B. T. Choi, and S. U. Lee. A dense stereo matching using two-pass dynamic programming with generalized ground control points. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1075–1082. IEEE, 2005.

[28] E. Kruijff, J. Swan, and S. Feiner. Perceptual issues in augmented reality revisited. In *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, pages 3–12. IEEE, 2010.

[29] M. A. Livingston. Evaluating human factors in augmented reality systems. *Computer Graphics and Applications, IEEE*, 25(6):6–9, 2005.

[30] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and

measuring ecological statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 416–423. IEEE, 2001.

[31] L. Matthies, T. Kanade, and R. Szeliski. Kalman filter-based algorithms for estimating depth from image sequences. *International Journal of Computer Vision*, 3(3):209–238, 1989.

[32] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang. On building an accurate stereo matching system on graphics hardware. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 467–474. IEEE, 2011.

[33] Microsoft. Kinect. `http://msdn.microsoft.com/en-us/library/hh973078.aspx#Depth_Ranges`, May 2013.

[34] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE TRANSACTIONS on Information and Systems*, 77(12):1321–1329, 1994.

[35] M. Okutomi and T. Kanade. A multiple-baseline stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(4):353–363, 1993.

[36] J. D. Pfautz. *Depth perception in computer graphics*. PhD thesis, University of Cambridge, 2000.

[37] R. Reading. *Binocular vision: Foundations and applications*. Butterworths, 1983.

[38] S. Roy and I. J. Cox. A maximum-flow formulation of the n-camera stereo corre-

spondence problem. In *Computer Vision, 1998. Sixth International Conference on*, pages 492–499. IEEE, 1998.

[39] D. Scharstein. MiddleBury Evaluation. `http://vision.middlebury.edu/stereo/data/`, 2012.

[40] D. Scharstein. MiddleBury Evaluation. `http://vision.middlebury.edu/stereo/eval/`, 2012.

[41] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[42] I. Sobel. Neighborhood coding of binary images for fast contour following and general binary array processing. *Computer Graphics and Image Processing*, 8(1):127–135, 1978.

[43] SoftKinetic. DepthSence Cameras. `http://www.softkinetic.com/en-us/products/depthsensecameras.aspx`, May 2013.

[44] X. Sun, X. Mei, S. Jiao, M. Zhou, and H. Wang. Stereo matching with reliable disparity propagation. In *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2011 International Conference on*, pages 132–139. IEEE, 2011.

[45] J. E. Swan, A. Jones, E. Kolstad, M. A. Livingston, and H. S. Smallman. Egocentric depth judgments in optical, see-through augmented reality. *Visualization and Computer Graphics, IEEE Transactions on*, 13(3):429–442, 2007.

[46] R. Szeliski. *Computer vision: algorithms and applications*. Springer, 2011.

[47] J. P. Wann, S. Rushton, and M. Mon-Williams. Natural problems for stereoscopic depth perception in virtual environments. *Vision research*, 35(19):2731–2736, 1995.

[48] W. M. Wells. Efficient synthesis of gaussian filters by cascaded uniform filters. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (2):234–239, 1986.

[49] J. N. Wilson and G. X. Ritter. *Handbook of computer vision algorithms in image algebra*. CRC press, 2000.

[50] K. Zhang, J. Lu, and G. Lafruit. Cross-based local stereo matching using orthogonal integral images. *Circuits and Systems for Video Technology, IEEE Transactions on*, 19(7):1073–1079, 2009.