Using Texture Features To Perform Depth Estimation

Bhavi Bharat Kotha

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Masters In Science
in
Mechanical Engineering

Alfred L Wicks, Chair
Alan T Asbeck
Steve C Southward

December 15, 2017
Blacksburg, Virginia

Keywords: Computer Vision, Stereo Vision, Depth Estimation, Ranking

Using Texture Features To Perform Depth Estimation

Bhavi Bharat Kotha

Academic Abstract

There is a great need in real world applications for estimating depth through electronic means without human intervention. There are many methods in the field which help in autonomously finding depth measurements. Some of which are using LiDAR, Radar, etc. One of the most researched topic in the field of depth measurements is Computer Vision which uses techniques on 2D images to achieve the desired result. Out of the many 3D vision techniques used, stereovision is a field where a lot of research is being done to solve this kind of problem. Human vision plays a important part behind the inspiration and research performed in this field. A large variety of algorithms are being developed to find the measure of depth of ideally each and every point on the pictured scene giving us a very high spatial resolution as compared to other methods.

Real world needs of depth estimation and the benefits provided by using stereo vision are the main driving force behind this approach. Stereovision gives a very high spatial resolution which is used for obstacle avoidance, path planning, object recognition, etc. Stereovision makes use of image pairs taken from two cameras with different perspective to estimate depth. The two images in the image pair are taken with two cameras from different views (translational change in view) and those two images are processed to get depth information.

Processing stereo images has been one of the most intensively sought after research topics in computer vision. Many factors affect the performance of this approach like computational efficiency, depth discontinuities, lighting changes, correspondence and correlation, electronic noise, etc.

An algorithm is proposed which uses texture features obtained using Laws Energy Masks and multi-block approach to perform correspondence matching between stereo pair of images. This is followed by forming disparity maps to get the relative depth of pixels in the image. An analysis is also made between this approach to the current state-of-the-art algorithms. A robust method to score and rank the stereo algorithms is also proposed. This approach provides a simple way for researchers to rank the algorithms according to their application needs.

Using Texture Features To Perform Depth Estimation

Bhavi Bharat Kotha

General Audience Abstract

There is a great need in real world applications for estimating depth through electronic means without human intervention. There are many methods in the field which help in autonomously finding depth measurements. Some of which are using LiDAR, Radar, etc. One of the most researched topic in the field of depth measurements is Computer Vision which uses techniques on 2D images to achieve the desired result. Out of the many 3D vision techniques used, stereovision is a field where a lot of research is being done to solve this kind of problem. Human vision plays a important part behind the inspiration and research performed in this field. A large variety of algorithms are being developed to find the measure of depth of ideally each and every point on the pictured scene giving us a very high spatial resolution as compared to other methods.

Real world needs of depth estimation and the benefits provided by using stereo vision are the main driving force behind this approach. Stereovision gives a very high spatial resolution which is used for obstacle avoidance, path planning, object recognition, etc. Stereovision makes use of image pairs taken from two cameras with different perspective to estimate depth. The two images in the image pair are taken with two cameras from different views (translational change in view) and those two images are processed to get depth information.

The software tool developed is a new approach to perform correspondence matching to find depth using stereo vision concepts. This software tool developed in this work is written in MATLAB. The tools efficiency was evaluated using standard techniques which have been described in detail. The evaluation was also performed by using the software tool with the images collected using a pair of stereo cameras and a tape measure to measure the depth of an object by hand. A scoring method has also been proposed to rank the algorithms in the field of stereo vision.

Acknowledgments

Exploring a new topic which you have no knowledge about at a graduate level is very challenging and intimidating. It has been a tough year performing this research. There are many individuals who have influenced me in a positive way, giving me hope and pushing me to attain more knowledge along the way. Virginia Tech has been a very good learning instrument which provided me with various opportunities to explore new areas and improve my skills in multiple fields of science and engineering.

Firstly, I want to deeply thank my academic advisor, Dr. Al Wicks, for taking a chance on me and giving me an opportunity to work in the Mechatronics Lab. His guidance has helped me explore new areas in the field of engineering and has also helped me become a better person. This work would not have been possible without his continuous support. His guidance, advisement and also our conversations on various topics are deeply and dearly appreciated.

Secondly, I would like to thank my committee members, Dr. Steve Southward and Dr. Alan Asbeck for their guidance and for being a part of my committee.

Thirdly, I would like to acknowledge my parents, Shanthan and Jhansi, for their trust in me and their continuous support through this endeavor. I would also like to extend my thanks to all my family members and friends who have supported me during this phase.

Finally, I would like to acknowledge all the members of the Mechatronics Lab and friends at Virginia Tech. The mechatronics lab and its members have helped me gain more perspective in life. They have helped me achieve a high level of personal comfort during my stay at Virginia Tech.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the last couple of decades there has been a lot of research and development in the field of engineering. Making various kinds of machines which reduce human effort and help us do our day to day work with more comfort and luxury has been the main focus. Especially in the field of robotics high amount of research is rendered where robots are created and engineered for the main purpose of assisting humans in various tasks. There are many advantages to having robots in our life, for example, robots can perform tasks in environments which are hazardous to humans. "Autonomous vehicles" is another field of interest which has received a lot of attention in the past couple of years. These technical areas which have only been spoken about in the past as fiction; have started becoming reality and are growing at an exponential rate.

One major area of research within these engineering fields of robotics and autonomous vehicles is Computer Vision. As the name indicates, it is a research area which deals on providing vision to computers/machines. It has been inspired from the vision of living beings with a mission to achieve capabilities of performing tasks similar to the visual system of humans. In 1966, the pioneer Dr. Marvin Minsky from MIT gave a task to a student, "Connect a camera to the computer and have the machine describe what it sees". And after 50 years, we are still working on the problem. In the past couple of years, we have been very successful in reinventing the eye. We have been able to capture photo and video through various types of sensors with very high precision. The hardware part of recreating vision has been achieved with great success but we are still struggling in the aspect of the machine understanding the images it has obtained. The theory of obtaining information from 2D images is the main aspect of computer vision. There are many areas in which computer vision is used like agriculture, autonomous vehicles, security, geoscience, robotics, etc.

The areas in computer vision where most of the research is focused upon is object/feature detection, segmentation, structure from motion, 3D modelling, image tracking, etc. Mathematical techniques are being continuously developed in computer vision to achieve these tasks. According to one of the top researchers in computer vision - Richard

Szeliski, vision is one of the most difficult problems because it is an inverse problem. He says that, "we seek to recover some unknowns given insufficient information to fully specify the solution". In computer vision, we try to reconstruct the properties of the world we see in an image like color distributions, shape and light intensities. Computer vision started with digital image processing and has advanced to a great degree.

There has been a lot of research conducted in the field of 3D reconstructions. One of the main parameters required for 3D reconstruction from a 2D image is the depth of the objects in the image. Depth estimation is a very important and highly focused area in the field of computer vision. Many algorithms have been developed as an attempt to achieve this task. All these algorithms so far have only been partially successful. We have yet to see a fully functional algorithm in computer vision which provides us with 100% depth in all the scenarios with the highest spatial resolution. Depth estimation for images in computer vision is obtained by either using a single image of the scene or a pair of images (stereo pair). Many techniques and models have been developed to estimate depth using single images. The research in this thesis deals with depth estimation using a stereo pair of images.

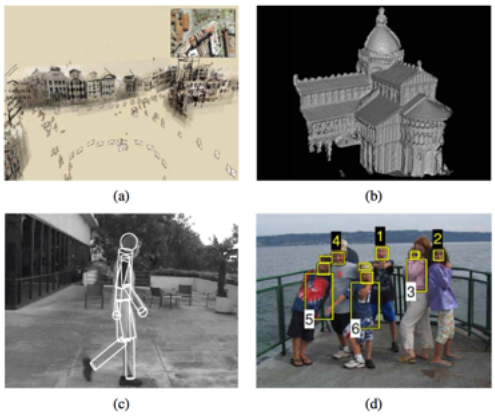


Figure 1.1: (a) Structure from motion (b) Stereo matching
(c) Human tracking (d) Face detection

The image of the world perceived by us due to the projection of light rays on our retina is inherently two-dimensional.The ability of human beings to view the surroundings with a 3D effect is called Stereopsis. Stereopsis in a general sense can be attributed to depth perception, which is responsible in reconstructing the depth dimension in our world. A human being with two fully functional eyes looks at an object and two images which are slightly different from each other get recorded. An often-told simple experiment is to hold your finger in front of your eye in a vertical fashion and close each eye alternatively; you can observe that the position of the finger moves right and left relative to the background. These two images are sent back to the brain for processing and due to the slightly different perceptive in the images, a 3D effect/sensation of depth is created. This kind of vision is called Stereo Vision.

Human brain is very proficient in combining the two different images and providing a view of a scene in 3-D. Human brain has the capacity to produce a sensation of depth which is by estimating the relative distance of similar points in the image pair and the obtained relative distance is called disparity. Disparity is the difference measure of the location of an object in the images produced by our two eyes. This disparity is mainly used by our brain to extract the information of depth from the 2D images provided by the eyes. Stereo vision is totally based on disparity which helps us estimate depth. Of course, there are several other factors like shading/shadows, texture gradients, etc. in our surroundings which help us gauge depth but the most accurate and most important among all is binocular disparity.

If disparity between two images is known perfectly, other cues like shadows, texture, perspective, etc. are not necessary for us to estimate depth. The main problem now becomes matching two similar points in two images with slightly different perspectives. This is commonly known as the Correspondence Problem. By finding the accurate disparities of the two images, i.e., relative depth of each pixel in the image, we can develop a 3D model of the scene from the 2D image pair.

Stereo matching algorithms have come a long way through the years using a lot of different techniques to accomplish the goal. Early algorithms focused on feature based detection, where reliable features with a high level of matching confidence were first gathered and then other pixels in the images were mapped using the reliable features as reference. Now, the state of the art stereo matching algorithms include techniques from machine learning, deep learning, neural networks, etc. providing a way to find stereo matches in the images and also refining the obtained matches through other different techniques.

With so many types of algorithms out in the real world with a single goal of finding disparity by solving the correspondence problem for estimating depth, there is a great need of finding a good classification and ranking method. The establishment of a method which can classify and rank algorithms for a single purpose will aid all researchers and engineers to better approach the subject and give a better idea and understanding of the state-of-the-art.

1.1 Motivation

There is a crucial need for estimating the 3-D location of objects in a scene. We can reconstruct the 3D profile of an object using 3D reconstruction techniques. In the field of computer vision, there is a high level of need in finding the depth information of the objects in the images. It acts as an essential ingredient for many applications such as distance/depth measurement, navigation in autonomous vehicles, tracking and object classifications algorithms in computer vision, various robotic applications, etc. 3D structure can be recovered by using multiple technologies, like LiDAR and Radar ranging techniques, using computer vision algorithms with images and structure from flow, stereo vision, etc.

The main question which comes from dealing with these technologies is, "Which one is the most optimal technology/method to use for our particular application?". From the research and the state-of-the-art technology which we have now, we can conclude that stereo-vision when performed to images compared to the other technologies gives us more depth resolution pixel-wise, i.e., we can obtain a higher spacial resolution of depth. There are still plenty of drawbacks from using stereo-vision but if you need to estimate depth of every pixel in your image then stereo-vision can provide the best estimate of it. Also due to the fact that we can obtain a depth estimate better than other technologies at a very low cost, there has been a lot of research happening in this field. The method of dense reconstruction using stereo matching has been a very preferred method, especially when there is the issue of higher spacial resolution in a small run-time. It provides a rich description of the environment in 3 dimensions. The computational cost for the same level of spacial resolution when compared to other methods is less when it comes to stereo vision.

Every year there are large amount of new methods developed to calculate depth using stereo vision. Getting a complete survey of the existing stereo methods is a very intimidating and a formidable task. There is a great need and value to have a classification and ranking method for other researchers to utilize. Simply listing all the methods being researched or used will not yield any benefit. Hence, it is best to make an evaluation by ranking the algorithms based on factors which effect the applications. Such an approach has been seen in many publications used by authors which resulted in having a dramatic effect on the development of other algorithms[6]. By doing this, researchers can better understand the positives and negatives of different algorithms with a practical approach[7,8]. Even in stereo correspondence, there has been a lot of work done to better categorize and rank algorithms [9,10,11,12]. It is always a better approach to evaluate and rank the algorithms when the inner workings of the algorithms are not considered. The inner workings of an algorithm should be seen as a blackbox and the ranking should be based on the inputs and the outputs. This helps in evaluating the algorithms by eliminating many kinds of biases. For example, if an algorithm uses a better cost function but is still outperformed by a different algorithm due to a different factor, then evaluation becomes more complicated.

Here, an algorithm is developed which computes disparity maps from a pair of stereo images. The algorithm is primarily based on multi-block matching and uses other concepts from texture analysis to surpass the results obtained by the author. This work also show cases a classification and ranking strategy for the state-of-the-art algorithms. This approach will attempt to answer questions about the algorithms based on the factors used to judge their applicability in real world.

1.2 Thesis Objective and Outline

The objective of this effort is to develop an algorithm by using texture analysis to estimate depth using stereo vision for high baseline stereo cameras; to propose a flexible and a robust scoring system for the algorithms in the field. This effort will mainly focus on using texture analysis methods and multi-block approach and use them to do stereo correspondence matching with cameras with high baseline. This process will result is the formation of a disparity map which tells us the relative distance between pixels in the image. This is a novel approach to finding stereo correspondence matching and for cameras with high baseline. This effort also focuses on a scoring system for stereo correspondence matching algorithms which are used for depth estimation. The current literature and surveys only propose comparisons on particular areas in this field and no good way to compare algorithms from different areas in the field. The approach which has been proposed will give a scoring system which can be used by researchers to rank algorithms based on their application requirements.

The following is a small description of the content in the chapters of this thesis:

Chapter 1 gives an introduction to the thesis. It gives a general idea of the different technologies used for depth estimation. The applications which are benefitted from this research are also discussed in this chapter. You can find gentle introductions to various topics relating to this field. An explanation of choosing this research and the motivation behind it is also explained.

Chapter 2 is a detailed literature review in this field. It gives a description of the different approaches, techniques, algorithms used to achieve depth measurements using stereovision. It also explains in detail about the different surveys performed in this field.

Chapter 3 explains the theory behind this research. A review of all the concepts used is performed. A detailed explanation is given about the various techniques used in this research. Mathematical modeling is performed to all the theoretical concepts used and is shown in great details.

Chapter 4 deals with the approach used in this research to obtain depth. The details of the main techniques which lead to depth estimation of the objects in the images is explained. The cost function developed and which is used is also explained here.

Chapter 5 explains about stereo vision algorithms. The steps performed in a stereo vision algorithm is detailed. The proposed evaluation metrics and scoring strategy is talked about in this chapter.

Chapter 6 shows the results obtained by the research performed in this thesis. It also talks about the scope of future work which can be performed for the betterment of the results.

Chapter 2

Literature Review

The need for effective and low cost ways of depth estimation through electronic means without human intervention is increasing day by day. Depth estimation is a very sought out subject of research. There are vast applications in the day to day world which make use of techniques for finding distance of an object from a known frame. Some of the applications which require the estimation of depth from images are robotics, autonomous vehicles, etc. Stereo vision, a concept of computer vision, is one of the techniques which makes this possible. This makes the research performed in correspondence matching a very important sector.

Real world needs of depth estimation and the benefits provided by using stereo vision drive a lot of research in this topic. Stereo vision gives a very high spatial resolution which is used for obstacle avoidance, path planning, object recognition, etc. Depth estimation using stereo vision is one of the most versatile solutions for 3D sensing and reconstruction. One of the major drawbacks for this approach is the high computational effort required to extract depth information from the stereo images which is done by correspondence matching between the images in the stereo pairs.

There has been a lot of work which is performed in the field of stereo correspondence matching and it is important to understand the origin of the work and also the current state-of-the-art. There has been very little work done when it comes to surveys of stereo methods. The last exhaustive surveys date back to about a decade by Barnard and Fischler[1], L.G. Brown[2]. One of the most recent and valuable studies on stereo algorithms was performed by Scharstein and Szeliski [3] which dealt mainly on dense two frame methods.

Almost all the algorithms compute disparity using correspondence matching for stereo vision under a lot of assumptions. These assumptions are usually related to the camera geometry, camera calibration[12,13] and epipolar geometry[15]. There has been a lot of work and research done in this part of stereo vision. It is arguably the most understood part of stereo vision. Here are a few references on stereo camera calibration and rectification [14,15,16,17].

Many methods have been used to achieve this goal. Scharstein and Szeliski[3] in their taxonomy mentioned that most algorithms perform four steps: matching cost computation, cost aggregation, disparity computation/optimization and disparity refinement. There has been a lot of work on development of effective methods of cost aggregation and it even dates back to the 70s [5,6,7,8]. We also need methods which can do real time cost aggregation[11].

Method	Matching Cost	Aggregation	Optimization
SSD (traditional)	Squared Difference	Square Window	WTA
Hannah	Cross-Correlation	Square Window	WTA
Nishihara	Binarized Filters	Square Window	WTA
Kass	Filter Banks	-none-	WTA
Fleet et al.	Phase	-none-	Phase – Matching
Jones and Malik	Filter Banks	-none-	WTA
Kanade	Absolute Difference	Square Window	WTA
Scharstein	Gradient-based	Gaussian	WTA
Zabih and Woodfill	Rank Transform	Square Window	WTA
Cox et al.	Histogram eq.	-none-	DP
Birchfield and Tomasi	Shifted Absolute Difference	-none-	DP
Marr and Poggio	Binary Images	Iterative Aggregation	WTA
Prazdny	Binary Images	3D Aggregation	WTA
Szeliski and Hinton	Binary Images	Iterative 3D aggregation	WTA
Okutomi and Kanade	Squared Difference	Adaptive Window	WTA
Yang et al.	Cross-Correlation	Non-linear filtering	WTA
Shah	Squared difference	Non-linear diffusion	Regularization
Boykov et al.	Threshold Absolute Difference	Connected-component	WTA
Scharstein and Szeliski	Robust Square Difference	Iterative 3D aggregation	Mean-field
Zitnick and Kanade	Squared difference	Iterative aggregation	WTA
Veksler	Absolute difference Average	Adaptive Window	WTA
Quam	Cross Correlation	-none-	Hier. Warp
Barnard	Squared Difference	-none-	SA
Geiger et al.	Squared Difference	Shiftable window	DP
Belhumeur	Squared Difference	-none-	DP
Roy and Cox	Squared Difference	-none-	Graph cut
Bobick and Intille	Absolute Difference	Shiftable window	DP
Boykov et al.	Squared Difference	-none-	Graph Cut
Kolmogorov	Squared Difference	-none-	Graph Cut

Table 2.1: Summary of the taxonomy by Scharstein and Szeliski [3]

Computing cost is one of the most important aspect of stereo correspondence matching. We need a very reliable and robust cost function which can help us match the pixels of both images in the stereo pair. The measure of pixel dissimilarity between locations of pixels in the two images of the image pair is called matching cost [40]. The most common matching costs which are found in the literature include squared intensity differences [19,20,21,22] and absolute intensity differences [18]. Gradient based measures are also used which provide a benefit of being indifferent to camera gains [23,24]. One of the first algorithms with an idea of using a set of windows to improve the accuracy of stereo correspondence is Shiftable windows [3]. Also algorithms were developed with windows of adaptable sizes [25,26,27,28].

There are also global methods which have been developed to perform disparity computation. Energy minimization framework is often used in many of the global methods [29]. Traditional approaches associated with regularization and Markov Random Fields include continuation [30], highest confidence first [31], and mean-field annealing [32]. Dynamic programming is also one more area in global optimization methods for correspondence matching. Edge-based methods [33,34] are the few of the first algorithms which used dynamic programming.

Some of the earliest methods used for disparity computation were involving the cooperative algorithms. They are similar to global optimization methods. They use nonlinear operations and iteratively perform local computations [35,36,37,38]. This method was mostly inspired from computational models of human stereo vision. A new algorithm was developed recently with is a variant to the Maar and Poggio's original cooperative algorithm [39].

Techniques from other areas of computer vision are also being used to develop stereo correspondence matching algorithms. Deep learning using neural networks is one of the most extensively used technique while developing these algorithms these days. Convolutional neural networks are used to train with pairs of stereo images to get a similarity score between pixels and then used the trained networks to predict disparity maps.

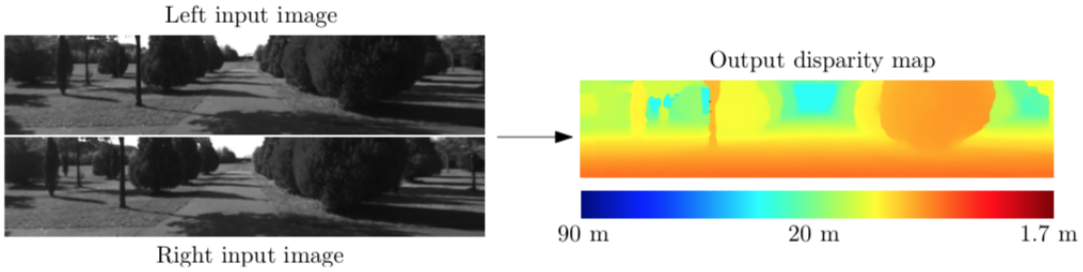


Figure 2.1: Input pair of images from left and the right cameras and their output disparity map [39]

The ground truth depth data from these datasets are mainly used to evaluate the performance of a stereo-algorithm. Apart from providing a way to evaluate performance and compare stereo algorithms, the ground truth data from these datasets can also be used for applying machine learning techniques for improving stereo algorithms in many ways. Image patches and not the entire image can also be trained by deep convolutional neural networks to find a relation between the pair of stereo images [45].

Zbonrar and Le-Cun [46,47] were one of the first to use deep convolutional neural networks to train image patches and performed a cost aggregation, followed by regularization to produce state-of-the-art results at that time. Luo et al. used a Siamese network to perform multi-label classification of disparities by computing local matching costs [48]. There was a lot of need to have large datasets to train these machine learning algorithms. A large

synthetic dataset was created to train a network for disparity estimation by Mayer et. al [49] which helped improve these state-of-the-art algorithms.

Pixel wise image labeling is another interesting and a challenging problem with great significance in the computer vision community. Such a technique can also be used in improving the performance of stereo matching algorithms. Stereo algorithms which use this technique use machine learning in parallel to improve results of disparity estimation. Gidaris and Komodakis [50] use a dense labelling algorithm which detects the initial label estimates and are incorrect and replaces them with new ones. The algorithm they developed also refines the renewed labels by predicting residual corrections.

Shaked and Wolf [51] use a constant highway network and a reflective confidence learning algorithm for the stereo matching problem. They propose a three-step pipeline which uses a highway network architecture for calculating the matching cost at each disparity location which also support multilevel comparison of image patches. As a post processing step they use a deep convolutional neural network for pooling global information from multiple disparities. They develop a new technique called the reflective loss which gives a confidence score and based on this score they better detect outliers so that they can further refine the disparities which they calculated.

Guney and Geiger [52] perform regularization of disparities by performing object-category specific disparity proposals. They perform semantic segmentation to extract objects from the image pairs and refine and regularize the disparities of these objects. Such algorithms take advantage of the object knowledge in the images of the stereo pair. Guney and Geiger also use a superpixel based CRF framework for large objects like cars and show its benefits. A few other superpixel based methods which perform stereo correspondence matching are [53,54]. Wei et al. [55] follows a data-driven approach which directly transfers disparity information from regions with similar appearance in the training data using SIFT flow.

As stated above, researchers are using many techniques from different fields to achieve this goal. Markov random fields is one more area of science which is applied to find disparity maps from stereo image pairs. Zhang and Seitz are researchers which used Markov random field regularization parameters to optimize disparity maps [41] . Conditional random field (CRF) parameters were used by Scharstein and Pal [42] and non-parameteric conditional random field parameters were used by Li and Huttenlocher [43]. Haeusler et al. [44] used learning by employing a random forest approach to estimate the confidence of a traditional stereo algorithm. Apart from the methods mentioned, there are a lot of different approaches used by researches to better their stereo correspondence matching algorithms.

In the last few decades, there have been a lot of algorithms which have been developed by researchers for a single objective, i.e. disparity computation. Each and every algorithm has its own advantages and disadvantages depending on the application for which it is being used. While one algorithm outperforms the other in outdoor scenes; the same algorithm can be computationally very costly which makes it not worthy for video applications. There

have also been very few surveys which have been performed to study and compare these algorithms. Scharstein and Szelinski [3] were one of the few which gave us a valuable taxonomy and evaluation of dense stereo matching algorithms for rectified image pairs. Tombari et al. performed a study on classification and the evaluation of cost aggregation methods for stereo correspondence where they evaluated and ranked algorithms [4]. Some of the other surveys which have been performed in this field are [9,10,11].

Algorithm	Rank Accuracy	Tsukuba NonOcc	Tsukuba Disc	Venus NonOcc	Venus Disc	Teddy NonOcc	Teddy Disc	Cones NonOcc	Cones Disc	Rank Time	Time (mm:ss)	Average Rank
Segment support	1.00	2.28 1	7.50 1	1.21 1	5.88 1	10.99 1	22.01 1	5.42 1	11.83 1	17	33:34	9.00
Adaptive weight	2.50	4.66 3	8.25 2	4.61 3	13.30 4	12.70 2	22.40 2	5.50 2	11.90 2	15	18:14	8.75
Variable Windows	4.00	4.10 2	10.79 3	10.66 13	9.94 2	13.93 3	25.53 3	7.24 3	13.86 3	11	00:25	7.50
Reliability	5.38	5.14 4	18.31 5	3.86 2	11.51 3	16.96 6	30.62 6	13.52 13	21.55 4	16	21:51	10.69
Shiftable Windows	5.63	6.53 7	21.80 8	6.60 5	13.54 5	16.16 5	30.19 5	9.55 4	22.99 6	7	00:15	6.31
Segmentat. based	7.38	8.18 10	18.77 6	8.06 8	20.85 7	15.78 4	29.66 4	13.22 12	24.55 8	2	00:02	4.69
Multi. Win. (25W)*	8.13	6.52 6	21.91 9	6.77 6	21.57 8	18.60 10	33.11 9	11.87 10	23.69 7	9	00:16	8.56
Recursive Adaptive	10.00	9.22 14	26.69 16	8.36 9	14.86 6	18.48 9	32.93 8	11.60 9	24.80 9	14	16:55	12.00
Gradient Guided	10.50	7.51 9	16.20 4	13.07 14	33.10 13	20.39 13	32.82 7	13.67 14	25.60 10	3	00:03	6.75
Mult. Win. (9W)*	10.75	8.51 11	27.59 17	6.47 4	34.30 15	17.57 8	38.04 13	10.75 6	26.60 12	6	00:13	8.38
Mult. Win. (5W)*	11.25	9.36 15	25.74 14	8.57 10	38.65 17	17.11 7	37.45 11	9.86 5	25.33 11	8	00:16	9.63
Mult. Win. (25W)	11.38	6.34 5	24.13 11	9.04 11	29.61 10	20.77 14	36.77 10	14.20 15	27.45 15	10	00:17	10.69
Mult. Win. (9W)	11.50	7.12 8	25.00 13	10.21 12	33.44 14	18.91 12	37.76 12	10.95 7	27.05 14	5	00:09	8.25
Mult. Win. (5W)	13.00	8.94 13	23.55 10	16.33 15	35.56 16	22.29 15	38.09 14	11.13 8	26.99 13	4	00:07	8.50
Fixed Window	14.25	8.66 12	36.67 20	7.05 7	40.53 19	18.68 11	41.95 17	12.79 11	33.32 17	1	< 1 s	7.63
Multiple Adaptive	15.88	15.11 19	32.85 19	16.88 16	22.31 9	25.40 16	39.44 15	21.40 17	29.66 16	18	39:47	16.94
Max Connected	15.88	11.81 17	26.39 15	42.47 20	50.87 20	34.46 18	41.01 16	17.70 16	22.70 5	20	≈ 2 h	17.94
Oriented Rod	16.63	11.29 16	24.21 12	26.33 18	30.09 11	37.68 19	47.94 19	39.60 19	47.88 19	12	12:22	14.31
Oriented Rod*	17.50	15.87 20	27.75 18	26.40 19	30.58 12	30.65 17	42.72 18	30.52 18	41.82 18	13	12:33	15.25
Radial Adaptive	17.50	14.84 18	21.79 7	22.40 17	40.40 18	49.64 20	50.13 20	50.18 20	53.60 20	19	58:52	18.25

Table 2.2: Ranking performed by Tombari et al. [4]

Chapter 3

Theory

3.1 Stereovision

The ability to estimate depth of objects, i.e. 3D reconstruction, from a pair of 2D images based on the relative positions of the object in both the images is called stereo vision or binocular stereopsis. This process has been inspired by how the natural human vision works. This process mainly requires matching features from both the images which are horizontally separated by a baseline distance.

In theory, human eyes collect two images of the surroundings around us. These collected images are processed in the brain to perform stereo vision. The processing in the brain involves matching similar points between the two images and calculating the disparity to estimate depth of each point in the image. The final result is fully achieved with a series of steps which begin in the primary visual cortex and requires a series of computations in various areas of the visual cortex. J.C.A. Read from the institute of neuroscience has performed an extensive study in the process of stereo vision [13] in our eyes.

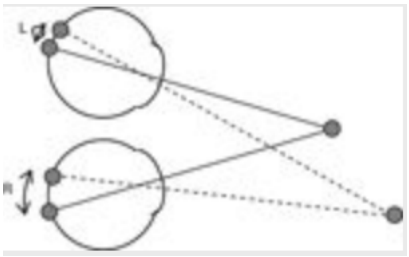


Figure 3.1: Human Vision: showing the disparity between the images of an object in the left and the right eyes.

We try to replicate this phenomenon by using a pair of cameras. Below is a figure shown, describing the theory of matching similar points in an image using a pair of stereo

cameras.

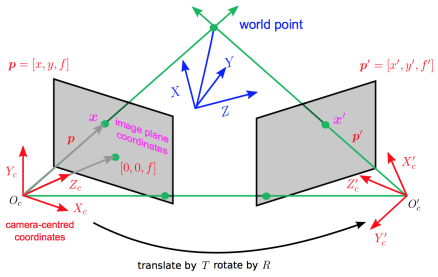


Figure 3.2: Stereo-Vision using a pair of cameras[14]

We usually simplify the problem more by placing the cameras on a baseline with a view in the same direction (parallel views). This way we eliminate the rotation aspect of the camera. Below is a figure showing how stereo vision is usually performed and is followed by mathematical modeling of the problem.

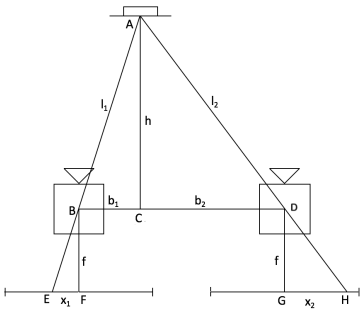


Figure 3.3: Stereo-Vision using a pair of cameras with a parallel view

In the above figure, two cameras, B & D, are positioned with a distance of b (baseline) between them. The cameras are also placed such that they have a parallel view of the real world scene. E and H are the two points on the image plane depicting the point A from the real world scene. As you can see, x_1 & x_2 are the distances between the image points and the image centers of the images produced by cameras B & D respectively.

Let,
$b = b_1 + b_2$ = baseline - the distance between the camera centers with parallel views.
f = focal length of cameras.
h = depth of the point A (perpendicular distance) from the cameras.
$d = x_1 + x_2$ = disparity of the image point.
l_1 & l_2 = distance between the object in the real world scene and the camera center.

$$d = \text{EF} + \text{GH}$$

$$= \text{BF}*(\tfrac{\text{EF+GH}}{\text{BF}})$$
$$= \text{BF}*(\tfrac{\text{EF}}{\text{BF}} + \tfrac{\text{GH}}{\text{BF}})$$
$$= \text{BF}*(\tfrac{\text{EF}}{\text{BF}} + \tfrac{\text{GH}}{\text{GD}}) \ (\because \text{BF} = \text{GD} = f)$$

We know that,

$\triangle$ABC and $\triangle$ BEF are similar triangles $\implies \frac{\text{EF}}{\text{BF}} = \frac{\text{BC}}{\text{AC}}$

$\triangle$ACD and $\triangle$ DGH are similar triangles $\implies \frac{\text{GH}}{\text{GD}} = \frac{\text{CD}}{\text{AC}}$

$$\implies d = \text{BF}*(\tfrac{\text{BC}}{\text{AC}} + \tfrac{\text{CD}}{\text{AC}})$$
$$= \text{BF}*(\tfrac{\text{BD}}{\text{AC}}) \ (\because \text{BC} + \text{CD} = \text{BD})$$
$$= \text{BF}*(\tfrac{b}{h}) \ (\because \text{BD} = b)$$
$$= f * (\tfrac{b}{h}) \ (\because \text{BF} = f)$$
$$\therefore \text{depth} = h = \tfrac{f*b}{d}$$

From the above derived equation, we can deduce that,

$$h \propto \frac{1}{d}$$

where,
h = depth of the point in a real world scene from the cameras.
d = disparity of the point formed by the pair of the cameras.

Similarly we can find the relative depth between two points in a scene as follows,

Let the two points be Point 1 and Point 2,

$$\implies \frac{h_1}{h_2} = \frac{d_2}{d_1}$$

Here,
h_1,h_2 are the depths of two points in a real world scene from the cameras.
d_1,d_2 are the disparity of the two points in the image formed by the pair of cameras.

As you can see from the above figures and derivations, we search for our points in our stereo pair of images only in the horizontal directions along a single line. Theoretically, here we place both the cameras at the same level. Practically, placing the cameras at the same level without any error (tolerance = 0%) is impossible. To achieve this state, we perform image rectification before we do correspondence matching of points in the images. For us to perform image rectification, we need to also perform camera calibration to find the intrinsic and extrinsic parameters of the cameras. Also, we perform image rectification for stereo images by finding what is known as Epipolar Geometry for the two images.

3.2 Camera

A device which is used to capture images of the real world is called a Camera. The concept of a camera evolved from "Camera Obscura", which is Latin for "Dark Room", invented by Alhazen (Ibn Al-Haytham) in the year 1500. The "Pinhole Image" is another term used to describe this phenomenon. When an image is obtained due to the light being projected from a scene through a small hole in a screen (like a wall) on the surface opposite to the hole is called a Pinhole Image. The image obtained due to this natural optical phenomenon is reversed, i.e., an inverted image of the scene is formed. To obtain a clear pinhole image, the surroundings of the image being projected has to be relatively dark and for this reason many of these experiments were performed in a dark room in the past.

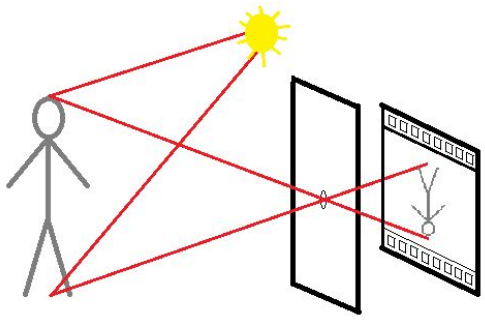


Figure 3.4: Pinhole Camera with its inverted image [15]

Cameras as stated above started with camera obscura and evolved through ages passing through different technologies like dry plates, films, etc. and finally reached the modern day digital cameras. The first ever photographic image was constructed by the inventor Nicrphore Niepce in 1816 using different chemicals. His process of capturing an image took around eight hours and would fade away depending on the material used [16]. Later in 1839, Louis Daguerre, the former partner of Niepce, developed the first photographic process using a silver-plated sheet of copper and iodine vapor and named it the "daguerreotype". Simultaneously, a different process was developed by Henry Fox Talbot, and was named the "calotype". The first patent issued for a camera in America was in 1840 to Alexander Wolcott for his camera design. In 1861, Oliver Wendell Holmes invented the first stereoscope viewer. The usage of dry plates started in 1857 by Desire van Monckhoven which led to the invention of the gelatin dry plate in 1871 by Richard Leach Maddox. This is when the wet plate process had competition in speed and quality.

In 1885, George Eastman invented the first flexible photographic roll film with a paper base. In 1889, he developed and patented the first plastic roll film which was made from a highly flammable nitrocellulose which is now usually called "nitrate film". He named the first camera he developed the "Kodak" which he started selling in 1888. The first design was a simple fixed focal length and a single shutter speed camera. As time passed, the world of cameras saw the technology develop from 35mm cameras, TLR (Twin-lens reflex

cameras), SLR (single lens reflex cameras) , instant cameras, auto-exposure cameras and finally reached the era of digital cameras.

The digital cameras mainly differ from their analog predecessors in the way the captured images are stored. All the cameras before digital cameras, used some sort of film or a chemical solution to capture images whereas all the digital cameras capture and store images in some sort of digital memory like a memory card. The first electronic image capturing array was the charge-coupled device (CCD) array with a resolution of 800x800 pixels used in a satellite. On 6 September 1968, the first flat screen target to receive and store an optical image on a matrix was constructed and filed for a patent under the name of "All Solid State Radiation Imagers" by Edward Stupp, Pieter Cath and Zsolt Szilagyi who worked at Philips Labs in New York. This matrix was composed of an array of photodiodes which were connected to a capacitor to form an array of two terminal devices connected in rows and columns. The patent was granted on 10 November 1970. The first digital camera was built by Steven Sasson, an engineer at Eastman Kodak who used a solid state CCD image sensor chip which was developed by Fairchild Semiconductor. Since 2003, digital cameras have outsold film cameras and Kodak officially announced in January 2004 that they would no longer sell Kodak branded film cameras.

As stated above, through the evolution of cameras, the image capture instrument moved from films to image sensors. A sensor that detects and conveys the information that constitutes an image is called an image sensor. The working principle of an image sensor is based on the conversion of the variable attenuation of light waves which come form the different objects in the scene into electrical (current) signals. These current signals convey information about the intensity of the light coming from the object as well as the properties of light. The waves received by the sensor can be in the form of visible light or also any other kind of electromagnetic radiation. The following image sensors are currently used: semiconductor charge coupled devices (CCD), complementary metal oxide semiconductors (CMOS), N-type metal oxide semiconductor (NMOS). CMOS sensors are used most extensively among these types as they have faster performance rate and lower power consumption. Some of the parameters which are used to evaluate a image sensor include dynamic range, signal to noise ratio, etc. There are also many types of color image sensors. The main ones are the Bayer filter sensor and Foveon X3 sensor. A demosaicing algorithm is used to convert the raw color image data into a full color image.

A camera lens is one more part of the camera which contributes to the performance of the camera and dictates the type and the properties of image captured by the camera. The attributes like focal lengths and apertures of the camera are lens specific properties used in the camera. In a camera, usually, a compound lens made up of a number of optical lens elements is used. A compound lens is preferred over a simple lens to correct the various optical aberrations that arise. Glass is mostly used to make lens. Other materials like fluorite, germanium, plastics like acrylic, quartz glass, etc. are also used sometimes to make different kinds of lenses. To regulate the amount of light that passes through the lens, lenses are usually equipped with an aperture adjustment mechanisms like an iris diaphragm.

Aperture and focal length are the two fundamental parameters of a camera which are actually properties of the lenses used in the camera. The focal length of a lens dictates the magnification of the captured image on the image plane. The aperture of the lens dictates the intensity of light from the world scene which falls on the sensor. The angle of view of a camera is also dependent on the focal length of the camera. Usually, a wide field of view is obtained by using lenses with short focal lengths and lenses with high focal length have a short field of view. The maximum usable aperture of a lens also depends on the focal length. It is specified with a dimensionless quantity called the f-number. Focal ratio or f-number is the ratio between the lens's focal length and the effective aperture.

3.3 Camera Model and its Parameters

To understand and get the properties of a camera we mathematically model cameras so that we can use the obtained properties in various aspects of computer vision applications. The main purpose of a camera model is to obtain geometry of projection of 3D points, curves, and surfaces of a real world scene onto a 2D image plane. There are many camera models which have been developed for this purpose like the thin lens model, the pinhole camera model, etc. The pinhole camera model is generally widely used to model and give a mathematical relationship between the coordinates of a point on the image plane and its real world coordinates in the three-dimensional space. The pinhole camera model does not take into consideration blurring of images due to not being focused, geometric distortions, and any aperture sizes with a finite radius.

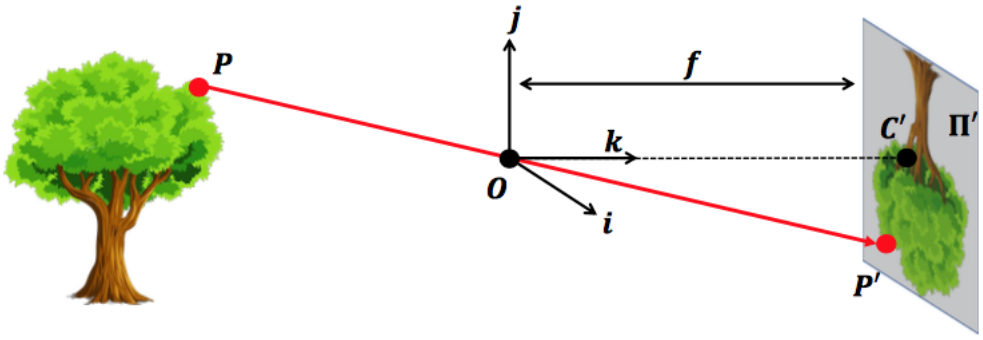


Figure 3.5: Mathematical construction of a Pinhole Camera Model [17]

We use the pinhole camera model to get an approximation of the geometric properties of the images made by our cameras. The matrix which represents the pinhole camera parameters is called as the camera matrix. This camera matrix is used to map a point in the real world to its projection on the image plane, i.e., map the point from world coordinates to pixel coordinates. The camera matrix consists of two parts: intrinsic parameter matrix

and the extrinsic parameter matrix.

Let,

Homogeneous Pixel Coordinates on the Image Plane = $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

Homogeneous Real World Coordinates = $\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$

Converting from Homogeneous Coordinates:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \implies \left(\frac{x}{w}, \frac{y}{w} \right)$$

$$\begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \implies \left(\frac{x}{w}, \frac{y}{w}, \frac{z}{w} \right)$$

P = Camera Matrix

$$\implies w \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = P * \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Also, Camera Matrix = Intrinsic Parameter's Matrix $*$ Extrinsic Parameter's Matrix

$\implies$ P = Intrinsic Parameter's Matrix $*$ Extrinsic Parameter's Matrix

Let,
Intrinsic Parameter's Matrix = K
Extrinsic Parameter's Matrix = $\begin{bmatrix} R & T \end{bmatrix}$

$$\implies w \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = K * \begin{bmatrix} R & T \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\implies \begin{pmatrix} \text{2D} \\ \text{point} \\ \text{(3x1)} \end{pmatrix} = \begin{pmatrix} \text{Camera to} \\ \text{pixel coord.} \\ \text{trans. matrix} \\ \text{(3x3)} \end{pmatrix} \begin{pmatrix} \text{Perspective} \\ \text{projection matrix} \\ \text{(3x4)} \end{pmatrix} \begin{pmatrix} \text{World to} \\ \text{camera coord.} \\ \text{trans. matrix} \\ \text{(4x4)} \end{pmatrix} \begin{pmatrix} \text{3D} \\ \text{point} \\ \text{(4x1)} \end{pmatrix}$$

3.4 Lens Distortion

The optical aberration which causes the lenses to deviate from rectilinear projection is called as lens distortion. This kind of optical aberration generally deforms and bends straight lines in images. This error in the images is completely a lens error. The pin hole camera model does not account for lens distortion because it does not have a lens in its model and also lens distortion is a non-linear parameter while the pin-hole camera model is a linear model. The two main types of lens distortions are Radial Distortion and Tangential Distortion.

3.4.1 Radial Distortion

The distortion caused due to the imperfections in curvature of the lens is called radial lens distortion. This distortion is symmetric is nature. Radial distortions are usually of two types: Barrel distortion and Pincushion distortion. Both of these distortions are quadratic in nature, i.e. the distortion increases proportional to the square of the distance from the center of the image. Barrel distortion is a type of optical aberration which is said to have caused when you see straight lines curving inwards into a shape of a barrel. When the size of the image sensor is smaller than the lens then this distortion occurs because the image has to be squeezed to fit on the sensor. Fisheye lenses which are used to take panoramic photos utilize this kind of distortion to its advantage.

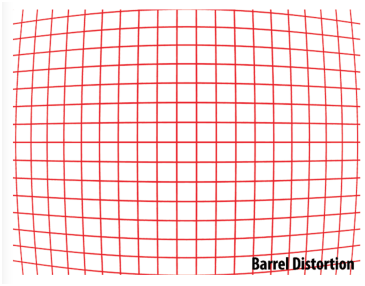


Figure 3.6: Barrel Distortion [18]

When straight lines are caused to curve outwards from the center in images, then we can observe Pincushion distortion. This usually occurs because the image magnification increases towards the edges of the frame from the optical axis. Here, this happens because the field of view of the lens is smaller than the size of the image sensor.

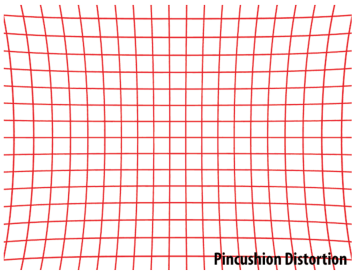


Figure 3.7: Pincushion Distortion [18]

Distortion which is caused by a combination of both barrel and pincushion distortions is called as Mustache distortion or wavy distortion or complex distortion. The straight lines are curved outward at the corners (Pincushion distortion) and inwards at the center (Barrel distortion) of the image.

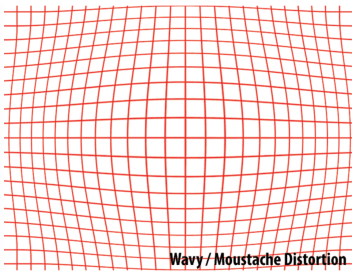


Figure 3.8: Pincushion Distortion [18]

The radial distortion coefficients model this type of distortion.

$$x_{distorted} = x(1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6)$$
$$y_{distorted} = y(1 + k_1 * r^2 + k_2 * r^4 + k_3 * r^6)$$

Also, $r^2 = x^2 + y^2$

where,
x, y - Undistorted pixel locations
k_1, k_2, k_3 - Radial distortion coefficients of the lens.

3.4.2 Tangential Distortion

When the lens is not parallel to the image sensor then a tangential distortion is produced.

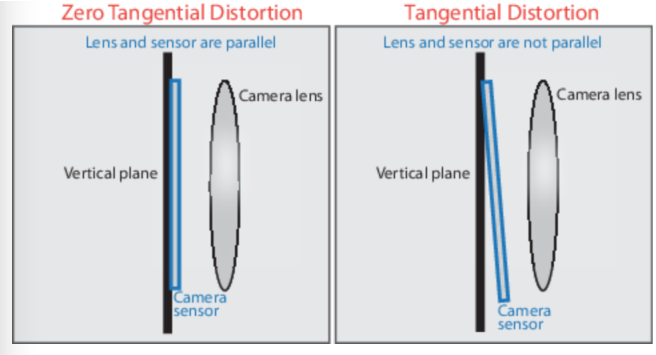


Figure 3.9: Tangential Lens Distortion [19]

We fix this kinds of distortions using the coefficients model as follows:

$$x_{distorted} = x + [2 * p_1 * x * y + p_2 * (r^2 + 2 * x^2)]$$
$$y_{distorted} = y + [2 * p_1 * (r^2 + 2 * y^2) + p_2 * p_2 * x * y]$$

Also, $r^2 = x^2 + y^2$

where,
x, y - Undistorted pixel locations
p_1, p_2 - Tangential distortion coefficients of the lens.

3.5 Camera Calibration

The process of estimating the parameters, i.e. intrinsic and extrinsic parameters, of a camera model is called as Geometric Camera Calibration or Camera Resectioning. This process also involves corrections due to lenses. In the early stages of many algorithms concerning computer vision, camera calibration is performed. The estimated parameters can be used to correct lens distortions, determine the location of the objects, measure the size of an object, etc. There are many methods which have been developed over the ages to perform camera calibration.

There is a loss of dimension when an image is captured by a camera, i.e. a 3D scene is converted into a 2D image. Each pixel on the obtained image belongs to a point in the 3D scene. Camera calibration helps in determining the relationship between the 2D image pixel and the 3D point in the real world scene. Apart from obtaining the image camera matrix, there are also corrections which have to be performed due to lens distortions. In stereo vision, we calculate the 3D world coordinates of a point from a 2D image, hence camera calibration is often performed in this stereo vision application.

3.5.1 Intrinsic Parameters

The parameters of the camera related to the focal length, format of the image sensor and the optical center of the camera lens form the intrinsic parameter matrix. There are a total of 5 parameters in the Intrinsic matrix of the camera: focal length, optical center (x, y), skew, aspect ratio (scale factor). We usually use number of pixels as the distance unit in all the above parameters.

$$\implies K = \begin{bmatrix} \alpha_x & \gamma & u_0 \\ 0 & \alpha_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

where,
f = Focal Length of the lens
Aspect Ratio = $m_x : m_y$
$\alpha_x = m_x * f$ = Scale Factor (in x direction) $*$ Focal Length
$\alpha_y = m_y * f$ = Scale Factor (in y direction) $*$ Focal Length
(u_0, v_0) = Optical Center (the principal point) of the Camera/Lens (the center of the image)
γ = Skew Coefficient between the x and y axis.

There are some other intrinsic parameters like lens distortion. The lens distortion is a nonlinear parameter which cannot fit into the linear pinhole camera model. There are other non-linear optimization techniques which are used for optimizing the camera in this aspect.

3.5.2 Extrinsic Parameters

Extrinsic parameters deal with the coordinate system transformations from camera coordinates to the world coordinates. The extrinsic parameter matrix consists of a rotation, R, and a translation, T. Intuitively it tells us the position of the camera center in the world coordinates. The translation matrix tells us the position of the origin of the world coordinate system relative to the coordinates of the camera system. Similarly, the rotation matrix gives the matrix required to relatively align both the coordinate systems.

$$\implies \text{Extrinsic Parameter Matrix } \begin{bmatrix} R & T \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix}$$

The T matrix is often misunderstood to be the position matrix for the camera. The position of the camera can be determined using the following equation:

$$\implies C = -\begin{bmatrix} R^{-1} & T \end{bmatrix} = -\begin{bmatrix} R^T & T \end{bmatrix}$$

3.5.3 3D Rotation of Points

A counter-clockwise rotation of points around the coordinate axes is given by the following rotation matrices:

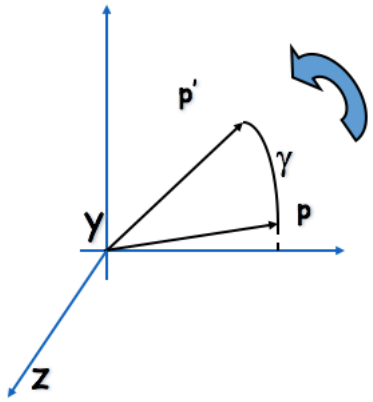


Figure 3.10: Rotation of a Point

$$\text{Rotation w.r.t X-Axis} = R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha \\ 0 & sin\alpha & cos\alpha \end{bmatrix}$$

$$\text{Rotation w.r.t Y-Axis} = R_y(\beta) = \begin{bmatrix} cos\beta & 0 & sin\beta \\ 0 & 1 & 0 \\ -sin\beta & 0 & cos\beta \end{bmatrix}$$

$$\text{Rotation w.r.t Z-Axis} = R_z(\gamma) = \begin{bmatrix} cos\gamma & -sin\gamma & 0 \\ sin\gamma & cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

3.6 Zhang's Camera Calibration

This method of camera calibration is one of the most commonly used techniques in the field. One of the most advantageous aspects of using this technique is, one does not need to have knowledge of the 3D geometry to perform this technique. This technique only uses a camera which is used to observe a planar pattern shown at different orientations. Orientation change can be achieved either by moving the camera or the pattern while performing this process. This method mainly uses calibration points from the images with different orientations of a similar pattern and the correspondence between these points. The minimum number of images (orientations of the pattern) required to perform this method is three but having more images gives us more accurate results. [20]

The procedure for calibrating using Zhang's method is as follows:

1. By either moving the model or the camera (or both) we obtain images $I_0, I_1, ...I_{M-1}$ in different orientations/views.

2. Assuming 1:1 correspondence with the points on the planar pattern, N sensor points $u_{i,0}, ..., u_{i,j}, ..., u_{i,N-1}$ are obtained from each image I_i $(i = 0, ..., M-1)$.

3. From the observed points, the linear mappings are obtained which link the model points and the observed 2D image points, i.e. the homographies $H_0, ..., H_{M-1}$.

4. From the Homgraphies H_i, all the intrinsic parameters of the camera can be estimated using a closed-form (linear) solution.

5. We can estimate the extrinsic 3D parameters, i.e. R_i, t_i once the intrinsic properties of the camera are known.

6. By using linear least-square minimization, we can estimate radial distortion parameters k_0, k_1.

7. Finally, using all M views, we refine the parameters by performing non-linear optimization. The estimated parameter values are used as the initial guess.

3.7 Epipolar Geometry

Epipolar geometry is the most fundamental geometric relationship between two cameras with different perspectives. There are a number of relations which can be derived between the 3D points of a real world scene and the 2D points on the image from two different camera views of a 3D scene. These relationships are derived using a pinhole camera model and the concepts of epipolar geometry. The intrinsic projective geometry between two views is given by the epipolar geometry. The epipolar geometry depends only on the camera's internal parameters and relative pose and does not depend on the structure of the scene.

The main motivation into developing epipolar geometry is the search for corresponding points in stereo matching. The epipolar geometry between two views is essentially the geometry of the intersection of the image planes with the pencil of planes having the baseline as axis. The baseline of the planes is the line joining the camera centers.

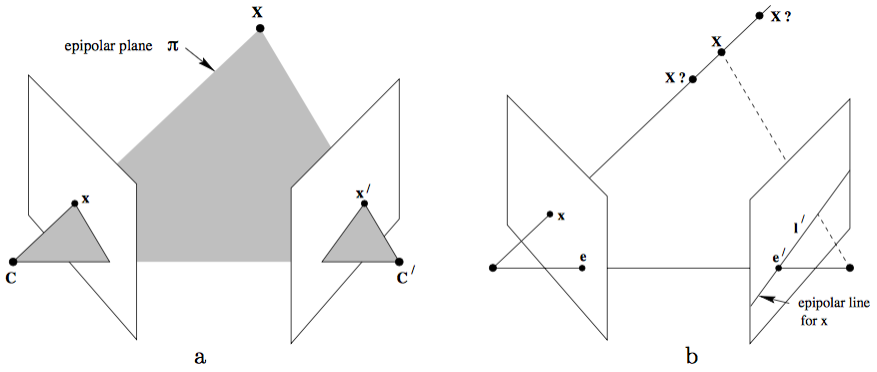


Figure 3.11: Point Correspondence Geometry [21]

The above figure shows you two cameras with their centers placed at points C and C' viewing the point X from two different perspectives. The image planes are actually at the back of the camera centers/focal centers, however, here the problem is simplified for our convenience by depicting a virtual image plane in front of the optical center of the camera. Points x and x' are the image points on the two image planes of the point X from the real world scene. Let the plane formed by joining the points C, C' and X by denoted by π. The image points x and x' also fall on the plane π.

In correspondence matching in stereo vision, we look for x' by trying to match the point with points on the other image by comparing it to x or vice versa. Here, let us consider x as the known point and we are searching for x'. As shown in the above figure (b), we can form the epipolar plane π by joining the points C, C' (baseline) and C, X. We know that x' lies on the plane π and also the image plane made by the camera C'. Let the intersection of plane π and the image plane be the line l'. Hence now we can narrow the search for the point x' on to a single line l'.

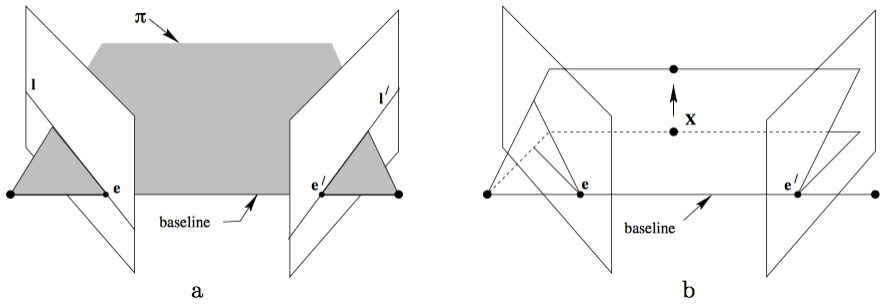


Figure 3.12: Epipolar Geometry [21]

The following is the terminology used in studies relating to epipolar geometry:

1. Baseline: The line joining the two optical centers of the cameras is called the baseline.

2. Epipole: The point of intersection of the baseline and the image plane is called the epipole.

3. Epipolar Plane: A plane containing the baseline is called the epipolar plane.

4. Epipolar Line: The line formed from the intersection of the epipolar plane and the image plane is called the epipolar line.

In real world scenarios, we do now know the exact 3D location of the point X, but we still can determine its projection in one of the image planes x. We can also find the camera locations, orientations and the camera matrices by performing camera calibrations. By knowing the camera locations C & C' and the image point x, we can define the epipolar plane, π, which in turn can help us derive the epipolar lines, l'. We now know that the projection of X on the second image plane, x', lies on the epipolar line l'. This basic knowledge of the epipolar geometry helps us understand and create a constraint between the image point pairs from the two camera planes without having to know the 3D location of the point X.

3.7.1 Essential Matrix

Every point in one image can be mapped on to an epipolar line in the second image which consists of the corresponding matching point. The relationship between this point and the line can be formed using the essential matrix. The concept of Essential Matrix was first introduced by Longuet-Higgins in 1981.

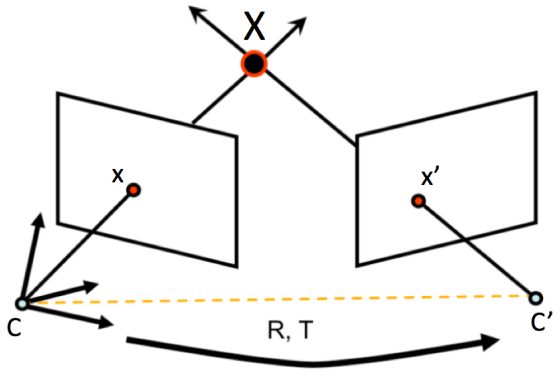


Figure 3.13: Transformations to find Essential and Fundamental Matrices [22]

Consider a camera matrix decomposed as $P = K[R\,|\,t]$. Let us define M and M' as the projection matrices which use the camera matrix and represent the projection matrices that map 3D points into their respective 2D image plane locations. As shown in the figure above, let us assume that the world reference system is associated to the first camera with

the second camera offset first by a rotation R and then by a translation L. From this, we can write the camera projection matrices as follows:

$$M = K \begin{bmatrix} I & 0 \end{bmatrix} \qquad\qquad M' = K' \begin{bmatrix} R & L \end{bmatrix}$$

For our convenience, let us assume that both the cameras have the same intrinsic properties and are canonical cameras, i.e. $K = K' = I$. This assumption gives us the following equations:

$$M = \begin{bmatrix} I & 0 \end{bmatrix} \qquad\qquad M' = \begin{bmatrix} R & L \end{bmatrix}$$

From basic geometry, we can also derive the location of x' in the first cameras reference system as follows:

$$R^T(x' - L)$$

The vectors $R^T(x' - L)$ and $R^T L$ lie on the epipolar plane, π. By taking a cross product of these two vectors, we get a resulting vector which is normal to the epipolar plane.

$$R^T L \times (R^T x' - R^T L) = R^T L \times R^T x' = R^T (L \times x')$$

This resulting vector is also normal to the point x which lies on the epipolar plane, π. We know that the dot product of a vector and a point which are normal to each other is zero. By applying this constraint, we get the following:

$$(R^T(L \times x'))^T x = 0$$

$$(L \times x')^T R x = 0$$

From linear algebra, we can introduce a different and compact expression for the cross product: we can represent the cross product between any two vectors a and b as a matrix-vector multiplication:

$$a \times b = \begin{bmatrix} a & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{bmatrix} \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} = [a_\times] b$$

Using the above expression, we can convert the cross product term into matrix multiplication term and we get the following:

$$([L_\times] x')^T R x = 0$$

$$x'^T [L_\times]^T R x = 0$$

$$x'^T [L_\times] R x = 0$$

Therefore, the Essential Matrix, $E = [L_\times] R$. Substituting E in the above equation, we get the following:

$$x'^T E x = 0$$

The essential matrix helps us relate the two points x and x' in the two image planes. The essential matrix also helps us to compute the epipolar lines associated with x and x' as follows:

$$l' = Ex \qquad\qquad l = E^T x'$$

The essential matrix is a 3×3 matrix with a rank of 2, 5 degrees of freedom and is singular.

3.7.2 Fundamental Matrix

The fundamental matrix is the algebraic representation of epipolar geometry. The fundamental matrix also maps a point in one image plane to its respective epipolar line in the other image plane. The nature of this map turns out to be a correlation (singular), which is a projective mapping from points to lines. This mapping between the point to the line is represented by the fundamental matrix, F. The concept of fundamental matrix was introduced by Faugeras and Luong in 1992. We derived a similar relationship above between x and x' under the assumption that both the cameras are canonical which gave us the essential matrix. Now let us derive a more general relationship where the cameras are no longer canonical. Now the projection matrices are given by the following expressions:

$$M = K \begin{bmatrix} I & 0 \end{bmatrix} \qquad\qquad M' = K' \begin{bmatrix} R & L \end{bmatrix}$$

Let us define $\hat{x} = K^{-1}x$ and $\hat{x}' = K'^{-1}x'$ be the projections of X to the corresponding camera images if the cameras were canonical.By using the expression, we derived for the essential matrix, we get the following:

$$\hat{x}'^{T}[L_{\times}]R\hat{x} = 0$$

By substituting the values of $\hat{x}$ and $\hat{x}'$, we get the following:

$$x'^{T}K'^{-T}[L_{\times}]RK^{-1}x = 0$$

Therefore, the Fundamental Matrix, $F = K'^{-T}[L_{\times}]RK^{-1} = K'^{-T}EK^{-1}$. Substituting F in the above equation, we get the following:

$$x'^{T}Fx = 0$$

Also,

$$x'^{T}K'^{-T}EK^{-1}x = 0$$

Similarly we can calculate the corresponding epipolar lines for x and x' as follows:

$$l' = Fx \qquad\qquad l = F^{T}x'$$

The fundamental matrix has 7 degrees of freedom.

3.7.3 The Eight-Point Algorithm

It is very unlikely that we would have knowledge of the Fundamental matrix which is a matrix product consisting of the camera parameters. However, it is possible to estimate the fundamental matrix without knowing the intrinsic and extrinsic parameters of the cameras. This method of estimating the fundamental matrix was proposed by Longuet-Higgins in 1981 and was extended by Hartley in 1995. This method is known as the Eight-Point Algorithm.

As the name suggests, the Eight-Point Algorithm assumes that a set of at least 8 pairs of corresponding points between two images are available.

Let each pair of points be denoted as follows: $x_i = (u_i, v_i, 1)$ and $x'_i = (u'_i, v'_i, 1)$. From the above derivation of the fundamental matrix, we can write the following equation: $x_i'^T F x_i = 0$. This constraint is a scalar equation and hence has only 1 degree of freedom. We can rewrite this equation as follows in the expanded matrix form:

$$\begin{bmatrix} u_i u'_i & v_i u'_i & u'_i & u_i v'_i & v_i v'_i & v'_i & u_i & v_i & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

$$\implies \begin{bmatrix} u_1 u'_1 & v_1 u'_1 & u'_1 & u_1 v'_1 & v_1 v'_1 & v'_1 & u_1 & v_1 & 1 \\ u_2 u'_2 & v_2 u'_2 & u'_2 & u_2 v'_2 & v_2 v'_2 & v'_2 & u_2 & v_2 & 1 \\ u_3 u'_3 & v_3 u'_3 & u'_3 & u_3 v'_3 & v_3 v'_3 & v'_3 & u_3 & v_3 & 1 \\ u_4 u'_4 & v_4 u'_4 & u'_4 & u_4 v'_4 & v_4 v'_4 & v'_4 & u_4 & v_4 & 1 \\ u_5 u'_5 & v_5 u'_5 & u'_5 & u_5 v'_5 & v_5 v'_5 & v'_5 & u_5 & v_5 & 1 \\ u_6 u'_6 & v_6 u'_6 & u'_6 & u_6 v'_6 & v_6 v'_6 & v'_6 & u_6 & v_6 & 1 \\ u_7 u'_7 & v_7 u'_7 & u'_7 & u_7 v'_7 & v_7 v'_7 & v'_7 & u_7 & v_7 & 1 \\ u_8 u'_8 & v_8 u'_8 & u'_8 & u_8 v'_8 & v_8 v'_8 & v'_8 & u_8 & v_8 & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

We can write the above equation as follows:

$$Wf = 0$$

where W is an $N \times 9$ matrix derived from $N \geqslant 8$ correspondences and f is the fundamental matrix.

It is always better to have more than eight correspondences to get more accurate results. We can use Single Value Decomposition (SVD) to find the solution of this system of homogeneous equations. SVD will give us a estimate of the Fundamental matrix $\hat{F}$, which may have full rank. However, we know that the true fundamental matrix has rank 2. Hence, we have to search for the solution which is the best rank-2 approximation of $\hat{F}$. To do so, we solve the following optimization problem:

$$\min_F ||F - \hat{F}||_F$$

$$\text{subject to } |F| = 0$$

Then, this problem is again solved by SVD, where $\hat{F} = U\Sigma V^T$,then the best rank-2 approximation is found by:

$$F = U \begin{bmatrix} \Sigma_1 & 0 & 0 \\ 0 & \Sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T$$

3.8 Image Rectification

The image rectification problem in stereo-vision is the process of computing two homographies which can be applied to a pair of image planes to make them have a parallel perspective to each other. This helps in better correspondence matching of points in the pair of images. By performing rectification and making the two given images parallel, we make the corresponding matching points have the same y-coordinate, i.e. $v_i = v_i'$ for x and x'.

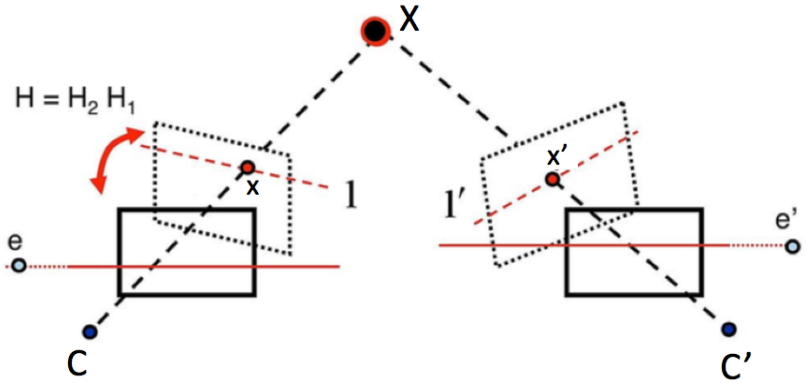


Figure 3.14: Image Rectification [22]

We do not require the knowledge of the two camera matrices K and K' nor the relative transformations R, T between the two images for rectifying a pair of images. We can estimate the fundamental matrix using the eight-point algorithm and use that to find the required homographies.

We can calculate the epipolar lines l_i and l_i' for each of the correspondence pair x_i and x_i' which can in turn be used to calculate the epipoles e and e' for each image in the image pair. The epipoles are the points of intersection of the epipolar lines in the image. Because of noise, we will notice that all the epipolar lines do not intersect at a single point. Hence, we can compute the epipole by minimizing the least squared error of fitting a point to all the epipolar lines. Each epipolar line is represented as a vector l where all the points on the line are in the set $\{x|l^T x = 0\}$. Using this, we can form a system of linear equations and

solve for the epipole e using SVD.

$$\begin{bmatrix} l_1^T \\ \vdots \\ l_n^T \end{bmatrix} e = 0$$

The epipoles which we find using the above equations, we can see that they are not points of infinity. Now, we need to find two homographies to map these epipoles e and e' to infinity along the horizontal axis.

We design the homography such that it acts like a transformation that applies a translation and rotation to the points near the center of the image. Lets start by finding the homography for epipole e' to a point on the horizontal axis at infinity $(f, 0, 0)$. First, translate the second image such that its center is at (0,0,1) in homogeneous coordinates. We can do this by applying the following transformation (translation) to the image:

$$T = \begin{bmatrix} 1 & 0 & -\frac{width}{2} \\ 0 & 1 & -\frac{height}{2} \\ 0 & 0 & 1 \end{bmatrix}$$

Now, lets rotate the image such that the epipole will come to some point $(f, 0, 1)$ on the horizontal axis. If the translated epipole Te' is located at homogeneous coordinates $(e'_1, e'_2, 1)$, then the rotation matrix is given by the following:

$$R = \begin{bmatrix} \alpha\frac{e'_1}{\sqrt{e'^2_1+e'^2_2}} & \alpha\frac{e'_2}{\sqrt{e'^2_1+e'^2_2}} & 0 \\ -\alpha\frac{e'_2}{\sqrt{e'^2_1+e'^2_2}} & \alpha\frac{e'_1}{\sqrt{e'^2_1+e'^2_2}} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

where,

$$\alpha = \begin{cases} +1, & \text{if } e'_1 \geqslant 0 \\ -1, & \text{if } e'_1 \leqslant 0 \end{cases}$$

Now by performing the following transformation, we move the point from $(f, 0, 1)$ to infinity on the horizontal axis $(f, 0, 0)$:

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\frac{1}{f} & 0 & 1 \end{bmatrix}$$

After the final transformation made to the epipole, we place the epipole at infinity. Now we can apply these transformations to the image to rectify it:

$$H_2 = T^{-1}GRT$$

Now we find the homography H_1 by minimizing the sum of square distances between the corresponding points of the images:

$$arg\min_{H_1} \sum_i ||H_1 p_i - H_2 p'_i||^2$$

The above equation can be simplified and can be derived to the following equation:

$$H_1 = H_A H_2 M$$

where,
$M = [e]_\times F + ev^T$

$v_T = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$

$H_A = \begin{bmatrix} a_1 & a_2 & a_3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$

We need to find the values of a_1, a_2, a_3. To find H_1, we need to minimize the sum of square distances between the corresponding points of the images. We can find the values of H_2 and M from the above derived equations, we can use the following and substitute these in the minimization problem:

$$\hat{p}_i = H_2 M p_i$$

$$\hat{p}'_i = H_2 p'_i$$

$$\implies arg\min_{H_A} \sum_i ||H_A \hat{p}_i - \hat{p}'_i||^2$$

let,
$\hat{p}_i = (\hat{x}_i, \hat{y}_i, 1)$
$\hat{p}'_i = (\hat{x}'_i, \hat{y}'_i, 1)$

$$\implies arg\min_{H_A} \sum_i (a_1\hat{x}_i + a_2\hat{y}_i + a_3 - \hat{x}'_i)^2 + (\hat{y}_i - \hat{y}'_i)^2$$

also, we know that $(\hat{y}_i - \hat{y}'_i)$ is a constant value, hence we can further simplify the above equation as follows:

$$\implies arg\min_{H_A} \sum_i (a_1\hat{x}_i + a_2\hat{y}_i + a_3 - \hat{x}'_i)^2$$

The above equation, makes the whole process about finding H_A into finding the $\boldsymbol{a}$ value by solving a least-squares problem $W\boldsymbol{a} = b$.
where,

$$W = \begin{bmatrix} \hat{x}_1 & \hat{y}_1 & 1 \\ \vdots & \vdots & \vdots \\ \hat{x}_n & \hat{y}_n & 1 \end{bmatrix} \qquad b = \begin{bmatrix} \hat{x}'_1 \\ \vdots \\ \hat{x}'_n \end{bmatrix}$$

After finding the values of a, H_A and M we can finally calculate H_1. With both the homographies H_1 and H_2, we can rectify all the pair of stereo-images for better correspondence matching.

The following is an example of image rectification performed on a pair of stereo images captured by two cameras. Zhang's Calibration method is used to perform camera calibration and then image rectification is performed. A checker board is used with the size of the square being 8.8cm x 8.8cm. The corner points of the squares are detected. There points are used to find the parameters of the cameras using the eight point algorithm. Also they are used to find the epipolar lines and the required homographies to perform image rectification on the obtained images.

Figure 3.15: Stereo Pair of Images

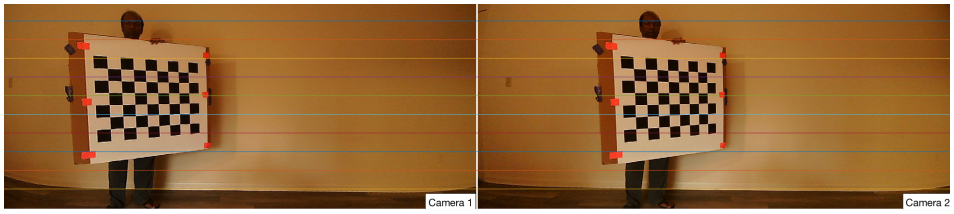


Figure 3.16: Rectified Images of the Stereo Pair of Images with the Epipolar Lines

3.9 Texture Analysis

Quantifying the texture of an object or a segment of an image in the field of image processing is called as texture analysis. Texture analysis is applied in various areas of computer vision. Segmentation of images, classification of images, etc. are a few examples in which texture analysis is used. Understanding, modeling and processing texture are few of the major goals of texture research which is conducted in computer vision. Image textures are naturally found in scenes captured in images or can also be artificially generated in an image. The following are a few features which make textures unique:

1. Size or granularity
2. Directionality
3. Randomness

4. Texture elements (texels) and their spatial arrangement

Every natural and man-made object in the world has its own unique texture. Texture analysis is mainly used in the process of segmentation of images into different interest regions so that the images can be classified according to different regions. Texture tells us about many different properties of the image like the spacial arrangement of colors and intensities in an image.

There are two main approaches to perform texture analysis:

1. Structural Approach: Texture analysis is seen as the analysis of a regular or repeated pattern of a set of primitive texels. The fundamental unit/element of a kind of texture in an image is called as a texel, texture pixel or texture element.

2. Statistical Approach: Obtaining the quantitative measure of the spacial arrangement of intensities and color in a region is called as the statistical approach.

Chapter 4

Depth Estimation Using Texture

The proposed solution for calculating disparity maps for stereo cameras with large baseline is as follows:

4.1 Disparity Calculation

Our main challenge for calculating disparity maps for stereo cameras with large baseline is to accommodate the huge change in perspective. Hence, I use the method of Multi-Block Matching approach proposed by Nils Einecke and Julian Eggert. This algorithm produces rough depth estimates and uses low computing resources. This kind of algorithm suits applications which uses multiple techniques for depth estimation like robotic applications and autonomous vehicles etc. It is a semi-global-matching algorithm which is a new novel approach to multi-block matching algorithms; compared to the standard block-matching stereo, this can be implemented using low memory-footprint and low computational complexity.

4.2 Multi-Black Matching

Standard block-matching stereo is basically correlating image patches, i.e. blocks or filters, between the left and right stereo images, resulting in correspondences between the images being found. From the positions of the correspondences between the two images, we find the disparity image which in turn results in the calculation of the depth. For block matching, there are lot of cost functions which are used: sum of absolute difference (SAD), normalized cross-correlation (NCC), rank transform (RT) or census transform, etc. The block matching cost function can be described as follows:

$$\hat{C}(p,d) = \sum_{q \in N_p} C(q,d)$$

where,
$\hat{C}$ = Cost Function
N_p = Number of pixels around pixel p
C = per pixel cost function

$$C(p,d) = \theta(I_1(p), I_2(p+d))$$

where,
I_1 = First Image
p = pixel position in the first image I_1
d = disparity
I_2 = Second Image
θ = Actual Pixel Cost Metric

A problem called fattening is introduced when the averaging of C is performed. This smears the disparity values in the disparity map. Foreground-fattening also known as bleeding is when foreground disparities are smeared over background disparities. A trade-off between the size of the matching block is introduced due to fattening. Large blocks result in dense disparity map with strong fattening whereas small blocks result in noisy and sparse maps with very weak fattening. To tackle this problem lot of solutions have been tried and tested; for example instead of building a straight averaging sum, contribution of each pixel is weighted with its similarity to the center pixel.

As mentioned, the problem with stereo is the assumption that the pixel contains a part of only one object. Similarly block matching stereo has an assumption of homogeneity, i.e., that all pixels within one block have the same disparity. When it comes to outdoor real world scenes, this assumption is violated to a large extent. Image pre-warping is one of the solutions proposed to tackle this problem but is successful only in some particular cases and scenes.

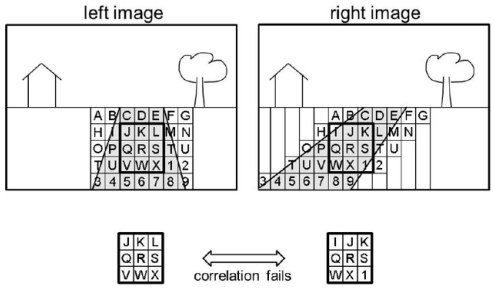


Figure 4.1: Multi-block Matching [22]

The focus of this kind of stereo is to tackle the violations due to the inherent assumption of homogeneity. The main idea is to combine matching blocks of different sizes and shapes in a probabilistic manner to get the best results of each matching block. As shown in the above image, having squared blocks does not help and leads to high violation. If we used only horizontally elongated matching blocks, we will have less problems since violation of the homogeneity does not occur but this results in high horizontal fattening.

For combining matching blocks, the costs of the typical block matching correlation search for a pixel p is considered as a probability distribution over the disparities d. This means that the costs of a set of different blocks B can be integrated in a multiplicative fashion:

$$\hat{C}(p,d) = \prod_{b \in B} \frac{\hat{C}_b(q,d)}{S_b(p)}$$

$$\hat{C}(p,d) = \sum_{q \in N_p^b} C(q,d)$$

$$S_b(p) = \sum_{d} \hat{C}_b(p,d)$$

where,
$\hat{C}_b$ = matching cost of block b
S_b = sum of all costs $\hat{C}_b(p,d)$ for a pixel over all disparities
S_b is also the normalization factor which turns the cost distributions into probability distributions.

The following shows a schematic visualization of the multi-block-matching (MBM) principle. We regard the values obtained during the correspondence search for each block as a probability distribution for each pixel. As you can see the figure below, the top two graphs show the probability distribution for the horizontal and vertical matching blocks and the third graph shows the combination of the two blocks.

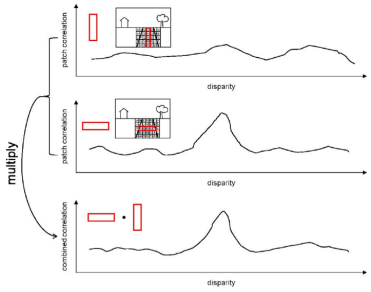


Figure 4.2: Disparity graph for different blocks [22]

It is not necessary to know $S_b(p)$. That is because we can calculate the final disparity for a pixel p in block matching stereo by selecting the disparity with the best cost:

$$D(p) = argmax\hat{C}(p,d)$$

This means for the final disparity for this algorithm is given by the following equation:

$$D(p) = argmax\tilde{C}(p,d)$$

$$D(p) = argmax \prod_{b \in B} \frac{\hat{c}(p,d)}{S_p(p)}$$

$$D(p) = argmax \prod_{b \in B} \hat{c}(p,d)$$

4.3 Cost Function

I calculate the cost function by using the HSV color space and the results obtained from Laws masks which performs texture analysis to the images of the stereo pair.

4.3.1 HSV Color Space

HSV is a color model which describes the colors in terms of their shade and their brightness value. HSV is an acronym for Hue, Saturation, Value. The HSV model is inspired from how the humans perceive color and its attributes. The saturation dimension depicts the various shades of bright colors and the dimension of value tells us how these mix.

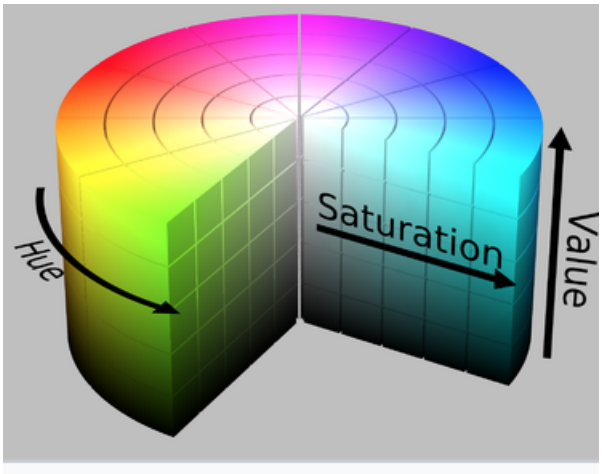


Figure 4.3: The cylinder model of HSV [22]

The hue has an angular dimension which starts at 0° at red, passes through green at 120°, blue at 240° and finally wraps back to red at 360°. The vertical axis represents the value dimension which starts at 0 at black and goes all the way to the top with white at 1.

The formal definitions are as follows:

1. Hue: The attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors: red, yellow, green, and blue, or to a combination of two of them.
2. Saturation: The colorfulness of a stimulus relative to its own brightness.
3. Value: The brightness relative to the brightness of a similarly illuminated white.

4.3.2 Laws Masks

Laws masks were developed by K.I. Laws in 1980 as a texture energy measure by using local masks to detect various types of texture. This method is extensively used in many diverse applications. This is one of the statistical approaches for doing texture analysis which gives a quantitative measure of the texture features. This approach basically lets us find the amount of variation in terms of texture features within a fixed-size window.

The energy measure is obtained by performing convolutions with the help of small kernels (windows/masks) to a digital image. This is a signal-processing based algorithm which uses filters which are applied to an image. These filtered images are where we extract the textured features. There are 8 1-D kernels from which the 2-D kernels are derived. The following are the 1-D kernels of length three and five:

$$L_3 = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

$$E_3 = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

$$S_3 = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$

$$L_5 = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

$$E_5 = \begin{bmatrix} -1 & -2 & 0 & 2 & 1 \end{bmatrix}$$

$$S_5 = \begin{bmatrix} -1 & 0 & 2 & 0 & -1 \end{bmatrix}$$

$$W_5 = \begin{bmatrix} -1 & 2 & 0 & -2 & 1 \end{bmatrix}$$

$$R_5 = \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \end{bmatrix}$$

Each kernel is used to extract a particular feature out of the image. Each letter in the name of the filter stand for the following:

- L - Level - Average gray level
- E - Edge - Extract edge features
- S - Spot - Extract spots
- W - Wave - Extract wave features
- R - Ripple - Extract ripples

By using these 1-D kernels, we can form 2-D kernels which can include the directionality component into our texture analysis. The following are a list of unique 2-D texture masks formed from the 1-D kernels which give unique description features of a texture and are used for the calculation of energy statistics:

Kernel Name	Kernel Value	Description of the extracted texture features
L_3L_3	$L_3^TL_3$	grey level intensity in horizontal and in vertical direction
L_3E_3	$L_3^TE_3$	edge detection in horizontal direction and grey level intensity in vertical direction
L_3S_3	$L_3^TS_3$	spot detection in horizontal direction and grey level intensity in vertical direction
E_3L_3	$E_3^TL_3$	grey level intensity in horizontal direction and edge detection in vertical direction
E_3E_3	$E_3^TE_3$	edge detection in horizontal and vertical direction
E_3S_3	$E_3^TS_3$	spot detection in horizontal direction and edge detection in vertical direction
S_3L_3	$S_3^TL_3$	grey level intensity in horizontal direction and spot detection in vertical direction
S_3E_3	$S_3^TE_3$	edge detection in horizontal direction and spot detection in vertical direction
S_3S_3	$S_3^TS_3$	spot detection in horizontal and in vertical direction

Table 4.1: 2-D Kernels of Laws Masks

Similarly we can use L_5, E_5, S_5, W_5, R_5 to form 2-D kernels for feature extraction. The following 9 features formed from the 1-D kernels are defined as follows:

- L_5E_5/E_5L_5

- L_5R_5/R_5L_5
- E_5S_5/S_5E_5
- L_5S_5/S_5L_5
- E_5R_5/R_5E_5
- S_5R_5/R_5S_5
- S_5S_5
- R_5R_5
- E_5E_5

4.3.3 Cost Map

A vector of length 12 for each pixel from both images in the stereo pair is formed. This vector consists of the average values from the HSV color space and the 9 features from texture analysis from a block of pixels around the main pixel. The two vectors from both the images are then compared using sum of squared differences (Euclidean Distance) and normalized cross-correlation.

Sum of Squared Differences: The cost for each pixel is calculated by subtracting the vectors of pixels being compared from both the images (left and right) and squared. The following is the equation used to find the SSD for a pixel:

$$C(x, y, d) = \sum (I_R - I_L)^2$$

Normalized Corss-Correlation: The cost for each pixel is calculated by find the normalized cross-correlation coefficient of the vector of the pixel for which the cost is being assigned. The following is the equation used:

$$C(x, y, d) = \frac{\sum (I_L I_R - \mu_L \mu_R)}{\sigma_L \sigma_R}$$

By using the above cost functions, I assign the costs for each pixel for windows in different disparity range and then select the most suitable cost from the outputs of different windows and form the final cost matrix. I use this cost matrix for finding the disparity map.

Chapter 5

Stereo Vision Algorithms

5.1 Introduction

One of the highly researched upon areas in the field of computer vision is Stereo Matching. A huge number of algorithms have been developed to solve this problem. But when it comes to classifying and ranking these algorithms, very little work has been done. Researchers usually find it very difficult to view the progress in this area due to little work being done when it comes to classification and ranking these algorithms. Mostly researchers publish their algorithms and their quantitative results and do not perform a wide comparative analysis of their algorithm with the state-of-the-art. Also these algorithms are made for different types of applications which make them effective only under certain conditions. There is a great need to perform surveys and analysis of their stereo matching algorithms. [23,24,25] are the only exhaustive surveys performed in this field. Szeliski and Scharstein performed a survey and provided a taxonomy of existing stereo algorithms[25]. Their main focus was on calibrated two-frame methods but their analysis can be expanded and generalized to other algorithms also.

Stereo correspondence algorithms, basically look for evidence that points in the two images match, i.e., those two points correspond to the same point in the real world and are simply different projections having different perspectives in the two images. Different algorithms look for these evidences based on different assumptions and methodologies. While some algorithms depend on the assumption that the surfaces do not change appearance with changing viewpoint while other algorithms model themselves in a different manner.

Designing and developing an algorithm for stereo matching is easier to do for indoor situations as compared to outdoor situations where we have no control of environmental factors like light, wind, etc. Apart from the difficulties which arise due to the light conditions, there is one more important factor which has to be given significance, i.e. the size of the pixel and the contents inside the pixel. When an image is read by a computer, the image

is represented by the values of intensity for each pixel. This means that the value of each pixel seen by the computer depends on the light falling on that pixel. This light is usually reflected onto the image sensor from objects in front of the camera which are captured in the image. As we know the size of the pixel keeps increasing as it moves from the camera, i.e., a pixel which is covering objects far from the camera cover more real world space than the objects near the camera. So then how does the pixel value for objects far from the camera be represented as? It's an average value of the intensities reflected from the objects, i.e. the average value is viewed in the image read by the computer. Hence the depth of an individual object itself is compromised for pixels containing more than one object.

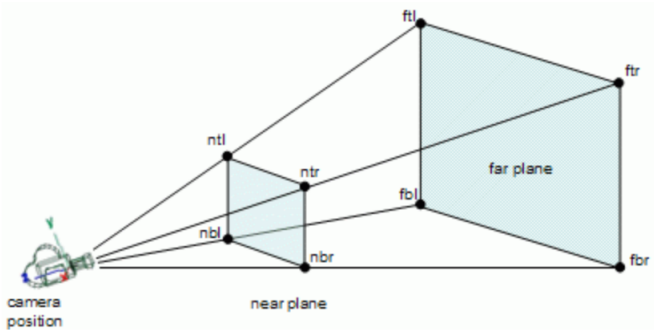


Figure 5.1: View Frustum from a Camera [28]

The most common assumption taken into consideration when developing a stereo algorithm is that the image is made of all Lambertian surfaces. A surface whose appearance does not change with different viewpoints is called as a Lambertian surface. There are researchers who also model and develop algorithms with specific kinds of noise to the camera or give a gain or a bias so that this assumption is not violated. The other main assumptions made while developing most of the stereo matching algorithms are uniqueness and smoothness of the disparity map in the image plane. These two factors are also part of the main and the most difficult challenges faced by researchers developing algorithms for stereo matching. Transparent surfaces or occlusions are the most common things which violate uniqueness of disparity. Occlusion is the inability to see something in the image due to some property of the cameras or the way the cameras are setup. Occlusion is a very difficult phenomenon which has to be dealt while developing stereo algorithms. Also when it comes to disparity maps, smoothness is often violated around the boundaries of the different objects in the image.

Main methods of depth estimation using computer vision are local and global methods. Correspondence matching of patches between the two images of a stereo image pair is done in the local methods. Energy function optimization is used to get the best transformation of one image onto the other and this is performed in the global methods. Semi-global-matching is the method which exists between the two groups. SGM does not apply a fully global energy function but energy function optimization is done along at least one-dimensional paths.

The algorithms can also be mainly classified into two classes, algorithms where the camera geometry is known and algorithms where the camera geometry is not known. The algorithms where the camera geometry is known, these algorithms are made to compute disparity maps by performing correspondence matching in the two images of the stereo pair. The algorithms can also be classified by the way the disparity is represented. The disparities which are used in the computer vision society are univalued, multi-valued, layer-based, voxel-based, etc.

The main goal of stereo correspondence algorithms are to produce a disparity map which will help tell us the shapes of the objects in 3D or help us gauge the distance of the object from the camera in some known coordinate frames. According to the study done by Daniel Scharstein and Richard Szeliski [1], any kind of stereo algorithm performs a subset of the following steps[25]:

1. Matching Cost Computation
2. Cost Aggregation
3. Disparity Computation/Optimization
4. Disparity Refinement

As an example to the above step process, the most tradition stereo algorithm using the sum of squared differences can be described as follows:

1. The matching cost is the squared difference of intensity values at a given disparity.
2. Aggregation is done by summing matching cost over square windows with constant disparities.
3. Disparities are computed by selecting the minimal aggregated value at each pixel.

5.1.1 Matching Cost Computation

The pixel coordinates and the disparities obtained from the matching cost form the disparity image initially. Squared intensity differences or mean-squared error and absolute intensity differences or mean absolute difference are the two most used and common matching cost methods. There are many other recent techniques which are adopted to compute the cost of matching and get a reliable match like the truncated quadratics and contaminated gaussians which limit mismatch influence during aggregation.

There are matching cost algorithms based on binary features like edges. Normalized cross correlation, gradient-based measures are other techniques which are used to form these algorithms. Some algorithms use phase and filter-bank responses to designate the cost for the matches.

5.1.2 Aggregation Of Cost

There are two main methods used to aggregate the matching cost – Local and window based methods. Aggregation is done by averaging or summing over a region. The size of the region being considered depends on the method being used. Two types of regions have been implemented across the different algorithms – 2D and 3D (x, y, d). The 2D algorithms are based on windows – local or adaptive sizes. 2D or 3D gaussian convolution is also used as an aggregation process:

$$C(x, y, d) = w(x, y, d) * C_0(x, y, d)$$

Sometimes even box filters are also used to implement this process.

5.1.3 Disparity Computation And Optimization

The final disparity computation is a trivial issue as compared to the above two steps. Here disparity for each pixel is selected in such a way that the cost for the selected disparity values is the minimum. This is a brute force - "winner-take-all" (WTA) optimization for every pixel in the image.

The most traditional method used for computation of disparity which was inspired by computational models of human stereo vision are the cooperative algorithms. They are similar to algorithms called global algorithms. They perform computations locally with a usage of non-linear operations which give the overall behavior of disparity.

There is one more methods where the minimization is computed for the whole image using disparities and uses an energy minimization framework to get a final disparity map. In this method, the main goal is to find a disparity function which minimizes the energy globally.

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

$E_{data}(d)$ is the measure of the agreement between the disparity function and the input image pair. The smoothness function is different for each algorithm and takes into account most of the assumptions on which the algorithms are based on. In recent algorithms, max-flow and graph-cut methods have been proposed to solve a special class of global optimization.

5.1.4 Refinement Of Disparities

This is an additional step and is not required for most of the applications of stereo vision. But to improve the results of your algorithm this step is performed with a little additional computation. The resolution of stereo algorithm is increased by calculating sub-pixel disparity estimates. These calculations are done through iterative gradient descent and fitting a curve to the matching costs at discrete disparity levels.

Apart from the above mentioned sub-pixel computations, there are of course other ways of post processing the computed disparities. Cross-checking can be performed to detect occlusions. Spurious mismatches can also be removed by applying a median filter.

5.1.5 Other Methods

The above broad classification of the process for computing disparities can only account for some of the stereo vision algorithms. There are a lot of algorithms which do not fall under this classification. Most of the algorithms which use techniques of optical flow, machine learning, deep neural networks do not follow the above approach. For example: there is continuous motion estimation which is continuously updated in algorithms using optical flow techniques, there is a training phase for all algorithms using machine learning and deep neural network techniques.

Also when it comes to representing the disparity map, the above process will only produce univalued representation of disparity maps. There are other ways to represent a disparity map like the voxel-based representation which is a multi-valued representation. These type of multi-valued representations represent several depth values along each line of sight. These algorithms which produce multi-valued representations of disparity maps do not follow the above process.

5.2 Evaluation Of Stereo Vision Algorithms

Once we find the disparity map, we need to establish a credible way to verify the response obtained from the algorithm. We need a quantitative way to estimate the quality of the algorithm and find the effects of the algorithm by varying its parameters. There have been a lot of approaches to obtain quality metrics for evaluating the performance of the algorithms which perform correspondence matching with the help of ground truth estimates.

There are two common methods to accomplish this goal. Firstly, we can find the error of the disparity maps which have been computed using some ground truth data. Secondly, we can also form synthetic images which we can obtain by warping the images by the disparity map computed using the algorithms. The following are ways described by [25] to find the error and the percentage of bad matching pixels from the disparity maps obtained by the algorithms:

1. RMS (root-mean-squared) error (measured in disparity units) between the ground truth map $d_T(x, y)$ and the obtained disparity map $d_C(x, y)$,

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

where N is the total number of pixels.

2. Percentage of bad matching pixels,

$$B = \frac{1}{n} \sum_{(x,y)} (|d_C(x,y) - d_T(x,y)| > \delta_d)$$

where $\delta_d(eval\_bad\_thresh)$ is a disparity error tolerance.
I use $\delta_d = 1.0$ as [25] uses it and also some previously published studies.

5.3 Ranking Of Stereo Vision Algorithms

Lot of researchers think that, if each algorithm/method for finding disparities is treated as a "black box" then we can gain only limited value about it. It has been frequently stated in this field that more insight can be gained if the workings of different components are studied and understood. It is true that we can gain a better understanding of why an algorithm is better compared to the other algorithms if we know the complete workings of that algorithm, but there is also a need to establish a ranking scale to algorithms based on the factors used when treating the algorithm like a black box. A researcher develops an algorithm in stereo matching to obtain disparity maps keeping in mind the two main factors: Accuracy and time.

We also have a high need to rank by first performing classification on the basis of the need to pre-process. In this case, I also include training of algorithms in pre-processing. There has been a lot of research in this field in the past decade and researchers have been using various techniques to obtain good results. Algorithms developed these days use techniques from machine learning and deep learning which have a need to train the algorithm before it can be used in the field.

One of the main goals while developing a stereo matching algorithm is to obtain a disparity map with least possible error or zero error under all scenarios. The other main goal is the run time of these algorithms which is mostly based on the computational power required for these algorithms to run. These two main factors dictate the rank or standing of the algorithm to perform a classification for out of the box usage. Getting a classification done based on these two factors will help researchers who are performing research in different fields to pick and use the algorithm which best suits their application. We have to establish a ranking system by evaluating jointly the two parameters of comparison.

To effectively rank these algorithms we have to first provide a system which will score these algorithms based on the two main parameters: accuracy and time. Scoring system should also be robust and flexible to add any parameters which are considered important for the researcher.

5.3.1 Multiple Criteria Decision Making

The field which deals with the theory, methodology, professional practice, and the philosophy of decision making process in a formal manner is called as Decision Analysis. There have been many methods, procedures, tools for clearly establishing and understanding every aspect of a decision making process. Multiple-criteria decision making (MCDM) is a study or field which deals with decision making by evaluating multiple criteria. There is a great importance to properly structure the problem and evaluate multiple criteria explicitly. To make complex decisions, MCDM is one of the very valuable tools present.

To get more informed and good decisions, you should structure any kind of complex problem well and consider multiple criteria influencing the problem. There are many decision making softwares which consist of many different approaches and methods which have been developed for various applications ranging from energy and sciences to business, management and even politics.

Multiple criteria decision making deals mainly with problems involving solving for decisions with multiple criteria. Usually there is no unique perfect solution for such problems and the optimal solution for a researcher/decision-maker completely depends on his preferences and the application. Searching for the best can always be seen as "the most preferred" for the decision-maker. There are also many interpretations for solving/searching like choosing the best among a set of alternatives, or grouping all things available into sets according to ones preferences or it can also be interpreted as finding the most efficient for a particular application.

MCDM problems are made of five components:

1. Goal
2. Decision maker
3. Decision alternatives
4. Evaluation criteria
5. Outcomes or consequences associated with alternative/interest combination

The main difficulty of the problem comes from having more than one criterion to measure. For any MCDM problem, there never exists a unique solution. One always has to have a set of preferences which have to be incorporated into the solution to make a decision. Solution made under these conditions are often called as a nondominated solution. A nondominated solution has a very important property where moving away from the solution causes to sacrifice at least one criterion. Hence MCDM solutions usually are a set of nondominated solutions. These set of nondominated solutions are usually large, hence many

various tools are used to focus on one solution which gives the best results for the researcher according to his application.

MCDM problems are mainly classified into two categories: Evaluation and Design problems. Evaluation problems usually have solutions which are explicitly defined. Decision makers who usually try to make decisions for the purpose of sorting or classification fall under this category. Design problems do not usually have solutions explicitly known. Such problems have an infinite set of nondominated solutions for decision makers. For both these categories, we keep in mind the preferences of the decision maker.

There are many methods in solving problems of decision making with multiple criteria. Some of the methods require the preferences of the decision maker at the start of the process where as some methods take the preferences to build a value function to solve the problem of decision making. There are also methods where there is requirement of the preference information at every step of the solution.

5.4 Scoring Stereovision Algorithms

A scoring function is proposed which assigns scores for stereo algorithms based on the criteria and weights desired by the researcher.

A posynomial is a function of the following form,

$$f(x_1, x_2, ..., x_n) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} .....x_n^{a_{nk}}$$

Each term in the posynomial is attributed to one major criteria. Each element in the term is attributed to a minor criteria. The power of the element gives us the weight of each minor criteria. Each term is also multiplied by a weight for each major criteria.

I have considered Error and Time of execution as my two main criteria to score and rank the stereo algorithms. I sub-divided my error criteria into RMS error and percentage of bad matching pixels. I have weighted all my criteria equally. Hence my scoring function using the above defined posynomial will become the following:

$$f(R, B, T) = \left(\frac{1}{2} * R^{\frac{1}{2}} * B^{\frac{1}{2}}\right) + \left(\frac{1}{2} * T\right)$$

For the implementations here, we use the following scoring function:

bg: Percentage of outliers averaged only over background regions
fg: Percentage of outliers averaged only over foreground regions
all: Percentage of outliers averaged over all ground truth pixels
T : Time of computation

$$f(fg, bg, all, t) = \left(\frac{1}{2} * fg^{\frac{1}{2}} * bg^{\frac{1}{4}} * all^{\frac{1}{4}}\right) + \left(\frac{1}{2} * T\right)$$

Chapter 6

Results and Conclusions

6.1 Disparity Calculation

As stated above, the disparity map is calculated by using the cost function generated with the help of the HSV color scheme and laws textures.

The following steps have been performed to achieve the desired results:

1) Original stereo pair of images:

(a) Left Image (b) Right Image

Figure 6.1: Stereo Pair of Images

2) Ground Truth Image:

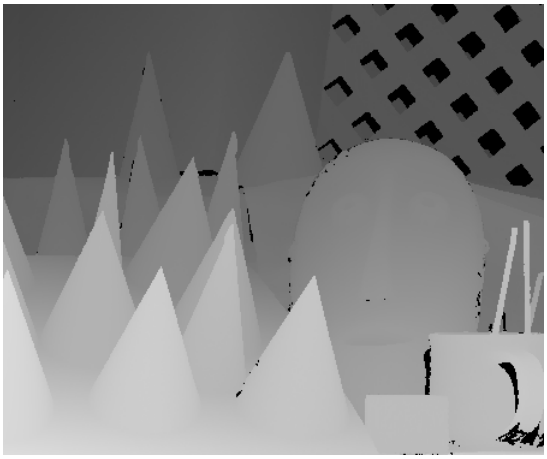


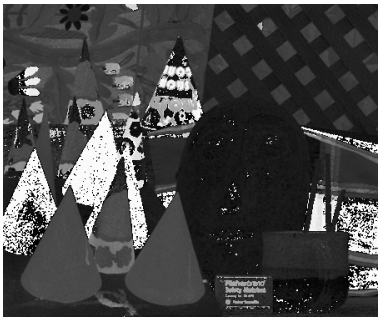
Figure 6.2: Ground Truth Image

3) Images depicting the HSV color scheme:

(a) Left Image

(b) Right Image

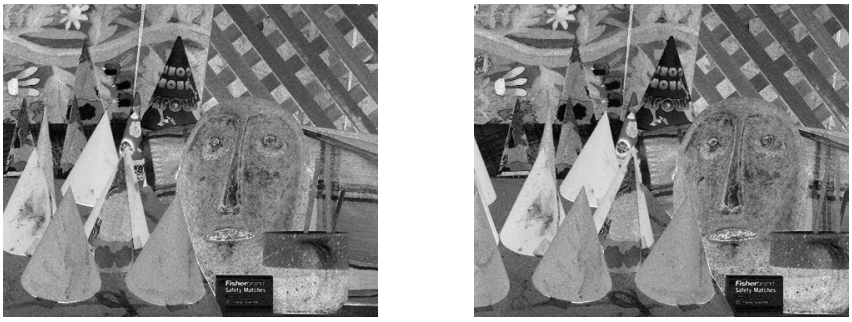
Figure 6.3: HSV color scheme of the Stereo Pair of Images



(a) Left Image

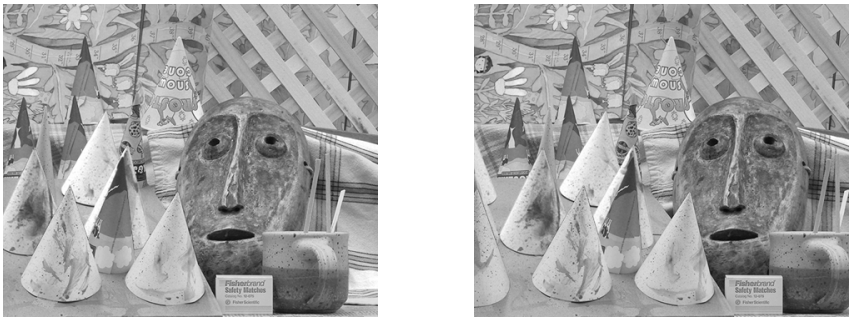
(b) Right Image

Figure 6.4: Hue channel of the Stereo Pair of Images



(a) Left Image (b) Right Image

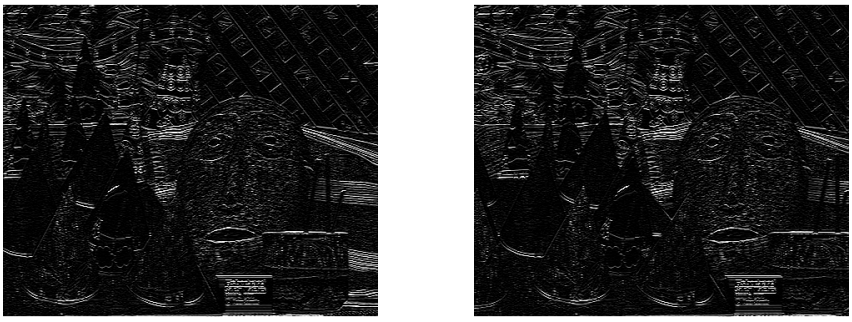
Figure 6.5: Saturation channel of the Stereo Pair of Images



(a) Left Image (b) Right Image

Figure 6.6: Intensity (Value) channel of the Stereo Pair of Images

4) Images showing the features of Laws textures for the images:

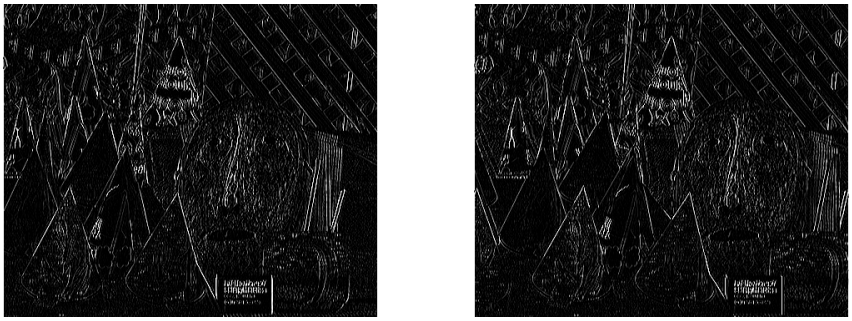


(a) Left Image (b) Right Image

Figure 6.7: Grey Level Intensity in the vertical and horizontal direction

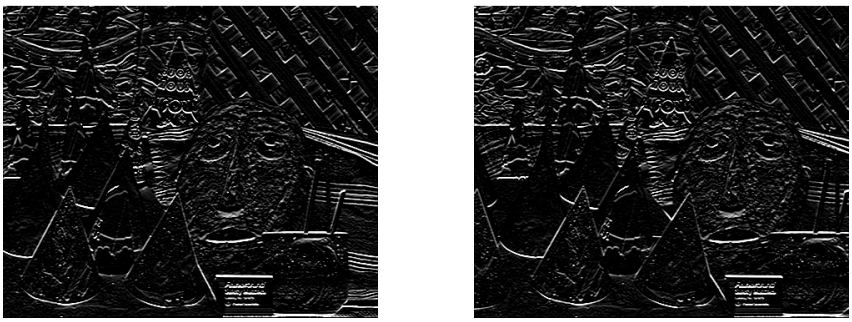
(a) Left Image (b) Right Image

Figure 6.8: Edge detection in the horizontal direction and grey level intensity in the vertical direction



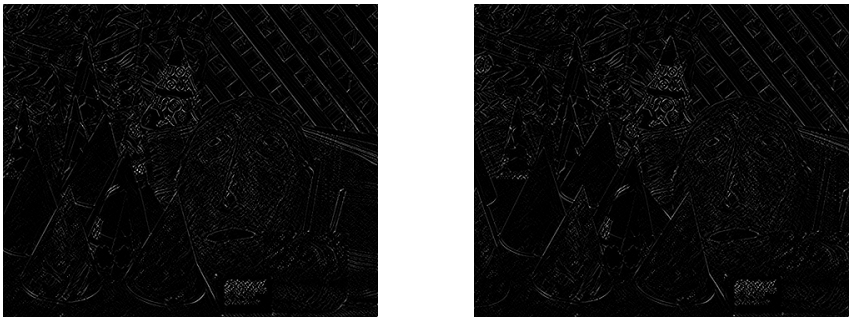
(a) Left Image (b) Right Image

Figure 6.9: Spot detection in the horizontal direction and grey level intensity in the vertical direction



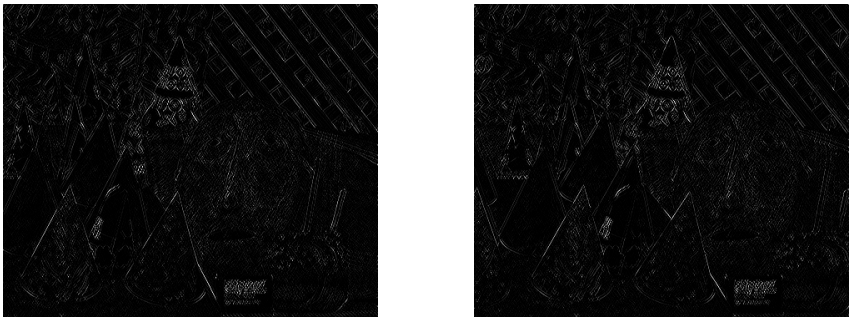
(a) Left Image (b) Right Image

Figure 6.10: Grey level intensity in the horizontal direction and edge detection in the vertical direction



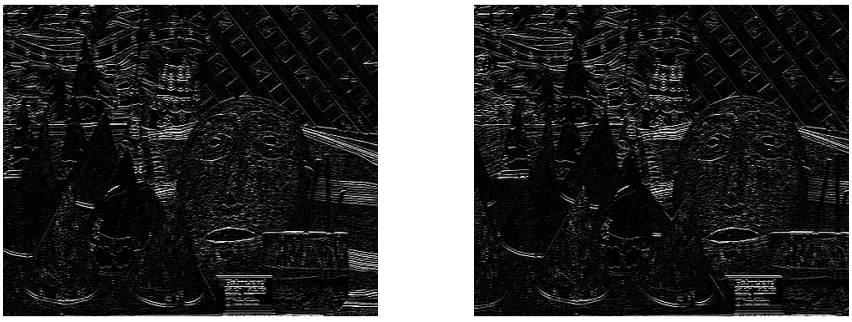
(a) Left Image (b) Right Image

Figure 6.11: Edge detection in the horizontal and vertical direction



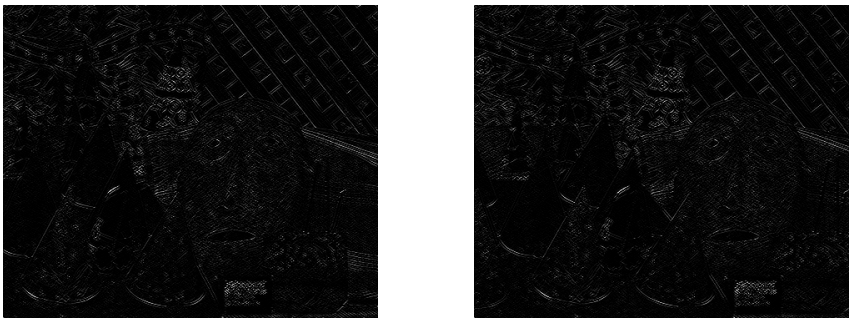
(a) Left Image (b) Right Image

Figure 6.12: Spot detection in the horizontal direction and edge detection in the vertical direction



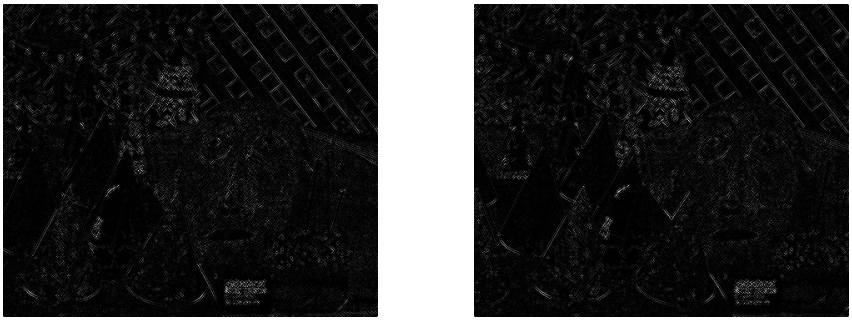
(a) Left Image (b) Right Image

Figure 6.13: Grey level intensity in the horizontal direction and spot detection in the vertical direction



(a) Left Image (b) Right Image

Figure 6.14: Edge detection in the horizontal direction and spot detection in the vertical direction



(a) Left Image (b) Right Image

Figure 6.15: Spot detection in the horizontal and in the vertical direction

5) Using Multi-block Matching to find the disparity map:

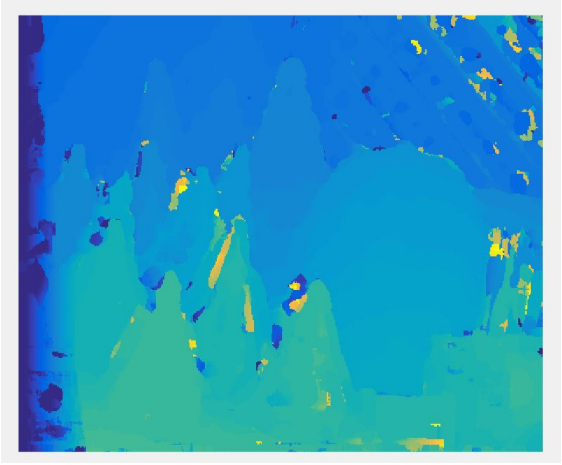


Figure 6.16: Disparity Map using multi-block matching

The lighter the blue shade means the closer the object in that pixel is to the camera. Comparing to the ground truth I get an error of 14.3%.

6.1.1 More Results

The following are the disparity maps calculated for different stereo pairs using the above method:

(a) Left Image (b) Right Image

Figure 6.17: Stereo Pair Of Images

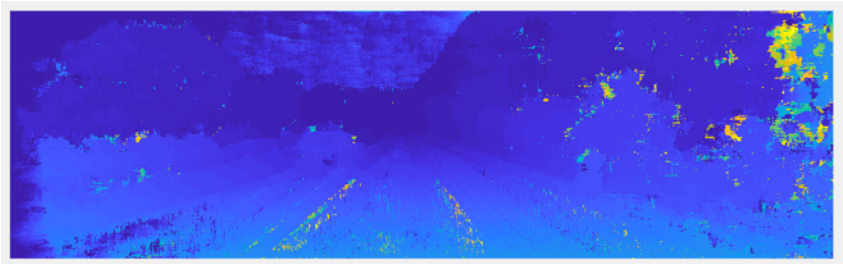


Figure 6.18: Disparity Map

(a) Left Image (b) Right Image

Figure 6.19: Stereo Pair Of Images

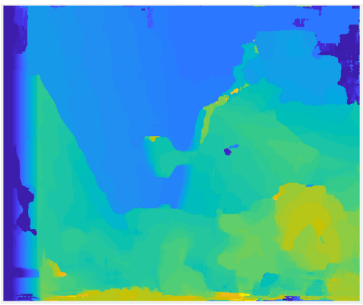
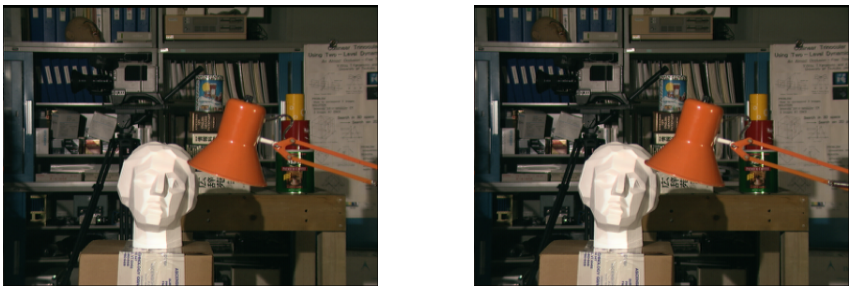


Figure 6.20: Disparity Map

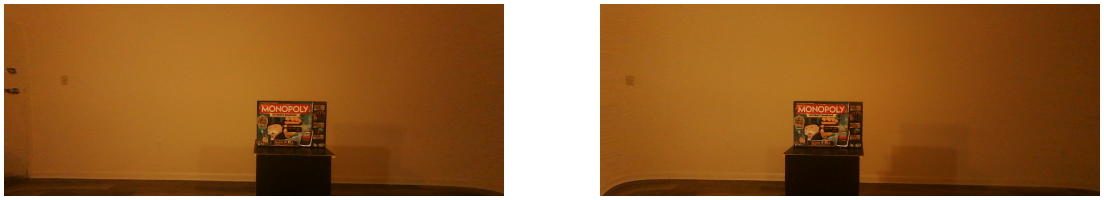


(a) Left Image (b) Right Image

Figure 6.21: Stereo Pair Of Images

Figure 6.22: Disparity Map

The following pair of Images have been taken by a camera from the lab and the disparity Map was calculated:



(a) Left Image (b) Right Image

Figure 6.23: Stereo Pair Of Images

Figure 6.24: Stereo Anaglyph of the Stereo Pair

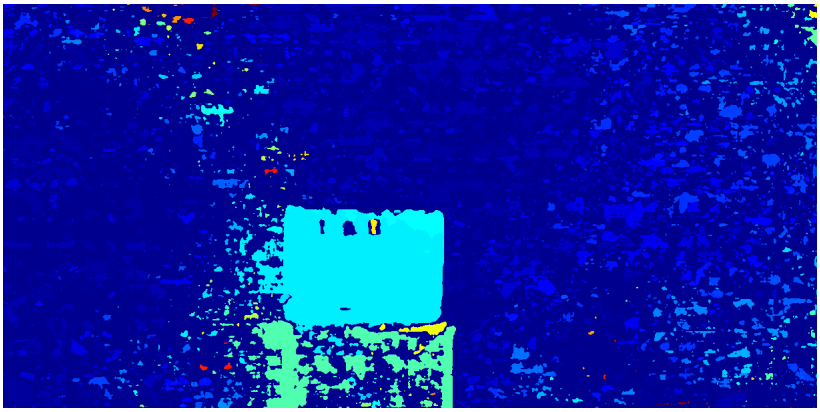


Figure 6.25: Disparity Map

The box placed on the table had an average disparity of 202 pixels. From this we can find the distance of the box from the camera frame. The distance was calculated to be 124.424 cm. The distance between the box and the camera frame was measured to be approximately 128 cm by tape.

6.2 Scoring Of Stereovision Algorithms

As mentioned in the previous chapter, we use the proposed scoring function to score five state-of-the-art algorithms and the implementation performed here using Texture analysis and Multi-block matching (TMBM):

Stereo Algorithm	fg (%)	bg (%)	all (%)	Run-time (sec.)	Score
Displets [56]	3.00	5.56	3.43	423	213.31
SPS [57]	3.84	12.67	5.31	172	88.81
CPM [58]	4.13	12.03	5.44	307	156.39
SED [60]	25.01	40.43	27.58	394	211.45
MBM [59]	4.69	13.05	6.08	286	146.23
TMBM	13.21	23.11	17.82	363	189.69

Table 6.1: Scoring of five state-of-the-art algorithms

All the implementations have been performed in MATLAB using a MAC with a 2.5 GHz Intel Core i7 Processor and a 16 GB 1600 MHz DDR3 RAM. All the codes have been wrapped using MEX wrappers which enables the implementation be performed on MATLAB.

6.3 Future Work

In the work shown, disparity was successfully calculated from the stereo pair of images using the cost function which was developed with the help of Laws masks. Machine learning and neural networks can be applied to these results to improve them in the future.

A scoring function was also successfully established and ranking was performed for five of the state-of-the-art algorithms. More algorithms for different applications and with different parameters have to be evaluated using the scoring function to validate it.

Bibliography

[1] G. Bradski and A. Kaehler, "Learning OpenCV: Computer Vision with the OpenCV Library," 2008.

[2] J. Y. Bouguet, "Camera Calibration Toolbox for Matlab."

[3] J. Heikkila and O. Silven, "A Four-step Camera Calibration Procedure with Implicit Image Correction," in *IEEE International Conference on Computer Vision and Pattern Recognition*, 1997.

[4] Z. Zhang, "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22(11), pp. 1330–1334, 2000.

[5] L. D. S. Federico Tombari, Stefano Mattoccia and E. Addimanda, "Classification and evaluation of cost aggregation methods for stereo correspondence."

[6] X. Lin, Y. Liu, and W. Dai, "Study of occlusions problem in stereo vision," in *2008 7th World Congress on Intelligent Control and Automation*, Jun. 2008, pp. 5062–5067.

[7] C. Chang, S. Chatterjee, and P. R. Kube, "On an analysis of static occlusion in stereo vision," in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Jun. 1991, pp. 722–723.

[8] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," in *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, 2001, pp. 131–140.

[9] U. R. Dhond and J. K. Aggarwal, "Structure from stereo," *IEEE Trans. on Systems, Man, and Cybern.*, vol. 19(6), pp. 1489–1510, 1989.

[10] S. T. Barnard and M. A. Fischler, "Computational stereo," in *ACM Comp. Surveys*, vol. 14(4), 1982, pp. 553–572.

[11] K. Hata and S. Savarese, *Epipolar Geometry*.

[12] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, Mar. 2004.

[13] W. Burger, "Zhang's Camera Calibration Algorithm: In-Depth Tutorial and Implementation," University of Applied Sciences Upper Austria, Tech. Rep., May 2016.

[14] K. Hata and S. Savarese, *Camera Models*.

[15] R. E. Turner, *Stereo Vision*, Nov. 2013.

[16] J. C. A. Read, *Stereo vision and strabismus.* Cambridge Ophthalmological Symposium, 2014.

[17] R. Szeliski, "Prediction error as a quality metric for motion and stereo," *ICCV*, pp. 781–788, 1999.

[18] V. I. J. Mulligan and K. Daniilidis, "Performance evaluation of stereo for tele-presence," *ICCV*, vol. 2, pp. 558–565, 2001.

[19] D. M. Y. C. Hsieh and F. P. Perlant, "Performance evaluation of scene registration and stereo matching for car- tographic feature extraction," *IEEE TPAMI*, vol. 14(2), pp. 214–238, 1992.

[20] H. H. B. R. C. Bolles and M. J. Hannah, "The JISCT stereo evaluation," *DARPA Image Understanding Work- shop*, pp. 263–274, 1993.

[21] A. Mitiche and P. Bouthemy, "Computation and analysis of image motion: A synopsis of current problems and methods," *IJCV*, vol. 19, pp. 29–55, 1996.

[22] M. Otte and H.-H. Nagel, "Optical flow estimation: advances and comparisons," *ECCV*, vol. 1, pp. 51–60, 1994.

[23] D. J. F. J. L. Barron and S. S. Beauchemin, "Performance of optical flow techniques," *IJCV*, vol. 12, pp. 43–77, 1994.

[24] T. P. Gian F. Poggio, "The Analysis Of Stereopsis," *Annual Review Neuroscience*, vol. 7, pp. 379–412, 1984.

[25] R. Szeliski, *Computer Vision: Algorithms and Applications.* Springer, 2010.

[26] N. Mansurov, "What is Distortion?" [Online]. Available: https://photographylife.com/what-is-distortion

[27] "Design and Build Your Own Pinhole Camera." [Online]. Available: http://www.instructables.com/id/Design-and-Build-your-own-Pinhole-Camera/

[28] "The History of the Camera," Apr. 2016. [Online]. Available: http://historythings.com/the-history-of-the-camera/

[29] "What Is Camera Calibration? - MATLAB & Simulink." [Online]. Available: https://www.mathworks.com/help/vision/ug/camera-calibration.html

[30] "Geometric Approach – Extracting the Planes," Apr. 2011. [Online]. Available: http://www.lighthouse3d.com/tutorials/view-frustum-culling/geometric-approach-extracting-the-planes/

[31] "MCDA." [Online]. Available: https://projects.ncsu.edu/nrli/decision-making/MCDA.php

[32] A. Mardani, A. Jusoh, K. Nor, Z. Khalifah, N. Zakwan, and A. Valipour, "Multiple criteria decision-making techniques and their applications – a review of the literature from 2000 to 2014," *Economic Research*, vol. 28, pp. 516–571, Jul. 2015.

[33] "Texture Analysis and its Applications." [Online]. Available: https://www.cs.auckland.ac.nz/~georgy/research/texture/thesis-html/node7.html

[34] "HSL and HSV," Nov. 2017, page Version ID: 808145655. [Online]. Available: https://en.wikipedia.org/w/index.php?title=HSL_and_HSV&oldid=808145655

[35] "Color Appearance Models." [Online]. Available: http://rit-mcsl.org/fairchild//CAM.html

[36] S. T. Barnard and M. A. Fischler, "Computational Stereo," *ACM Comp. Surveys*, vol. 14, no. 4, pp. 553–572, 1982.

[37] J. Žbontar and Y. LeCun, "Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches," *Journal of Machine Learning Research*, vol. 17, pp. 1–32, Oct. 2015, arXiv: 1510.05970. [Online]. Available: http://arxiv.org/abs/1510.05970

[38] L. Brown, "A Survey of Image Registration Techniques," *Computing Surveys*, vol. 24, no. 4, pp. 325–376, 1992.

[39] M. Levine, D. O'Handley, and G. Yagi, "Computer Determination of Depth Maps," *Computer Graphics and Image Processing*, vol. 2, pp. 131–150, 1973.

[40] M. Okutomi and T. Kanade, "A Locally Adaptive Window for Signal Matching," *IJCV*, vol. 7, no. 2, pp. 143–162, 1992.

[41] R. Arnold, "Automated Stereo Perception," Stanford University, Artificial Intelligence Laboratory, Technical Report AIM-351, 1983.

[42] D. Geiger, B. Ladendorf, and A. Yuille, "Occlusions and Binocular Stereo," *IJCV*, vol. 14, no. 3, pp. 211–226, 1995.

[43] M. Brown, D. Burschka, and G. Hager, "Advances in Computational Stereo," *IEEE Trans. PAMI*, vol. 25, no. 8, pp. 993–1008, 2003.

[44] H. Hirschmuller and D. Scharstein, "Evaluation of Cost functions for Stereo Matching," vol. 1, 2007, pp. 1–8.

[45] M. Gong, R. Yang, W. Liang, and M. Gong, "A Performance Study on Different Cost Aggregation Approaches Used in Real-Time Stereo Matching," *IJCV*, vol. 75, no. 2, pp. 283–296, 2007.

[46] C. Loop and Z. Zhang, "Computing Rectifying Homographies for Stereo Vision," *CVPR*, vol. 1, pp. 125–131, 1999.

[47] Z. Zhang, "Determining the Epipolar Geometry and its Uncertainty," *IJCV*, vol. 27, no. 2, pp. 161–195, 1998.

[48] R. Hartley and A. Zisserman, "Multiple View Geometry." Cambridge University Press, 2000.

[49] O. Faugeras and Q. Luong, "The Geometry of Multiple Images," 2001.

[50] T. Kanade, "A Stereo Machine for Video-Rate Dense Depth Mapping and its new Applications," *CVPR*, pp. 196–202, 1996.

[51] M. J. Hannah, "Computer Matching of Areas in Stereo Images," PhD, Stanford University, 1974.

[52] P. Anandan, "A Computational Framework and an Algorithm for the measurement of Visual Motion," *IJCV*, vol. 2, no. 3, pp. 283–310, 1989.

[53] L. Matthies, R. Szeliski, and T. Kanade, "Kalman Filter Based Algorithms for Estimating Depth from Image Sequences," *IJCV*, vol. 3, pp. 209–236, 1989.

[54] E. Simoncelli, E. Adelson, and D. Heeger, "Probability Distributions of Optic Flow," *CVPR*, pp. 310–315, 1991.

[55] D. Scharstein, "Matching Images by comparing their Gradient Fields," *ICPR*, vol. 1, pp. 572–575, 1994.

[56] P. Seitz, "Using Local Orientation information as Image Primitive for Robust Object Recognition," *SPIE Visual Communications and Image Processing*, vol. 4, pp. 1630–1639, 1989.

[57] T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiment," *IEEE TPAMI*, vol. 16, no. 9, pp. 920–932, 1994.

[58] S. Kang, R. Szeliski, and J. Chai, "Handling Occlusions in Dense Multi-View Stereo," *CVPR*, 2001.

[59] O. Veksler, "Stereo Matching by Compact Windows via Minimum Ratio Cycle," *ICCV*, vol. 1, no. 540-547, 2001.

[60] D. Terzopoulos, "Regularization Of Inverse Visual Problems Involving Discontinuities," *IEEE TPAMI*, vol. 8, no. 4, pp. 413–424, 1986.

[61] A. Blake and A. Zisserman, "Visual Reconstruction," 1987.

[62] P. Chou and C. Brown, "The Theory and Practice of Bayesian Image Labeling," *IJCV*, vol. 4, no. 3, pp. 185–210, 1990.

[63] D. Geiger and F. Girosi, "Parallel and Deterministic Algorithms for MRF's: Surface Reconstruction," *IEEE TPAMI*, vol. 13, no. 5, pp. 401–412, 1991.

[64] H. Baker and T. Binford, "Depth from Edge and Intensity Based Stereo," *IJCAI*, vol. 81, pp. 631–636, 1981.

[65] Y. Ohta and T. Kanade, "Stereo by Intra- and Inter- Scanline using Dynamic Programming," *IEEE TPAMI*, vol. 7, no. 2, pp. 139–154, 1985.

[66] P. Dev, "Segmentation Processes In Visual Perception: A Cooperative Neural Model," Univerity Of Massachusetts at Amherst, Technical Report 74C-5, 1974.

[67] D. Marr and T. Poggio, "Cooperative Computation Of Stereo Disparity," *Science*, vol. 194, pp. 283–287, 1976.

[68] J. Marroquin, "Design of Cooperative Networks," MIT, Working Paper 253, 1983.

[69] R. Szeliski and G. Hinton, "Solving Randomdot Stereograms Using the Heat Equation," *CVPR*, pp. 284–288, 1985.

[70] H. Hirschmu and D. Scharstein, "Evaluation of Cost Functions for Stereo Matching," in *Computer Vision and Pattern Recognition*, 2007.

[71] L. Zhangand and S. Seitz, "Estimating Optimal Parameters for MRF Stereo from a Single Image Pair," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 331–342, 2007.

[72] D. Scharstein and C. Pal., "Learning Conditional Random Fields for Stereo," in *Computer Society Conference on Computer Vision and Pattern Recognition*, 2007.

[73] Y. Li. and D. Huttenlocher, "Learning for Stereo Vision Using the Structured Support Vector Machine," in *Computer Vision and Pattern Recognition*, 2008.

[74] R. Haeusler, R. Nair, and D. Kondermann, "Ensemble Learning for Confidence Measures in Stereo Vision," in *Computer Vision and Pattern Recognition*, 2013, pp. 305–312.

[75] S. Zagoruyko and N. Komodakis, "Learning to Compare Image Patches via Convolutional Neural Networks," in *Computer Vision and Pattern Recognition*, Jul. 2015, pp. 4353–4361.

[76] J. Bontarand and Y. LeCun, "Computing the Stereo Matching Cost with a Convolutional Neural Network," Jul. 2015, pp. 1592–1599.

[77] ——, "Stereo Matching by Traning a Convolutional Neural Network to Compare Image Patches," 2015.

[78] W. Luo, A. Schwing, and R. Urtasum, "Efficient Deep Learning for Stereo Matching," *CVPR*, 2016.

[79] N. Mayer, E. IIg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, "A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation," *CoRR*, 2015.

[80] F. Guney and A. Geiger, "Displets: Resolving Stereo Ambiguities using Object Knowledge," *CVPR*, 2015.

[81] A. Shaked and L. Wolf, "Improved Stereo Matching with Constant Highway Networks and Reflective Confidence Learning," in *Computer Vision and Pattern Recognition*, 2017, pp. 4641–4650.

[82] S. Gidaris and N. Komodakis, "Detect, Replace, Refine: Deep Structured Prediction For Pixel Wise Labeling," in *Computer Vision and Pattern Recogni- tion*, 2017, pp. 5248–5257.

[83] M. Schonbeinand and A. Geiger, "Omnidirectional Reconstruction in Augmented Manhattan Worlds," 2014.

[84] K. Yamaguchi, D. McAllester, and R. Urtasun, "Efficient Joint Segmentation, Occlusion Labelling, Stereo and Flow Estimation," 2014.

[85] D. Wei, C. Liu, and W. Freeman, "A Data-Driven Regularization Model for Stereo and Flow," 2014.

[86] M. Menze and A. Geiger, "Object Scene Flow for Autonomous Vehicles," 2015.

[87] N. Einecke and J. Eggert, "A Multi-Block-Matching Approach for Stereo," 2015.

[88] A. Sutherland and D. Pena, "Disparity Estimatino by Simultaneous Edge Drawing," ACCV International Workshop, Tech. Rep., 2016.