12-2011

Wide Baseline Stereo Image Rectification and Matching

Wei Hao
whao@utk.edu

Recommended Citation
To the Graduate Council:

I am submitting herewith a dissertation written by Wei Hao entitled "Wide Baseline Stereo Image Rectification and Matching." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Electrical Engineering.

Mongi A. Abidi, Major Professor

We have read this dissertation and recommend its acceptance:

Seddik M. Djouadi, Ohannes Karakashian, Andreas Koschan

Accepted for the Council:

Carolyn R. Hodges

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)
Wide Baseline Stereo Image Rectification and Matching

A Dissertation Presented for
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Wei Hao
December 2011
Acknowledgement

First and foremost, I am deeply indebted to my parents, Ms. Jinxia Wang and Mr. Guoniu Hao, whose love and encouragement has always been inspiring me during the pursuit for higher education.

I would like to express my sincerest gratitude to my academic advisor, Dr. Mongi Abidi, who fired up my interest in computer vision and taught me the fundamentals of good research, writing, and presentation. His profound knowledge and deep insights in the imaging and vision area has always been enlightening during the whole course of pursuing Ph. D.

Special thanks go to Dr. Andreas Koschan, whose comments and suggestions provided valuable insights into stereo vision and have had tremendous impacts on this research. Also, I would like to thank Dr. Seddik Djouradi and Dr. Ohannes Karakashian. Their advice and counsel has been of equal importance. I greatly appreciate their time and input to this dissertation.

Dr. David Page and Dr. Besma Abidi, who once were research professors in the IRIS Lab, thank you for the opportunity and conversations that have helped a lot on my research.

I owe much to my fellow graduate students and friends in the IRIS Lab, Mr. Woon Cho, Mr. Jacob D’Avy, Mr. Muharrem Mercimek, Mr. Timothy Ragland, Mr. Michael Vaughan and Dr. Hong Chang, Dr. Zhiyu Chen, Dr. Chang Cheng, Dr. Chung-Hao Chen, Dr. Harishwaran Hariharan, Dr. Sung Yoel Kim, Dr. Yi Yao. Thank you all.

The administrative staff in IRIS Lab and the EECS department, Mr. Justin Acuff and Ms. Dana L. Bryson C.P.S., helps me a lot in the last few years. Thank you.
Abstract

Perception of depth information is central to three-dimensional (3D) vision problems. Stereopsis is an important passive vision technique for depth perception. Wide baseline stereo is a challenging problem that attracts much interest recently from both the theoretical and application perspectives. In this research we approach the problem of wide baseline stereo using the geometric and structural constraints within feature sets.

The major contribution of this dissertation is that we proposed and implemented a more efficient paradigm to handle the challenges introduced by perspective distortion in wide baseline stereo, compared to the state-of-the-art. To facilitate the paradigm, a new feature-matching algorithm that extends the state-of-the-art matching methods to larger baseline cases is proposed. The proposed matching algorithm takes advantage of both the local feature descriptor and the structure pattern of the feature set, and enhances the matching results in the case of large viewpoint change.

In addition, an innovative rectification for uncalibrated images is proposed to make wide baseline stereo dense matching possible. We noticed that present rectification methods did not take into account the need for shape adjustment. By introducing the geometric constraints of the pattern of the feature points, we propose a rectification method that maximizes the structure congruency based on Delaunay triangulation nets and thus avoid some existing problems of other methods.

The rectified stereo images can then be used to generate a dense depth map of the scene. The task is much simplified compared to some existing method because the 2D searching problem is reduced to 1D searching.

To validate the proposed methods, real world images are applied to test the performance and comparisons to the state-of-the-art methods are provided. The performance of the dense matching with respect to the changing baseline is also studied.
# Table of Contents

Abstract ........................................................................................................................................ iv
List of Figures.......................................................................................................................... vii

1. Introduction and Motivation.................................................................................. 1
   1.1. The Computational Model of the Human Vision System................................. 1
   1.2 Overview of Stereo Vision...................................................................................... 6
   1.3 Wide Baseline Stereo .............................................................................................. 11
   1.4 Motivation and Problem Statement ...................................................................... 14
   1.5 Contributions and Dissertation Outline .............................................................. 16

2. Algebraic Geometry Models of the Stereo Vision System ......... 20
   2.1 Introduction............................................................................................................... 20
   2.2 Basics of Algebraic Projective Geometry.............................................................. 22
   2.3 A Hierarchy of Geometric Transformations .......................................................... 28
   2.4 The Mathematical Modeling of a Pinhole Camera ................................................ 32
   2.5 The Geometry of Stereo Vision System ............................................................... 38
   2.6 Epipolar Geometry and the Fundamental Matrix ................................................. 40

3. Wide Baseline Sparse Feature Detection and Matching .......... 44
   3.1 The State-of-the-art of Wide Baseline Feature Points Detection and Description ... 45
   3.2 The SIFT Feature Detection and Description ....................................................... 51
   3.3 A Feature Matching Algorithm with Geometric and Structural Constraints in Wide
       Baseline Frames ........................................................................................................... 58
   3.4 Experiment Results and Comparisons with K-D Tree Algorithm ......................... 63

4. Rectification of Wide Baseline Images ................................................. 86
   4.1 The Estimation of Fundamental Matrix ................................................................. 87
   4.2 Related Work .......................................................................................................... 90
   4.3 Uncalibrated Image Rectification using structural congruency ............................ 93
   4.4 Experimental Results and Discussions ................................................................. 100

5. Wide Baseline Dense Matching ............................................................... 124
   5.1 A Survey of Stereo Dense Matching Algorithms .............................................. 124
   5.2 Wide Baseline Stereo Dense Matching ............................................................... 125
   5.3 Experimental Results and Discussions ................................................................. 127

6. Conclusion and Future Work .............................................................. 146
6.1 Summary of the Work........................................................................................................ 146
6.2 Future Work.................................................................................................................. 147

Bibliography .................................................................................................................. 149

Vita .................................................................................................................................... 163
List of Figures

Figure 1-1. The imaging mechanism of human eyes (Source: http://www.nei.nih.gov/photo/eyean/index.htm) ........................................................... 2
Figure 1-2. An illustration of all the layers that make up the retina. (Source: http://www.theness.com/images/blogimages/retina.jpeg) ............................................. 3
Figure 1-3. The visual pathway. Each eye represents two halves of the visual field. (Source: http://www.ski.org/Vision/Images/visualpathway.gif) .................................................. 3
Figure 1-4. Computational Scheme of Human Vision System Proposed by David Marr. .... 4
Figure 1-5. The stereo vision system of a robot. (Source: http://www.robotshop.com) ....... 6
Figure 1-6. The image formation scheme for the machine vision systems. ...................... 7
Figure 1-7. (a) A typical stereo setup; (b) Conceptual diagram of a canonical stereo system. ........................................................................................................... 8
Figure 1-8. Stereo Images and Disparity map generated by laser scanner for a sample stereo pair (Source: http://vision.middlebury.edu/stereo/). .................................................... 10
Figure 1-9. (a) A wide baseline stereo system monitoring certain urban area. (b) 2D surveillance images. (c) 3D information inferred from stereo surveillance images. (Source: www.intergraph.com/sgi/products/geomedia3d.aspx) ...................... 12
Figure 1-10. Example image frames of wide baseline of the Valbonne Church. (Source: http://www.robots.ox.ac.uk/~vgg/data/data-mview.html) ........................................... 13
Figure 1-11. Disparity changes \(d\) and \(d_0\) corresponding to the same depth change with a short baseline and with a wide baseline, respectively.................................................. 15
Figure 1-12. Pipeline of the wide baseline stereo research work. ........................................ 17
Figure 2-1. In projective geometry, there is no concept of parallel lines. ......................... 20
Figure 2-2. The cross-ratio of four collinear points \(X_1, X_2, X_3\) and \(X_4\) ..................... 26
Figure 2-3. The pencil of planes and the definition of cross-ratio in \(\mathbb{P}^3\) ...................... 27
Figure 2-4. Shapes equivalent to a square for different types of transformations ............... 31
Figure 2-5. Illustration of the geometric imaging mechanism of pinhole camera.............. 33
Figure 2-6. The geometric configuration of pinhole imaging........................................... 34
Figure 2-7. Transformation between the retinal plane and image plane.......................... 35
Figure 2-8. The Rigid transformation from the world coordinate system to the image coordinate system ................................................................. 36
Figure 2-9. The calibrated camera is a direction sensor. ............................................... 38
Figure 2-10. A mathematical interpretation of how to infer the depth information from a stereo vision system ................................................................. 40
Figure 2-11. The Epipolar Geometry ............................................................................. 41
Figure 3-1. Three Stages of keypoint features processing ............................................. 46
Figure 3-2. Gaussian Pyramid and DOG detector (Source:[Low-04]). ........................................ 53
Figure 3-3. Generate Keypoint Descriptor from Image Gradients (Source:[Low-04]). ............ 54
Figure 3-4. An example of Gaussian image pyramid of four scales .................................... 55
Figure 3-5. The Difference of Gaussian(DoG) Pyramid. ...................................................... 56
Figure 3-6. The locations of DOG extrema detected in the sample image. ......................... 56
Figure 3-7. The key points remain after removing low contrast extrema. ............................ 57
Figure 3-8. Key points remain after removing edge points. ................................................ 57
Figure 3-9. Final key points detected with scale and orientation. ........................................ 58
Figure 3-10. Two frames from the model house image sequence with wide baseline. ...... 66
Figure 3-11. The matches detected by the K-D Tree method. .......................................... 67
Figure 3-12. The matches detected by the proposed method. ........................................... 68
Figure 3-13. Two frames from the ‘library’ image sequence. ............................................. 69
Figure 3-14. The matches detected by the K-D Tree method. ........................................... 70
Figure 3-15. The matches detected by the proposed method. ........................................... 71
Figure 3-16. Two frames from the Valbonne Church sequence. ....................................... 72
Figure 3-17. The matches detected by the K-D Tree method. ........................................... 73
Figure 3-18. The matches detected by the proposed method. ........................................... 74
Figure 3-19. Two frames from the Cottage sequence. ....................................................... 75
Figure 3-20. The matches detected by the K-D tree method. ........................................... 76
Figure 3-21. The matches detected by the proposed method. ........................................... 77
Figure 3-22. Two frames of the male student face sequence. ............................................ 78
Figure 3-23. The matches detected by the K-D Tree method. ........................................... 79
Figure 3-24. The matches detected by the proposed method. ........................................... 80
Figure 3-25. Two images of a female face with changed expression. ................................ 81
Figure 3-26. The matches detected by the K-D Tree method. ........................................... 82
Figure 3-27. The matches detected by the proposed method. ........................................... 83
Figure 3-28. Comparison of the matching number detected for the data sets by the K-D Tree method and the proposed method. ......................................................... 84
Figure 3-29. Comparison of the good matches’ number detected for the data sets by the K-D Tree method and the proposed method. ......................................................... 85
Figure 3-30. Comparison of the good match rate detected for the data sets by the K-D Tree method and the proposed method. ......................................................... 85
Figure 4-1. An example of image rectification. On the top row, two input images are not rectified. On the bottom row, the two images are rectified using the proposed method. (Source of the input images: http://cvlab.epfl.ch/~tola/daisy.html ) ......................... 87
Figure 4-2. General set up for a stereo system and the illustration of epipolar rectification.

Figure 4-3. Rectified Images.

Figure 4-4. Image rectification with distortion minimized. By applying homographies $H$ and $H'$ to (a) and (b) respectively, we obtain image (a') and (b'). The corresponding pixels are on the same scanline after the operations but it is still very difficult to establish pixel to pixel dense matching for the two triangle shapes in the rectified images.

Figure 4-5. Using proposed method, we apply one more affine transformation $A$ on intermediate result (a') to maximize the shape congruency of the scenes and objects in the two frames. In this way, the task of dense matching is made easier.

Figure 4-6. The Delaunay triangle network generated using the feature points detected on an image of a silo.

Figure 4-7. The epipolar geometry.

Figure 4-8. One sample triangle $A'B'C'$ in net $N_2$ and its corresponding triangle $ABC$ in net $N_1$.

Figure 4-9. The two triangle nets are of the same topology but the corresponding meshes may differ in size.

Figure 4-10. Two wide baseline frames of a model house from a video sequence. (Source: http://www.robots.ox.ac.uk/~vgg/data/data-mview.html)

Figure 4-11. Two images with triangle nets.

Figure 4-12. (a)(b), the rectification results by Hartley’s method. (c),(d), the results of proposed methods. (e) and(f), the optimized and reference triangulation nets.

Figure 4-13. Rectification of wide-baseline stereo, where (a) and (b) are input images. (c) and (d) are results of Hartley’s method. (e) and (f) are results of the proposed methods.

Figure 4-14. Five images of a façade taken by cameras with changing viewpoints. (Data obtained from http://cvlab.epfl.ch/~tola/daisy.html)

Figure 4-15. (a) and (b) are frame 1 and frame 2 of the ‘Façade’ sequence. (c),(d) are frame 1 and 2 with triangle nets.

Figure 4-16. (a),(b) the rectification results by Hartley’s method, with rectified triangle nets.(c) and (d) the rectification results by proposed method, with rectified triangle nets.(e) and (f) results of the proposed method.

Figure 4-17. (a) and (b) are frame 1 and frame 3 of the ‘Façade’ sequence. (c),(d) are frame 1 and 3 with triangle nets.

Figure 4-18. (a),(b) are the rectification results by Hartley’s method, with rectified triangle nets. (c) and (d) the rectification by proposed method, with rectified triangle nets.(e) and (f) results of the proposed method.
Figure 4-19. (a) and (b) frame 1 and frame 4. (c),(d) frame 1 and 4 with triangle nets. .. 110
Figure 4-20. (a),(b) are the rectification results by Hartley's method, with rectified triangle nets. (c) and (d) are the rectification results by proposed method, with rectified triangle nets. (e) and (f) are results of the proposed method. ........................................ 111
Figure 4-21. (a) and (b) frame 1 and frame 5. (c),(d) frame 1 and 5 with triangle nets. ... 112
Figure 4-22. (a),(b) the rectification results by Hartley's method, with rectified triangle nets. (c) and (d) are the rectification results by the proposed method, with rectified triangle nets. (e) and (f) are results of the proposed method. ................................. 113
Figure 4-23. Five images of a fountain taken by cameras with changing viewpoints. (Data obtained from http://cvlab.epfl.ch/~tola/daisy.html) .................................................. 114
Figure 4-24. (a) and (b) are frame 1 and frame 2 of the 'Fountain' sequence. (c),(d) are frame 1 and 2 with triangle nets. .......................................................... 115
Figure 4-25. (a),(b) the rectification results by Hartley's method, with rectified triangle nets. (c) and (d) are the rectification results by the proposed method, with rectified triangle nets. (e) and (f) are results of the proposed method. ................................. 116
Figure 4-26. (a) and (b) are frame 1 and frame 3 of the 'Fountain' sequence. (c),(d) frame 1 and 3 with triangle nets. .......................................................... 117
Figure 4-27. (a),(b) the rectification results by Hartley's method, with rectified triangle nets. (c) and (d) are the rectification results by the proposed method, with rectified triangle nets. (e) and (f) are results of the proposed method. ................................. 118
Figure 4-28. (a) and (b) are frame 1 and frame 4. (c),(d) frame 1 and 4 with triangle nets. .......................................................... 119
Figure 4-29. (a),(b) the rectification results by Hartley's method, with rectified triangle nets. (c) and (d) are the rectification results by the proposed method, with rectified triangle nets. (e) and (f) are results of the proposed method. ................................. 120
Figure 4-30. (a) and (b) are frame 1 and frame 5. (c),(d) frame 1 and 5 with triangle nets. .......................................................... 121
Figure 4-31. (a),(b) the rectification results by Hartley's method, with rectified triangle nets. (c) and (d) are the rectification results by the proposed method, with rectified triangle nets. (e) and (f) are results of the proposed method. ................................. 122
Figure 5-1. Left: The left stereo image, Right: the right stereo image. ......................... 128
Figure 5-2. Left: the ground truth Disparity image for the left stereo image. Right: the ground truth Disparity image for the right stereo image............................ 128
Figure 5-3. Typical output disparity images. ................................................................. 128
Figure 5-4. The contaminated stereo images with Gaussian Noise (Var = 100). ............... 129
Figure 5-5. disparity images generated using the same parameters........................... 129
Figure 5-6. (a),(b) are frame 1 and 2 of the 'Facade' sequences. (c),(d) are rectified images of proposed method. (e),(f) are rectified images of Hartley's method. ............... 130
Figure 5-7. (a) is the disparity map generate from the rectified image pair of proposed method. (b) is the color-coded version of (a). (c) is disparity map generate from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).

Figure 5-8. (a),(b) are frame 1 and 3 of the 'Facade' sequences. (c),(d) are rectified images of proposed method. (e),(f) are rectified images of Hartley's method.

Figure 5-9. (a) is the disparity map generate from the rectified image pair of proposed method. (b) is the color-coded version of (a). (c) is disparity map generate from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).

Figure 5-10. (a),(b) are frame 1 and 4 of the 'Facade' sequences. (c),(d) are rectified images of proposed method. (e),(f) are rectified images of Hartley's method.

Figure 5-11. (a) is the disparity map generate from the rectified image pair of proposed method. (b) is the color-coded version of (a). (c) is disparity map generate from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).

Figure 5-12. (a),(b) are frame 1 and 5 of the 'Facade' sequences. (c),(d) are rectified images of proposed method. (e),(f) are rectified images of Hartley's method.

Figure 5-13. (a) is the disparity map generate from the rectified image pair of proposed method. (b) is the color-coded version of (a). (c) is disparity map generate from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).

Figure 5-14. (a),(b) are frame 1 and 2 of the 'Fountain' sequences. (c),(d) are rectified images of proposed method. (e),(f) are rectified images of Hartley's method.

Figure 5-15. (a) is the disparity map generate from the rectified image pair of proposed method. (b) is the color-coded version of (a). (c) is disparity map generate from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).

Figure 5-16. (a),(b) are frame 1 and 3 of the 'Fountain' sequences. (c),(d) are rectified images of proposed method. (e),(f) are rectified images of Hartley's method.

Figure 5-17. (a) is the disparity map generate from the rectified image pair of proposed method. (b) is the color-coded version of (a). (c) is disparity map generate from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).

Figure 5-18. (a),(b) are frame 1 and 4 of the 'Fountain' sequences. (c),(d) are rectified images of proposed method. (e),(f) are rectified images of Hartley's method.

Figure 5-19. (a) is the disparity map generate from the rectified image pair of proposed method. (b) is the color-coded version of (a). (c) is disparity map generate from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).

Figure 5-20. (a),(b) are frame 1 and 5 of the 'Fountain' sequences. (c),(d) are rectified images of proposed method. (e),(f) are rectified images of Hartley's method.

Figure 5-21. (a) is the disparity map generate from the rectified image pair of proposed method. (b) is the color-coded version of (a). (c) is disparity map generate from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).
1. Introduction and Motivation

This main topic of this research is to build a wide baseline stereo system that can infer depth information of a large outdoor area. Before we start the investigation of the topic, a review of the human vision system is beneficial.

1.1. The Computational Model of the Human Vision System

The perception of the three-dimensional world is a natural gift to human beings and many other kinds of animals. It is the stereo vision system which grants the human being this capability. Computer vision is the scientific and technological branch of building intelligent machines that can process visual information within images or videos. One of the objectives of computer vision or computational vision is to provide a computational model of vision. The concept of vision-related information processing comes from the Human Visual System (HVS) or more generally biological visual systems. Although the core mechanism of HVS remains unknown, it is now possible to imitate the functional components of HVS in man-made vision systems. In other words, we may develop computer vision system that resemble the function of the human vision system computationally without duplicate its mechanism from the psychological and neuroscience perspective. Nevertheless, we can borrow some important ideas and concepts from the evolutionary design of the human vision system. Here we will have a brief look at the functional components of Human Visual System.

Human Visual System is the most important information processing systems for human beings; more than 70% information about the world of a normal human being comes from the vision system. The image formatting mechanism of human eyes is illustrated in Figure 1-1. As we can see clearly, the cornea, lens, and vitreous are normally transparent so that light can reach the retina. Once the light pass through and hit the retina, the light receptors on the retina then transmit the energy of the light to the optical nerve. As we may notice, all
kinds of man-made imaging equipment resemble the image formatting mechanism of human eyes to some degree.

The structure of the human beings’ retina is illustrated in Figure 1-2. Light enters from the bottom and travel up to the receptors (rods and cones) near the top. Then an electrical signal is set up and transmitted by the optic nerve through pathways to the visual signal processing part of the human brain. In digital imaging system, the CCD (charged coupled device) functions similar to a retina, which send the electronic signal to the digital signal processing module so that a digital image is generated.

Finally, the stereo mechanism of visual signal processing in the human brains is illustrated in Figure 1-3. It is the stereospsis characteristics of the image formatting, reception and processing mechanism that grant humans the three-dimensional visual capability. As illustrated, the left half of each eye responds to the right visual field. The electrical signal from the optic nerves finally enters the visual portion of the brain, the “visual cortex”, where the image is actually perceived. The projection of the visual field of each half of the eye is preserved in the brain.
Figure 1-2. An illustration of all the layers that make up the retina. (Source: http://www.theness.com/images/blogimages/retina.jpeg)

Figure 1-3. The visual pathway. Each eye represents two halves of the visual field. (Source: http://www.ski.org/Vision/Images/visualpathway.gif)
As human beings, we perceive the three-dimensional structure of the world around us with apparent ease. However, a complete understanding remains elusive despite of the fact that psychologists have spent decades trying to determine how the human visual system works (See, for instance, [Mar-82], [Pal-99], [Bru-03] and [Liv-08]). Although we don’t know the details of the visual psychology in the HVS, computational models of HVS have been proposed so that computer vision systems can be built to process the visual information captured by various imaging equipment.

David Marr, in his seminal book [Mar-82], proposed one of the most influential computational theories regarding vision. His main idea is that the function of the visual system is to convert images projected onto our retina into representations of the world written in mentalese\(^1\). The process begins with the retinal images from two eyes and proceeds through different levels of representation, as shown in Figure 1-4.

![Figure 1-4. Computational scheme of the Human Vision System Proposed by David Marr.](image)

The six different levels of representation of David Marr’s model are as follows (For details, see [Mar-82]):

\(^1\) According to *The American Heritage Dictionary of the English Language*, Fourth Edition, the definition of mentalese is “a hypothetical language in which concepts and propositions are represented in the mind without words”.
1. **Gray-Level Representation.** The raw output from the rods and cones on the retina. It is a vector of gray-level values indicating the activity of each retinal neuron.

2. **Zero Crossing Map.** According to Marr's theory, the first step the vision system performs is to detect edges or object boundaries in the grey level representation. Usually an object boundary corresponds to a quick change in the intensity of light. Intensity changes can be computed by taking the difference between adjoining pixels in the image. Mathematically, it is the extreme (minimum or maximum) of the first derivative, or equivalently, the zero crossing of the second derivative.

3. **Primal Sketch.** Knowing where the edges of the objects are is not sufficient. The orientation of these edges must be computed, and junction points between the edges must be found. The primal sketch represents these important features along with the edges.

4. **2.5 D Sketch.** The 2.5 D sketch maintains all information about edges in the visual field, distinguishing between the changes in color on the surface of the object, changes in the angles on the surface, and changes due to lighting.

5. **Frame Neutral Sketch.** The brain needs to distinguish between what is moving in the real world and the changes in the images it receives that are due to body and eye motion. The frame neutral sketch represents only the changes in the real world, separating those from the changes due to eye and body motion.

6. **3D Sketch.** The 3D sketch is the final representation computed by the visual system. It provides a three dimensional representation of the objects in the world, allowing us to recognize what those objects are even though they look different from different points of view.

Although this scheme was proposed in the 1980s, it remains the dominant theoretical framework followed by most researchers in the computational vision area until today. Researchers in computer vision have, in parallel, been developing mathematical models for recovering the three-dimensional shape and appearance of objects in imagery. Although in the last several decades many advances in the computational vision research area have been made, a general solution to 3D reconstruction problem remains unsolved. Roughly speaking, the major difficulty of the vision lies in the fact that the three-dimensional vision
is an inverse problem, in which we seek to recover some unknowns when given insufficient information to fully specify the solution (See [Sze-11] for more related discussion).

1.2 Overview of Stereo Vision

In Figure 1-5, a typical stereo vision device is presented. As we may notice, there are two cameras with the same orientation and pose and there is an appropriate distance between the two cameras. Obviously, it is a rough mimic of the stereo system of human beings. The light enter the system from the lenses and then the electronic signal is captured by the sensor chips and through digital signal preprocessing, two digital images can be obtained and be ready for further analysis. Sometimes, there would also be a synchronization facility so that the left and right image captured is synchronized.

![Figure 1-5. The stereo vision system of a robot. (Source: http://www.robotshop.com)](image)

The conceptual imaging scheme of an average camera is shown in Figure 1-6. The image formation mechanism is comprised of two major parts, the geometric factors and the photometric factors. Both the geometric and photometric mechanisms are studied as important clues in vision problems.
From the geometric perspective, the three-dimensional world is projected onto a two-dimensional image plane through image capturing equipment. The mathematical modeling tool for this geometric projection is algebraic projective geometry. In the last several decades, there has been abundant research on the geometric analysis of stereo vision (See for instance, [Fau-93], [Fau-01] and [Har-04]). And this is also the major way we will approach the wide baseline stereo problems. From the physics perspective, the lighting, surface properties and camera optics will all affect the quality of the images we obtain. This is also a rich research area in the past several decades (See for instance, [Fol-95], [Dan-99] and [Deb-99]). In this research we will not make use of the optical information.

In Figure 1-7, we provide a conceptual illustration of a stereo vision system and introduce the technical vocabulary that will be used frequently in the following chapters. From this figure, we can get some intuitive ideas about how the depth information can be inferred from the comparison of the projections of the scene obtained by the left and right camera.
Figure 1-7. (a) A typical stereo setup; (b) Conceptual diagram of a canonical stereo system.
Figure 1-7 (a) shows the stereo system with two cameras. In practice, the cameras could be placed with arbitrary orientation, if only they are focused on the same scene so that a significant overlap in the two views is obtained. Apparently, only the three-dimensional structure of the scene which is visible in both the images captured by the two cameras can be reconstructed. Figure 1-7 (b) illustrates how each image of a scene point relates to the other. $C_L$ and $C_R$ are the camera optical centers, respectively. The line segment connecting the optical centers of the cameras is called the baseline. For multiple cameras the baseline is defined pair-wise for each camera pair.

From the geometry of the drawing, we notice that the projections $P_L$, $P_R$ and $Q_L$, $Q_R$ of the two scene points $P$ and $Q$ respectively, shift to the left in the right camera CCD plate (also known as the image plane) with respect to their location in the left camera CCD plate. This is true for any point in the left image that the point will shift to the left in the right image.

Intuitively, the depth of the scene point $P$ and $Q$ with respect to the camera will be reflected on the projections. We termed $x(P_L) - x(P_R)$ the disparity of the two projections of the scene point $P$ and similarly $x(Q_L) - x(Q_R)$ the disparity of the two projections of the scene point $Q$. The fact that $P$ is farther away than $Q$ is reflected by the constraint $x(Q_L) - x(Q_R) > x(P_L) - x(P_R)$, where $x(.)$ is the x-coordinate of an image point on the image plane. This is why we can infer the depth information of the scene by analyzing the projections of the scene on individual images. The detailed mathematical modeling of the stereo vision facility will be deduced in chapter 2.

For a vivid demonstration of measuring depth information from the disparity map, we can check out the example image shown in Figure 1-8. Figure 1-8 (a) and (b) are two stereo images, while Figure 1-8 (c) is the ground truth of disparity map generated by a laser scanner for the scene. The gray-level of the pixels in the map reflects the depth information of the scene. Ideally, stereo algorithms should be applied to infer the disparity information from the two stereo images.
Figure 1-8. Stereo Images and Disparity map generated by laser scanner for a sample stereo pair (Source: http://vision.middlebury.edu/stereo/).
Essentially, two problems need to be solved to reconstruct a 3D scene using stereo technique. One is the matching of the image pixels, where for a point in one images the disparity of its corresponding point in the other image is determined. The other is 3D reconstruction/triangulation of the scene points applying 2D image coordinates of the matched image points and calibration information of the cameras.

In this dissertation, we will focus on solving the stereo matching problem, which is much more difficult than the other one. We will consider the binocular stereo, where two cameras are located at two viewpoints on the baseline. The discussion on binocular stereo can be easily extended to multiple camera cases if the camera can be treated pair wise. We call the image pairs captured by the two cameras, the left and right stereo images. Then, more specifically, stereo matching can be described as a problem of matching the sites (pixels, features, or image regions) of the left image to their corresponding sites in the right image. Hundreds of algorithms have been published in the literature so far with new algorithms continuously developed and published. A survey of the state-of-the-arts in this area will be presented in Chapter 5.

1.3 Wide Baseline Stereo

In recent years, the increasing number of deployment of wide area surveillance systems demands more advanced wide baseline stereo techniques. Obviously, wide baseline stereo requires fewer types and numbers of equipment to cover a large area and depth information is also available for accurate analysis. For instance, we can use wide baseline stereo systems for large area surveillance as shown in Figure 1-9 (a), which can be used to monitor the large area shown in Figure1-9 (b). What make it different from the average surveillance system is that 3D model of the monitored area can be generated as shown in Figure 1-9 (c) and the cost is reduced greatly compared to traditional surveillance techniques.
A typical image pair of wide baseline is shown in Figure 1-10(a) and (b). They are two frames (frame 0 and frame 60) of wide baseline extracted from a video of Valbonne Church at very different viewing angles. Even though point correspondences between consecutive frames (e.g. frame 0 and 1) may be easily determined, it is a challenging problem to fully and automatically recover correspondences between non-consecutive frames (e.g. frame 0 and 60) with large baseline.

To summarize the characteristics of wide-baseline stereo, we may classify the factors into two categories: geometrically, there is large perspective difference in the stereo images,
such as large scale change, big rotation and projective distortion. On the other hand, the
photometric factors such as non-diffuse reflection, illumination changes and occlusions can
also contribute to the difficulties of processing wide baseline stereo images. We can easily
find these characteristics of wide baseline in the sample images in Figure 1-10.

![Figure 1-10](http://www.robots.ox.ac.uk/~vgg/data/data-mview.html)

(a)                                                                      (b)

Figure 1-10. Example image frames of wide baseline of the Valbonne Church. (Source:
http://www.robots.ox.ac.uk/~vgg/data/data-mview.html)

In the wide baseline cases, the traditional stereo techniques will not generate satisfactory
results. In some cases, they are not applicable at all. A great deal of matching problems can
be traced to the large difference between the views. Different viewpoints imply errors from
occlusions, perspective distortion and false matches near depth jumps and object borders.
When the viewpoints are close, these problems are not so serious, since occlusions and
distortions are relatively small. But in large baseline cases, these are central problems.
A number of other interesting applications also demand a reliable features matching over a wide baseline, such as view morphing [Sei-96] and reconstruction from multiple images (See, for example, [Fau-93], [Luo-97], [Sch-02a] and [Har-04]). In these applications, the external and internal parameters between any two views are also significantly different, and illumination may also be significantly different. All these problems demand innovative solutions, which are the central objectives of this research.

1.4 Motivation and Problem Statement

Apart from the motivation of the theoretical interest, considering the maturity of the narrow baseline stereo techniques, if one can choose the viewpoints of the stereo cameras when collecting data for the purpose of 3D reconstruction, why he bother to build a wide baseline system, or what is the motivation to apply wide baseline techniques?

The first reason is that, the range of disparity can be larger compared to the narrow baseline cases, thus the calculation may be more accurate. As illustrated in Figure 1-11, consider two points $P$ and $Q$ in the scene are projected to both a narrow baseline stereo system and a wide baseline stereo system. It is easy to notice that the disparity between the projections of $P$ and $Q$ are much bigger in the wide baseline case than in the narrow baseline case. In consequence, the resolution of the disparity map should be higher in the wide baseline case. In turn, the measure of the depth change between points $P$ and $Q$ will be more accurate.

Another reason for adoption of a wide baseline setup is the considering of the size of the acquired data for the purpose of large outdoor scene. Traditionally, these kinds of tasks requires huge amount of data captured by a video camera (A good example is the 3D reconstruction method of urban scene described in [Pol-08]). However, from properly selected wide baseline stereo images, one can significantly simplify and speed up the data acquisition procedure and at the same time, reduce the size of data to be stored and processed. This is especially useful when designing a 3D vision system to cover a large size area such as football stadiums, harbors or airports. To cover such a big area, a short
The baseline system may need to capture at least 300-500 images, while the same area may be covered by dozens of images in wide baseline setup.

Now we define the problems we are going to resolve and the assumptions we need in this research. We define wide baseline stereo when any of the following phenomena happens in the stereo image pairs:

- Large translation, rotation, scaling;
- Foreshortening;
- Non-diffuse reflections;
- Significant Illumination changes;
- Significant Occlusions.

The first two cases can be attributed to the geometric factors. The non-diffuse reflection and significant illumination changes are related to the photometric changes. The significant occlusions may be the results of geometric and/or photometric factors.
In this research, we will use key points with regional information as the feature for sparse stereo matching rather than blobs or edges/lines as used by other researchers (for instances, [Bro-05], [Mat-04], [Eld-01] and [Sin-08]). We only use the gray level intensity information and the geometric information as the clues for analysis rather than using the color information or photometric information. (For instance, see [Kos-08], [Vla-09])

In the related literatures, the word ‘wide baseline stereo’ may be used to refer to the sparse wide baseline stereo matching only. Only after year 2007, more people publish results of dense wide baseline stereo matching. In this research, we will consider both the sparse and dense matching problems. As an initial processing, we will generate SIFT feature points as basic information, then we will conduct sparse matching, image rectification and dense matching, sequentially.

We define **sparse wide baseline stereo matching** as generating a correspondence between two appropriately selected point sets on two images frames.

The **dense wide baseline stereo matching** is then defined as generating a correspondence for each scan-line pairs for the two image frames. The output of the dense matching is a disparity map, which contains the depth information of the scene.

We conduct **wide baseline stereo image rectification** to transform the correspondent scan lines so that they are aligned. In this way, the 2D search of the correspondent points will be reduced to 1D search so that the computation complexity is deduced greatly. Rectification also compensates the projective distortion between the two image frames, that is crucial for the wide baseline stereo tasks.

**1.5 Contributions and Dissertation Outline**

In this research we proposed and implemented a wide baseline stereo software system. The system paradigm is depicted in Figure 1-12. The advantage of the proposed system is that
we make use of various geometric and structural constraints to handle the perspective distortion exist in the wide baseline stereo images.

The inputs of the system are two uncalibrated images (or any two image pairs in multi-camera cases), then the system can generate reliable feature points and accurate sparse matching automatically. Based on the knowledge of the feature matching, the two images can be rectified so that the corresponding epipolar lines are aligned with scan lines. What is more important is that, the rectification can compensate the projective distortion existed in the original image pair, so that a dense pixel matching between the two rectified images is possible.
To facilitate the new paradigm, a new sparse matching algorithm which extends the state-of-the-art local feature detectors to larger viewpoint change cases is proposed. Within the wide baseline setup, the feature matching is much more difficult than in the small baseline cases. A survey of local feature detectors is conducted. Based on the survey, the proposed matching algorithm takes advantage of both the local feature descriptor and the topological structure of the feature pattern, and enhances the matching results in the case of large viewpoint change.

In addition, an innovative image rectification method for uncalibrated stereo images is proposed to make efficient dense matching possible. We noticed that the state-of-the-art of rectification methods did not take into account of the needs of further dense matching for the rectified images (such as the method described in [Har-04]) and they only use the rectification techniques to align the epipolar lines with correspondent scan lines. Because the projective distortion is not taken care of properly, it is difficult to attain dense matching for the rectified images. By taking into account of the geometric constraints of the pattern of the feature points, we proposed a rectification method which maximize the structure congruency between the Delaunay Triangulation nets and thus avoid the existing problems of other methods for wide baseline images.

The rectified stereo images can then be used to generate dense depth map of the scene. The task is much simplified compared to some existing method because the 2D searching problem is reduced to 1D searching. And since we introduce affine transformations to compensate for the perspective distortion, the result of our method is also better than the output generated by other existing methods such as in [Har-04].

To validate the proposed methods, real world images are applied to test the performance and comparisons to the state of the art are provided.

The research work presented in this dissertation relies heavily on the projective geometry description of the imaging process. So, we first provide necessary background knowledge
on projective geometry and the mathematical models of stereo vision systems in Chapter 2. For more specific related work, we summarize them in the following chapters.

The core of the dissertation is Chapters 3, 4 and 5. In Chapter 3, we describe the problem of wide baseline sparse matching and propose a new algorithm which makes use of the global pattern constraint on the feature points generated by the SIFT algorithm as described in [Low-04]. Then experimental results on real world images of the proposed algorithm are shown. To validate our algorithm, comparisons with a state-of-the-art algorithm is also provided.

To make dense matching possible and efficient, Chapter 4 is devoted to wide baseline image rectification. In this chapter, we also make use of the global constraint of feature point patterns. From this perspective, a new algorithm which does not require camera calibration is proposed and presented in detail. As usual, experimental results on real world images of the proposed algorithm are shown. To validate the algorithm, comparisons with the state-of-the-art are provided.

In Chapter 5, we describe the dense matching step based on the previous rectification results, which finally fulfill the task of stereo dense matching. To test the performance of the rectification algorithm and dense matching with respect to the changing baselines, two wide baseline image sequences are used for test purpose.

Finally, we summarize the research work and possible future work is suggested.
2. Algebraic Geometry Models of the Stereo Vision System

2.1 Introduction

It seems that Euclidean geometry describes a 3D world quite well. For instance, the length of an object is measureable, intersecting lines determine angles between them, and lines that are parallel on a plane will never meet. But when we consider the mathematical modeling of the imaging process of a camera, the Euclidean geometry is not sufficient, as it is not possible to determine lengths and angles anymore, and parallel lines may intersect. It is the perspective projection techniques developed by the artists and mathematicians in the Renaissance era that can describe the image formation by a pinhole camera or a thin lens. A most noticeable perspective effect is that parallel lines in the real world do not remain parallel in an image, as can be illustrated in Figure 2-1.

![Figure 2-1. In projective geometry, there is no concept of parallel lines.](image)

The perspective property of 2D images of the real world was studied thoroughly since the Renaissance era. But the strictly mathematical description of the projective geometry was
first pioneered by the 17 century mathematicians Johannes Kepler and Gerard Desargues etc, who established the basic concepts and theorems of synthetic projective geometry.

At the end of 18th and beginning of 19th century, the fundamental work of Gaspard Monge and Jean-Victor Poncelet helped to establish projective geometry as a standalone mathematical brunch. Thanks to the introduction of *homogeneous coordinates* and other algebraic tools by Mobius and Julius Plücker, projective geometry can be treated elegantly in the algebraic language.

Based on C.F. Klein’s great idea of geometric transformation groups, each kind of geometry has a group of transformations associated with it, which leaves certain properties of each kind of geometry invariant. These invariants, when recovered for certain geometry, allow for an upgrade to the next higher-level geometry.

All the linear geometries involved in the imaging of pinhole cameras can generally be divided into four geometry groups or strata, of which Euclidean geometry is the one with the smallest transformation group with the most invariants. The most general geometry is *projective geometry*, which forms the basis of all other geometries. Other geometries include *affine geometry* and *metric geometry*.

By introducing the algebraic representations of the geometric objects such as point, line, plane, conics and geometric transformations, it is possible for the computer to handle the task of geometric treatment of the vision problems so that we need to introduce the fundamental knowledge which is the basis of the further discussions in the following chapters.

Projective geometry can be used to describe perspective projections so that it models the imaging process very well. Once a model of this perspective projection is obtained, it is then possible to upgrade the projective geometry later to Euclidean, via the affine and
metric geometries. Algebraic projective geometry forms the basis of many computer vision tasks, especially in the fields of 3D reconstruction from images and camera self-calibration.

An overview of algebraic projective geometry is given in Section 2.2 and some of the notation used throughout the text is introduced. Objects such as points, lines, planes, conics and quadrics are described in two and three dimensions.

A standard book covering all aspects of algebraic projective geometry from the point of view of a mathematician is [Sem-79]. In the computer vision community, Faugeras applies systematic results of projective geometry to 3D vision and reconstruction problems in his books [Fau-93] and [Fau-01]. Hartley and Zisserman summarize the progress in this area in the last two decades in [Har-04]. Other concise introductions to projective geometry are [Moh-96] and [Bir-98]. Stratification is described in [Fau-01] and [Pol-99a]. The following sections are based roughly on the introductions to projective geometry in [Fau-93], [Fau-01] and [Har-03]. For further information on this topic, you are referred to these literatures.

### 2.2 Basics of Algebraic Projective Geometry

#### 2.2.1 Overview and notations

Projective geometry allows for the modeling of perspective projections, and thus models the imaging process of CCD cameras. This section presents an overview of algebraic representations of projective geometry objects and introduces some common notations.

Consider the \((n+1)\)-dimensional vector space without its zero elements, \(\mathbb{R}^{n+1} - \{(0,0,...,0)\}\), and define an equivalent relation between \((n+1)\)-dimensional vectors as follows:

\[
[x_1, \cdots, x_{n+1}]^T \equiv [x'_1, \cdots, x'_{n+1}]^T
\]

(2-1)

iff \(\exists \alpha \neq 0\) s.t. \([x_1, \cdots, x_{n+1}]^T = \alpha[x'_1, \cdots, x'_{n+1}]^T\)
In this way, the projective space $P^n$ can be thought as the quotient space under this equivalence relation. Then we can represent points in the projective space in *homogeneous* coordinates, which is denoted in uppercase with a tilde, e.g., $\tilde{x}$. Especially, such points can often be represented by vectors with a 1 at the rightmost position, e.g., $[x_1, \ldots, x_n, 1]^T$. This point is equivalent to any point that differs only by a non-zero scalar factor. Thus, a mapping from the $n$-dimensional Euclidean space $\mathbb{R}^n$ into $P^n$ is given by:

$$\mathbb{R}^n \mapsto P^n: [x_1, \ldots, x_n]^T \rightarrow [x_1, \ldots, x_n, 1]^T$$

(2-2)

In $P^n$, only the points with the last coordinate 0, i.e., $[x_1, \ldots, x_n, 0]^T$ don’t have a Euclidean counterparts in $\mathbb{R}^n$, which represent *points at infinity* in a particular direction. We can take it as a limiting case of $[x_1, \ldots, x_n, \alpha]^T$ assuming that $\alpha \rightarrow 0$, which in $P^n$ is equivalent to $[x_1/\alpha, \ldots, x_n/\alpha, 1]^T$. This corresponds to a point in $\mathbb{R}^n$ approaching infinity in the direction $[x_1/\alpha, \ldots, x_n/\alpha]^T$.

A *collineation* or projective point transformation is any linear mapping $P^n \rightarrow P^m$ which preserves collinearity of any set of points. It can be represented by a regular $(m + 1) \times (n + 1)$ matrix $H$, s.t. $\tilde{y} \equiv H\tilde{x}$. The equation is up to any scale so that for any nonzero scalar $\lambda$, $H$ and $\lambda H$ represent the same collineation. A special case is that when $H$ is a $(n + 1) \times (n + 1)$ matrix, $H$ defines a collineation from $P^n$ to itself.

A *projective basis* for $P^n$ is any set of $(n+2)$ points of $P^n$, such that no $(n+1)$ of them are linearly dependent. The set $e_i = [0, \ldots, 1, \ldots, 0]^T$, for $i = 1, \ldots, n+1$, where 1 is at the $i$th position, plus $e_{n+2} = [1, 1, \ldots, 1]^T$ form the *standard projective basis*. In consequence, a projective point of $P^n$ represented by any of its coordinate vectors $x$ can be described as a linear combination,

$$x = \sum_{i=1}^{n+1} x_i e_i$$

(2-3)

Any projective basis can be transformed by a collineation into a standard projective basis. A collineation can also map a projective basis onto a second projective basis.
2.2.2 Representations of Objects in a Projective Plane

In this research, we will concentrate on the three-dimensional projective space (or simply projective space as is commonly used in the analytical geometry), and the two-dimensional projective planes.

The two dimensional projective space $P^2$ is known as the projective plane. A point in $P^2$ is defined as a 3-vector $x = [x_1, x_2, x_3]^T$, with $(u, v) = (x_1/x_3, x_2/x_3)$ its Euclidean coordinates on the plane. A line can be identified as a 3-vector $l$ with coordinate $[l_1, l_2, l_3]^T$ and a point $x$ is located on the line $l$ if and only if,

$$l^T x = 0 \quad (2-4)$$

Notice that in Equation (2-4), the roles of $l$ and $x$ are interchangeable. This equation can be interpreted as a point equation, which means that a point $x$ is determined by a bunch of lines through it, or it can be interpreted as a line equation, which means that a line $l$ is determined by a set of points on it. This complies with the principle of duality. According to the principle of duality, any theorem or proposition that is true for the projective plane can be rephrased by substituting "points" for "lines" and "lines" for "points", and the resulting statement will also be true.

It is easy to shown that the homogeneous coordinate vector of the line $l$ through two points $x$ and $y$ is the cross product of the homogeneous coordinate vector of the points $x, y$:

$$l = x \times y \quad (2-5)$$

, which can also be expressed as follows,

$$l = [x]_x y \quad (2-6)$$

, where

$$[x]_x = \begin{bmatrix} 0 & x_3 & -x_2 \\ -x_3 & 0 & x_1 \\ x_2 & -x_1 & 0 \end{bmatrix} \quad (2-7)$$
be the anti-symmetric matrix of coordinate vector \( x \) associated with the cross product. According to the principle of duality, the intersection point of two lines can also be determined by the cross product, \( x = l_1 \times l_1 \).

All the lines passing through a specific point form a pencil of lines. If two lines \( l_1 \) and \( l_2 \) are within this pencil, then all the other lines are some linear combinations of \( l_1 \) and \( l_2 \):

\[
l = \lambda_1 l_1 + \lambda_2 l_2
\]

where \( \lambda_1, \lambda_2 \) are scalars.

**Collineation**

Collineations from \( P^2 \) to \( P^2 \) are represented by \( 3 \times 3 \) invertible matrices \( H \), defined up to a nonzero scale factor. Collineations transform points, lines and pencil of lines to points, lines and pencil of lines respectively. In \( P^2 \), collineation are also called *homographies*. A point \( x \) is transformed as follows:

\[
x' = Hx
\]

The induced line transformation of a line \( l \) to \( l' \) can be determined by (2-11), with \( H^{-T} = (H^{-1})^T = (H^T)^{-1} \):

\[
l' = H^{-T} l
\]

**Cross-Ratio**

Cross-ratio is the most important invariant of projective geometry. If four points \( x_1, x_2, x_3 \) and \( x_4 \) are collinear, then the *cross-ratio* can be defined as follows:

\[
\{x_1, x_2; x_3, x_4\} = \frac{\lambda_1 - \lambda_3}{\lambda_1 - \lambda_4} : \frac{\lambda_2 - \lambda_3}{\lambda_2 - \lambda_4}
\]

The cross-ratio is invariant under any collineation of projective plane. A similar cross-ratio can be derived for four lines in a pencil: for four lines \( l_1, l_2, l_3 \) and \( l_4 \) of \( P^2 \) intersecting at a point, their cross-ratio \( \{l_1, l_2; l_3, l_4\} \) is defined as the cross-ratio \( \{x_1, x_2; x_3, x_4\} \) of their four points of intersection with any line \( l \) not going through their point of intersection. See Figure 2-2 for a graphical illustration.
Conics

In Euclidean geometry, second-order curves such as ellipses, parabolas and hyperbolas are well defined. In projective geometry, all these notions are known as conics, because they are equivalent under appropriate collineation. A (point) conic $C$ is defined as the locus of all the points on the projective plane that satisfies the following equation:

$$S(x) = x^T C x = 0$$

(2-12)

, where $C$ is a $3 \times 3$ symmetric matrix defined up to a non-zero scalar. This implies that the conic can be identified with $C$ and be determined by 5 independent parameters.

2.2.3 Representations of Objects in the Projective Space

The three-dimensional projective space $P^3$ is called the projective space. A point $x$ of $P^3$ is defined by a 4-vector $x = [x_1 \ x_2 \ x_3 \ x_4]^T$. The dual entity of a point in $P^3$ is a plane, which can also be represented by a 4-vector $\pi = [\pi_1 \ \pi_2 \ \pi_3 \ \pi_4]^T$ with equation of

$$\sum_{i=1}^{4} \pi_i x_i = 0$$

(2-13)

In other words, a point $x$ is located on a plane $\pi$ if the following equation is true:
In \( P^3 \), the structure which is analogous to the pencil of lines of \( P^2 \) is the pencil of planes, or the set of all planes that intersect in a certain line.

![Figure 2-3. The pencil of planes and the definition of cross-ratio in \( P^3 \).](image)

The cross-ratio in \( P^3 \) can be defined as four planes \( \pi_1, \pi_2, \pi_3 \) and \( \pi_4 \) of \( P^3 \) that intersect at a line \( l \). That means that the cross-ratio \( \{\pi_1, \pi_2; \pi_3, \pi_4\} \) is defined as the cross-ratio \( \{l_1, l_2; l_3, l_4\} \) of their four lines of intersection with any plane not going through \( l \), as shown in Figure 2-3.

**Collineation in \( P^3 \)**

Collineations in \( P^3 \) are defined by \( 4 \times 4 \) invertible matrices \( T \), defined up to a non-zero scale factor. It can be shown that collineation transform points, lines, planes and pencil of planes to points, lines, planes and pencil of planes respectively, and preserve the cross-ratios. Transformations \( T \) of points \( x \) and the induced transformation with plane \( \pi \) in \( P^3 \) are as follows:
\[ x' \sim Tx \]  
\[ \pi' \sim T^{-T} \pi \]  

### Quadrics

The equivalence to conic of \( P^2 \) in \( P^3 \) is a quadric. A quadric is the set of all points \( x \) satisfying:

\[ x^T Q x = 0 \]  

where \( Q \) is a 4x4 symmetric matrix defined up to a non-zero scale factor with a degree of freedom of 9.

The dual quadric is the locus of all planes \( \pi \) satisfying:

\[ \pi^T Q^* \pi = 0 \]  

where \( Q^* \) is a \( 4 \times 4 \) symmetric matrix defined up to a scale factor and also depends on 9 independent parameters. Transformations \( T \) of the quadric and dual quadric are as follows (similar to transformations of the conic as in the projective plane):

\[ x'^T Q' x' \sim (Tx)^T T^{-T} QT^{-1} (Tx) = 0 \]  
\[ \pi'^T Q'^* \pi'^* \sim \pi^T T^{-1} Q^* T^T T^{-T} \pi = 0 \]  

And therefore, we have the related transformations,

\[ Q' \sim T^{-T} QT^{-1} \]  
\[ Q'^* \sim T Q^* T^T \]  

### 2.3 A Hierarchy of Geometric Transformations

The characterization of geometries in algebraic language naturally leads us to a classification of the geometric transformations according to the behaviors of the
transformation matrices on geometric entities. This follows the idea proposed by C.F. Klein in his renowned Erlangen program in 1872.

Among all the linear mappings from the set of all the points in the projective plane $P^2$ to itself, it is easy to verify that the point transformation in (2-21) will preserve the Euclidean distance, thus also angles and other related measures:

$$
\begin{pmatrix}
\varepsilon & 0 \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
\cos \theta & -\sin \theta & t_x \\
\sin \theta & \cos \theta & t_y \\
0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
x \\
y \\
1
\end{pmatrix}
$$

(2-21)

The upper left $2 \times 2$ sub-matrix of the transformation matrix is orthogonal. If $\varepsilon = 1$, the transformation is orientation preserving, and is called an Euclidean transformation. If $\varepsilon = -1$, the transformation will reverse the orientation. All transformations like (2-21) are called isometries. And it is easy to verify that all orientation-preserving isometries, i.e., the Euclidean transformations form a group.

A similarity transformation is a transformation with a non-zero isotropic scaling factor, as represented by the transformation (2-22):

$$
\begin{pmatrix}
x' \\
y' \\
1
\end{pmatrix} =
\begin{pmatrix}
 s & 0 & t_x \\
 0 & s & t_y \\
 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
x \\
y \\
1
\end{pmatrix}
$$

(2-22)

,where the scalar $s$ represents the isotropic scaling. A similarity transformation preserves the shape while alters the size. But the ratio of lengths or areas is invariant under a similarity transformation. All the similarity transformations also form a group. And the Euclidean transformation group is a subgroup of the similarity one.

An affine transformation is a non-singular transformation followed by a translation. It can be represented as in (2-24):
\[
\begin{pmatrix}
    x' \\
    y' \\
    1
\end{pmatrix} =
\begin{bmatrix}
    a_{11} & a_{12} & t_x \\
    a_{21} & a_{22} & t_y \\
    0 & 0 & 1
\end{bmatrix}
\begin{pmatrix}
    x \\
    y \\
    1
\end{pmatrix}
\] (2-23)

The upper left \(2 \times 2\) non-singular submatrix \(A\) can always be decomposed as:

\[
A = R(\theta)R(-\varphi)DR(\varphi)
\] (2-24)

Here, \(R(.)\) is a pure rotation matrix and \(D\) is a non-isotropic scaling. Compared to a similarity, the new element introduced is the non-isotropic scaling. The invariants under an affine transformation are: parallel lines, ratio of lengths of parallel line segments and ratio of areas. All the affine transformations form a group. And the similarity group is a subgroup of the affine group.

The most general linear transformations are the \textit{projective transformations} as in (2-25):

\[
\begin{pmatrix}
    x' \\
    y' \\
    1
\end{pmatrix} =
\begin{bmatrix}
    a_{11} & a_{12} & t_x \\
    a_{21} & a_{22} & t_y \\
    v_1 & v_2 & v
\end{bmatrix}
\begin{pmatrix}
    x \\
    y \\
    1
\end{pmatrix}
\] (2-25)

The matrix has nine elements but only their ratio is significant, so the transformation is specified by 8 parameters. The most important invariant under projective transformations are cross-ratio and the collinearity, so a projective transformation is also called collineation as already discussed in the previous sections.

For a vivid illustration, Figure 2-4 presents an example of a 2D object that is equivalent to a square under different geometric transformations.
In the case of projective space $P^3$, the classification is similar to the case of the projective plane. Here we list the main properties of the different strata briefly in Table 1-1. The geometric transformation groups are listed and the number of degrees of freedom, representations of transformations and their specific invariants are shown also.

The concept of stratification of geometries is closely related to the groups of transformations acting on geometric entities and leaving invariant some properties of configurations of these elements. Attached to projective geometry is the group of projective transformations, attached to the affine stratum is the group of affine transformations, attached to the metric stratum is the group of similarities and attached to the Euclidean stratum is the group of Euclidean transformations. It is important that these groups are...
subgroups of each other, e.g. the metric group is a subgroup of the affine group and both are subgroups of the projective group.

Table 1-1. Number of degrees of freedom, transformations and invariants corresponding to the different geometric hierarchy.

<table>
<thead>
<tr>
<th>Strata of geometries</th>
<th>Degree of Freedom</th>
<th>Transformation Matrix</th>
<th>Invariants</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projective</td>
<td>15</td>
<td>$T_{P} = \begin{bmatrix} p_{11} &amp; p_{12} &amp; p_{13} &amp; p_{14} \ p_{21} &amp; p_{22} &amp; p_{23} &amp; p_{24} \ p_{31} &amp; p_{32} &amp; p_{33} &amp; p_{34} \ p_{41} &amp; p_{42} &amp; p_{43} &amp; p_{44} \end{bmatrix}$</td>
<td>Cross-ratio, collinearity</td>
</tr>
<tr>
<td>Affine</td>
<td>12</td>
<td>$T_{A} = \begin{bmatrix} a_{11} &amp; a_{12} &amp; a_{13} &amp; a_{14} \ a_{21} &amp; a_{22} &amp; a_{23} &amp; a_{24} \ a_{31} &amp; a_{32} &amp; a_{33} &amp; a_{34} \ 0 &amp; 0 &amp; 0 &amp; 1 \end{bmatrix}$</td>
<td>Relative distances along direction, parallelism plane at infinity</td>
</tr>
<tr>
<td>Metric</td>
<td>7</td>
<td>$T_{M} = \begin{bmatrix} \sigma r_{11} &amp; \sigma r_{12} &amp; \sigma r_{13} &amp; t_{x} \ \sigma r_{21} &amp; \sigma r_{22} &amp; \sigma r_{23} &amp; t_{y} \ \sigma r_{31} &amp; \sigma r_{32} &amp; \sigma r_{33} &amp; t_{z} \ 0 &amp; 0 &amp; 0 &amp; 1 \end{bmatrix}$</td>
<td>Relative distances, angles, absolute conic</td>
</tr>
<tr>
<td>Euclidean</td>
<td>6</td>
<td>$T_{E} = \begin{bmatrix} r_{11} &amp; r_{12} &amp; r_{13} &amp; t_{x} \ r_{21} &amp; r_{22} &amp; r_{23} &amp; t_{y} \ r_{31} &amp; r_{32} &amp; r_{33} &amp; t_{z} \ 0 &amp; 0 &amp; 0 &amp; 1 \end{bmatrix}$</td>
<td>Absolute distances</td>
</tr>
</tbody>
</table>

Usually the world is perceived as a Euclidean 3D space. While in some cases (e.g. starting from images) it is not possible or desirable to use the full Euclidean structure of 3D space. It is interesting to only deal with the more restricted and thus simpler structure of projective geometry.

**2.4 The Mathematical Modeling of a Pinhole Camera**

Once the framework of projective geometry is introduced, it is possible to define 3D Euclidean space as embedded in a projective space $P^3$. In a similar way, the image plane of
the camera is embedded in a projective space $P^2$. Then a collineation exists which maps the 3D real world space to the image plane, $P^3 \mapsto P^2$ via a $3 \times 4$ matrix. In this research, we will use the images captured by average CCD cameras or video recorders. The imaging mechanism of the CCD camera and video recorder can be modeled as pinhole cameras with certain tolerant error. The imaging process can then be illustrated geometrically as in Figure 2-5.

![Illustration of the geometric imaging mechanism of pinhole camera.](image)

Figure 2-5. Illustration of the geometric imaging mechanism of pinhole camera.

To apply algebraic geometry in this modeling problem, we first set up two image coordinate systems as shown in Figure 2-6. In this figure, $O$ is the optical center of the camera. $f$ is the focal length of the camera. $OZ$ is the optical axis. The image plane is certain portion of the plane that is of focal length distance from the optical center and vertical to the optical axis. As shown in the illustration of Figure 2-6, a scene point $W$ in the three dimensional world is projected onto a point $m$ on the two dimension image plane. We set up two coordinate systems for the calculation. The first is the three dimensional frame $\{O; \vec{X}, \vec{Y}, \vec{Z}\}$, the second frame is the two dimensional frame $\{P; \vec{x}, \vec{y}\}$.
Let’s assume that a point $\mathbf{W} = [X,Y,Z]^T$ in the 3D scene is projected to a point $\mathbf{m} = [x,y]^T$ on the image plane. Then we have equation (2-26a),

$$
\begin{align*}
    x &= \frac{f}{Z} X \\
    y &= \frac{f}{Z} X
\end{align*}
$$

(2-26a)

Or if we represent point $\mathbf{W}$ and its image $\mathbf{m}$ as homogeneous coordinate vectors and the collineation from $p^3 \mapsto p^2$ as a matrix, we have,

$$
\begin{bmatrix}
    \bar{x} \\
    \bar{y} \\
    1
\end{bmatrix} = \begin{bmatrix}
    f & 0 & 0 & 0 \\
    0 & f & 0 & 0 \\
    0 & 0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
    X \\
    Y \\
    Z \\
    1
\end{bmatrix},
\bar{W} = \begin{bmatrix}
    X \\
    Y \\
    Z \\
    1
\end{bmatrix},
\bar{m} = \begin{bmatrix}
    \bar{x} \\
    \bar{y} \\
    1
\end{bmatrix}
$$

(2-26b)

For a CCD digital camera, we often use pixels as the metric unit, so we also need the transformation from the image plane to the retina plane. Here the \textit{retina plane} is defined as the digital image plane measured in pixels. The transformation from the image plane measured in mm to the retina plane measured in pixel can be established as shown in Figure 2-7.
In Figure 2-7, we have two frames, i.e., Frame A: \{O; \bar{u}, \bar{v}\} and Frame B: \{O'; \bar{x}, \bar{y}\}. And we have equation (2-27),

\[
\begin{align*}
\bar{x} &= k_u \bar{u} \\
\bar{y} &= k_v \bar{v}
\end{align*}
\]

(2-27)

\[
P_A = (u, v)^T \\
P_B = (x, y)^T \\
O_A' = (u_0, v_0)^T
\]

, where \(k_u\) and \(k_v\) are the effective pixel numbers per mm along the \(u\) and \(v\) axes (pixel/mm). From the equality \(\overrightarrow{OP} = \overrightarrow{OO'} + \overrightarrow{O'P}\), we have,

\[
\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + \begin{bmatrix} k_u x \\ k_v y \end{bmatrix} 
\]

(2-28)

Or equivalently,

\[
\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} k_u & 0 & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} 
\]

(2-29)

For we know that,

\[
\begin{align*}
\overrightarrow{OP} &= (\bar{u}, \bar{v}) \begin{bmatrix} u_0 \\ v_0 \end{bmatrix} + (\bar{u}, \bar{v}) \begin{bmatrix} u \\ v \end{bmatrix} \\
\overrightarrow{O'P} &= (\bar{x}, \bar{y}) \begin{bmatrix} x \\ y \end{bmatrix} = (\bar{u}, \bar{v}) \begin{bmatrix} k_u x \\ k_v y \end{bmatrix}
\end{align*}
\]

(2-30)
In real world applications, it is often convenient to set up one more three dimensional coordinate system called the World Coordinate System, as shown in Figure 2-8. The World Coordinate System can be represented as $W: \{O_W; \vec{X}_w, \vec{Y}_w, \vec{Z}_w\}$.

![Figure 2-8. The Rigid transformation from the world coordinate system to the image coordinate system.](image)

Given any point $\mathbf{P}$ in the three-dimensional space, its coordinates in the camera coordinate system and the world coordinate system can be represented as linear combinations of the coordinate basis vectors, or shortly, $P_c = [x_c\quad y_c\quad z_c]^T$ and $P_w = [x_w\quad y_w\quad z_w]^T$. Let’s also assume that the coordinate of the original point of the world coordinate system locates at $O_w = [x_t\quad y_t\quad z_t]^T$ in the camera coordinate system. And we assume that the rotation transformation between the basis of the world coordinate system and the camera coordinate system can be represented by a 3x3 orthogonal matrix $\mathbf{R}$. Then from the fact that $\overline{O_cP} = \overline{O_cO_w} + \overline{O_wP}$ and the fact that,

$$
\begin{align*}
\overline{O_cP} &= x_c\overline{X_c} + y_c\overline{Y_c} + z_c\overline{Z_c} \\
\overline{O_cO_w} &= x_t\overline{X_c} + y_t\overline{Y_c} + z_t\overline{Z_c} \\
\overline{O_wP} &= \mathbf{R}x_w\overline{X_c} + \mathbf{R}y_w\overline{Y_c} + \mathbf{R}z_w\overline{Z_c}
\end{align*}
$$

(2-31)
Compare both sides of the equalities listed in (2-31), we have,

\[
\begin{bmatrix}
    x_c \\
    y_c \\
    z_c 
\end{bmatrix}
= R
\begin{bmatrix}
    x_w \\
    y_w \\
    z_w 
\end{bmatrix} +
\begin{bmatrix}
    x_t \\
    y_t \\
    z_t 
\end{bmatrix}
\]

Or in a compact form,

\[
\begin{bmatrix}
    x_c \\
    x_c \\
    x_c \\
1 
\end{bmatrix}
= \begin{bmatrix} R & t \end{bmatrix}
\begin{bmatrix}
    x_w \\
    x_w \\
1 
\end{bmatrix}
\quad (2-32)
\]

Here \( R \) is the transformation between the two bases of the coordinate systems, as in (2-33):

\[
\begin{bmatrix}
    X_w \\
    Y_w \\
    Z_w 
\end{bmatrix}
= \begin{bmatrix}
    X_c \\
    Y_c \\
    Z_c 
\end{bmatrix} R
\quad (2-33)
\]

Combine (2-29) and (2-32), we know that equation (2-34) holds,

\[
\begin{bmatrix}
    u \\
    v \\
1 
\end{bmatrix}
= \frac{1}{Z}
\begin{bmatrix}
    f k_u & 0 & u_0 & 0 \\
    0 & f k_v & v_0 & 0 \\
    0 & 0 & 1 & 0 
\end{bmatrix}
\begin{bmatrix} R & t \end{bmatrix}
\begin{bmatrix}
    x_w \\
    y_w \\
    z_w \\
1 
\end{bmatrix}
= [K[R|t]]
\begin{bmatrix}
    x_w \\
    y_w \\
    z_w \\
1 
\end{bmatrix}
\quad (2-34)
\]

In this way, we finally establish the mapping from the world coordinate of a scene point measured in the unit of mm to its image points measured in the unit of pixel.

\[
sm = \begin{bmatrix}
    u \\
    v \\
1 
\end{bmatrix}
= K[R|t]
\begin{bmatrix}
    x_w \\
    y_w \\
    z_w \\
1 
\end{bmatrix}
= PM
\quad (2-34')
\]

where \( K = \begin{bmatrix}
    \alpha_u & 0 & u_0 \\
    0 & \alpha_v & v_0 \\
    0 & 0 & 1 
\end{bmatrix} \) and \( \alpha_u = f k_u, \alpha_v = f k_v \), being the focal length measured in width and height of the pixel. \( \vec{m} \) is the homogeneous coordinate vector of the image point and \( \vec{M} \) is the homogeneous coordinate vector of the scene point. \( P \) is called the projective matrix of the camera.
It is obvious that the matrix $K$ is determined by the intrinsic parameters of the camera and the matrix $[R|t]$ embodies the pose and orientation information of the camera, with respect to the world coordinate system. The procedure of determining the matrix $K$ and $[R|t]$ is called *camera calibration*.

### 2.5 The Geometry of Stereo Vision System

Based on the geometric camera model derived in the previous section, we can now deduce the basic equations for stereo vision. As shown in Figure 2-9, the coordinate of the optical center $C = [x_c \ y_c \ z_c]^T$ with respect to the world coordinate system can be determined if the projective matrix $P$ is known.

![Figure 2-9. The calibrated camera is a direction sensor.](image)

Because $C$ is the only point in the projective space that has no image, or equivalently, it is the null space of matrix $P$. If we write $P$ as a blocked matrix $[Q|\tilde{q}]$ where $Q$ is the $3 \times 3$ matrix on the left and $\tilde{q}$ is the $3 \times 1$ vector on the right, then we have equation (2-35),

$$
0 = P\tilde{C} = K[R|t]\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = [Q|\tilde{q}]\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}
$$

(2-35)
\[ \mathbf{Qc} + \overline{\mathbf{q}} = \mathbf{0} \]

From (2-35), we can infer the inhomogeneous coordinate vector of the optical center with respect to the world coordinate system \( \mathbf{C} = [x_c \ y_c \ z_c]^T \),

\[ \mathbf{C} = -\mathbf{Q}^{-1}\overline{\mathbf{q}} \tag{2-36} \]

Then all the points on the optical ray \( \mathbf{MC} \) in Figure 2-9 can be represented by the parametric point equation of

\[ \mathbf{MC} = \{ \mathbf{\bar{w}} = \mathbf{c} + \lambda \mathbf{Q}^{-1}\overline{\mathbf{m}} ; \lambda \in \mathbb{R} \} \tag{2-37} \]

Equation (2-37) implies that once a camera is calibrated, it can be used to determine from a image point \( \mathbf{m} \) the optical ray on which its scene point \( \mathbf{W} \) is located. Or in other words, a calibrated camera is a sensor of directions. This also explains in mathematics why we lose the depth information during the course of imaging. For all the point on the same optical ray will have the same image point on the image plane.

However, if we have more than one images of the same point in the scene, we can easily reconstruction the geometric structure of the scene by finding the intersection of two rays, as is shown in Figure 2-10.

In Figure 2-10, once we have two images \( \mathbf{m} \) and \( \mathbf{m}' \) of the same scene point \( \mathbf{M} \) captured by two calibrated cameras, the 3D coordinates of \( \mathbf{M} \) should be the intersection of ray \( \mathbf{I} \) and ray \( \mathbf{2} \). Now it is clear that the two most important problems of stereo vision are:

- Finding the geometric calibration information of the cameras;
- Establishing the correspondence of the features in different views.

As stated in the first chapter, in this research, we will concentrate on the second problem.
Figure 2-10. A mathematical interpretation of how to infer the depth information from a stereo vision system.

2.6 Epipolar Geometry and the Fundamental Matrix

In our research, we make no assumption on camera calibration information. Instead, we rely heavily on the epipolar geometry, which is the intrinsic projective geometry between two views. It is the intrinsic geometry in the sense that can be determined on the camera’s internal parameters and relative pose.

As illustrated in Figure 2-11, we can find there are important geometric constraints on the locations of the two image points \( \mathbf{m} \) and \( \mathbf{m}' \) of the same scene point \( \mathbf{W} \). In fact, Any scene point \( \mathbf{W} \) and the two optical centers of the left and right camera \( \mathbf{C} \) and \( \mathbf{C}' \) will define an epipolar plane \( \pi \). The plane \( \pi \) will intersect with the two image plane \( I_1 \) and \( I_2 \) on two epipolar lines \( l_1 \) and \( l_2 \), respectively.
The linear segment of $CC'$ is the baseline of the stereo system. It intersects with the image planes at epipoles $e$ and $e'$ respectively. Obviously, all the epipolar lines will intersect at the epipoles $e$ and $e'$ respectively. No matter where the location of the scene point $M$ is within the space, the corresponding epipolar planes will always contain the baseline $CC'$ and the epipoles $e$ and $e'$, i.e., there is a pencil of epipolar planes. This gives us important clues on finding the correspondent point, i.e., the correspondence of image point $m$ will definitely be on the epipolar line $l_2$, and symmetrically, the correspondence of image point $m'$ will always be on the epipolar line $l_1$.

The fundamental matrix is the algebraic representation of epipolar geometry. From Figure 2-11, the epipolar line $l_2$ can be viewed as the projection of the ray from the point $m$ through the camera optical center $C$ of the left camera. This is a linear mapping from $m$ to $l_2$:

$$l_2 = Fm$$  \hspace{1cm} (2-38)

If given the projective matrix of the two cameras $P$ and $P'$, the ray $Cm$ can be represented from (2-36) as a parameterized point set with respect to $\lambda$,

$$M(\lambda) = P^+m + \lambda C$$  \hspace{1cm} (2-39)
Here $P^+$ is the pseudo-inverse of $P$, i.e., $PP^+ = I$. It is obvious that $M(0) = P^+m$ and $(\infty) = C$. These two points are imaged by the right camera $P'$ at $P'P^+m$ and $P'C = e'$. Thus the epipolar line $l_2$ can be expressed as the cross product,

$$l_2 = (P'C) \times P'P^+m = [e']_x(P'P^+)m$$ \hspace{1cm} (2-40)

Compare (2-38) and (2-40), we can derive that,

$$F = [e']_x(P'P^+)$$ \hspace{1cm} (2-41)

Here are a short summary of the properties of the fundamental matrix $F$:

1) For all correspondent point pairs $m \leftrightarrow m'$, $m'^T Fm = 0$.

2) If $F$ is the fundamental matrix for camera pair $P$ and $P'$, then $F^T$ is the fundamental matrix for camera pair $P'$ and $P$.

3) For any point $m$ in the first image, its corresponding epipolar line on the second image is $l_2 = Fm$. And the corresponding epipolar line for $m'$ in the first image is $l_1 = F^Tm$.

4) The epipole $e$ is the right null space of $F$ and $e'$ is the left null space of $F$. Or $Fe = 0$ and $F^Te = 0$.

5) $F$ is a rank 2 homogeneous matrix with 7 degrees of freedom.

There is a special case when the two cameras are of the same intrinsic parameters and orientation, with only an appropriate translation, i.e., the two camera matrices are $P = K[I|0]$ and $P' = K[I|t]$. In this case, the fundamental matrix is $F = [e']_xKK^{-1} = [e']_x$.

If the translation of the camera is parallel to the x-axis, then $e' = (1,0,0)^T$ and hence the fundamental matrix for this setup is,

$$\bar{F} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$ \hspace{1cm} (2-42)
This is the *canonical form* of the fundamental matrix. In this case, the relation between corresponding points $\mathbf{m}'^T \mathbf{F} \mathbf{m} = 0$ will reduce to a very simple case $\mathbf{m}_y = \mathbf{m}'_y$, i.e., the two corresponding points are on the same scan line. This will make the searching for the corresponding point a 1-D searching problem rather than a 2D searching problem.

The processing on stereo images so that the fundamental matrix will be transformed to the form as in (2-42) is called epipolar rectification. We will discuss it in detail in Chapter 4.

We establish the algebraic framework of stereo vision system in this chapter so that we can handle the problem using a computer. In Chapter 3, we will start working on the wide baseline stereo problem from the perspective of engineering and computing rather than the treatment here as pure mathematics.
3. Wide Baseline Sparse Feature Detection and Matching

As discussed in Chapter 1 and 2, feature detection and matching components is necessary for automatic vision systems. Such important information as camera intrinsic and orientation parameters, or epipolar geometry between two cameras can now be estimated automatically from carefully chosen features. For instance, the method described in [Zha-98b] can generate calibration information from several images of a planar check board, while estimation of fundamental matrix between two cameras can be obtained accurately based on the knowledge of a bunch of feature points matching as shown in [Har-04].

On the other hand, automatic feature detection and matching is an important research problem by itself. Based on Marr’s computational vision model, it is a crucial step to implement the vision mechanism. From the perspective of applications, it is very important in such areas as images registration, 3D reconstruction, motion tracking, robot navigation, object detection and recognition, etc. In recent years, there are many huge image databases emerging on the Internet. Reliable feature detection and matching are crucial techniques to identify interesting personal or objects in images captured from very different viewpoints.

In this chapter, we will first conduct a brief survey on the state-of-the-art of feature detection and description, then we will choose the prevailing SIFT feature detector and descriptor as an example to demonstrate how we can enhance the matching accuracy between the wide baseline features by introducing structural and geometric constraints within the feature point set. Image data sets of different scenes are used to test the performance of the proposed feature matching methods. Comparisons to the state-of-the-art matching method are provided so that the advantage of the proposed method can be validated.
3.1 The State-of-the-art of Wide Baseline Feature Points Detection and Description

In the present literature, there are a variety of feature detectors and descriptors that can be used for the purpose of describing and matching for features. Among them, the majority is points or patches-like features (For instance, see [Bro-05] or [Mat-04]). Some other researchers use edges or lines that are ubiquitous in man-made scenes such as urban buildings (An example in this category is [Sin-08]). In this research we will concentrate on point like features, with necessary information from appropriate neighborhood. A key advantage of point like feature detection is that they permit to carry out correspondence matching even in the presence of clutter, occlusion, large scale, and orientation changes.

The feature point means specific pixel locations in an image, such as mountain peaks or building corners. These kinds of localized features are also called keypoint features or interest points in the literature. In general, keypoint features are often described by the appearance of patches of pixels surrounding their point locations.

Comprehensive descriptions of feature detection and description techniques can be found in a series of survey and evaluation papers by Schmid, Mikolajczyk and their research group covering both feature detection (For detailed information, you are referred to [Sch-02b], [Mik-04], [Tuy-08]) and feature descriptors (See for example, [Mik-05]). Shi and Tomasi and Triggs also provide nice reviews of feature detection techniques (See [Shi -94], [Tri-04]). Another comprehensive survey of interest point detector is [Sch-00]; a detailed performance evaluation based on real world images testing is provided in it.

In general, we can divide the keypoint detection and matching into three sequential stages, as shown in Figure 3-1,

1. feature detection/extraction;
2. feature description;
3. feature matching/tracking.
In the first stage, each image is searched for locations that are likely to match well in other images. In the following feature description stage, each region around detected keypoint locations is converted into a more compact and stable (invariant) descriptor that can be matched against other descriptors generated by the other images. The feature matching stage efficiently searches for likely matching candidates in other images. The feature tracking stage is an alternative to the third stage that only searches a small neighborhood around each detected feature and is therefore more suitable for video processing.

Sparse feature points processing is a crucial step for automatic vision systems. In the 1990’s, three-dimensional geometric reconstruction techniques have been matured while the most successful feature points detection method remains the quite old Harris detector. The Harris detector was introduced by Harris and Stephens in [Har-88]. Although it is proposed more than 20 years ago, the Harris corner detector has been adapted to many new techniques in this area so that it deserves a short discussion here.

Harris corner detector detects corner in the images as two dominant directions of intensity gradient. It is popular due to its strong invariance to rotation, scale, illumination and image noise. In 1996, a comparison of different detectors under varying conditions has shown the Harris detector obtain the most repeatable results.
Given a small window shift \((dx,dy)\), the auto-correlation function \(C_w(dx,dy)\) of a local window \(W\) in the image is defined as,

\[
C_W(dx,dy) = \sum_{(x_i,y_i)\in W} \left( I(x_i + dx, y_i + dy) - I(x_i,y_i) \right)^2
\]  

(3-1)

If we approximate the pixel intensity function in the shifted window with a first-order Taylor expansion, we get

\[
I(x_i + dx, y_i + dy) \approx I(x_i,y_i) + I_x(x_i,y_i)dx + I_y(x_i,y_i)dy
\]  

(3-2)

Thus we have,

\[
C_W(dx,dy) \approx \sum_w [dx,dy] \begin{bmatrix} I_x \\ I_y \end{bmatrix} \begin{bmatrix} I_x \\ I_y \end{bmatrix}^T [dx,dy]
\]

(3-3)

So that \(C_w(dx,dy)\) is a quadratic form determined by a weighted summation matrix. Let \(\lambda_1,\lambda_2\) be the eigenvalues of the matrix. They capture the structure of the local neighborhood by measuring both eigenvalues of this matrix.

Considering a local window in the image and a little shift of this window in one direction we can have three cases:

- If the window is flat (the intensity is almost constant in the window) then any shift will result in a small change of the window.
- If the window is over an edge then a shift along the edge direction will result in a small change while a significant one in the direction perpendicular to the edge.
- If the window is over a corner then shift in any direction will result in a big change of what we see in the window.
The phenomena described above can be detected as a large auto-correlation function as shown in (3-3) and in turn can be detect using eigenvalue analysis. As summarized in Table 3-1, if both of the eigenvalues are big, then the auto-correlation function will be large given any $(dx, dy)$. It can be interpreted as a corner. Otherwise, if only one of the eigenvalues is big, then it should be detected as an edge. If both of the eigenvalues are small, then it seems a uniform area.

**Table 3-1. The detection results of Harris corner detector and the characteristics of the Eigen values of the weighted summation matrix.**

<table>
<thead>
<tr>
<th>Cases</th>
<th>Eigen values</th>
<th>Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>2 large eigenvalues</td>
<td>corner point detected</td>
</tr>
<tr>
<td>b</td>
<td>1 large eigenvalue</td>
<td>edge detected</td>
</tr>
<tr>
<td>c</td>
<td>both eigenvalues small</td>
<td>uniform region</td>
</tr>
</tbody>
</table>

In practice, a circular Gaussian window is often applied before the auto-correlation analysis to make the results less noisy and also rotation invariant. The Harris detector is based on the first derivatives of the image, whereas the Hessian detector is based on the second derivatives and the Hessian matrix. The first use of the Hessian matrix seems to be by Beaudet in 1978 (See [Bea-78] for details). Given a window $W$, we compute the average Hessian matrix $\text{Hessian}_W$,

$$
\text{Hessian}_W = \begin{bmatrix}
\sum_W I_{xx}(x_i, y_i) & \sum_W I_{xy}(x_i, y_i) \\
\sum_W I_{xy}(x_i, y_i) & \sum_W I_{yy}(x_i, y_i)
\end{bmatrix}
$$  \hspace{1cm} (3-4)

The local maxima of the corner strength denote the corners in the image. The determinant is related to the Gaussian curvature of the signal and this measure is invariant to rotation. [Bea-78] proposed to use the local maxima of $\text{det}(\text{Hessian}_W)$ and [Dre-82] proposed to use the zero-crossing of $\text{det}(\text{Hessian}_W)$. An extended version, called Hessian-Laplace detector
as proposed in [Mik-04] detects points that are invariant to rotation and scale change (local maxima of the Laplacian-of-Gaussian). To compute the Hessian, we can use the following 3*3 masks listed in Table 3-2.

Table 3-2. Second order difference operators.

<table>
<thead>
<tr>
<th>Mask</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I_{xx}$</td>
<td>$\frac{1}{3} \begin{bmatrix} 1 &amp; -2 &amp; 1 \ 1 &amp; -2 &amp; 1 \ 1 &amp; -2 &amp; 1 \end{bmatrix}$</td>
</tr>
<tr>
<td>$I_{yy}$</td>
<td>$\frac{1}{3} \begin{bmatrix} 1 &amp; 1 &amp; 1 \ -2 &amp; -2 &amp; -2 \ 1 &amp; 1 &amp; 1 \end{bmatrix}$</td>
</tr>
<tr>
<td>$I_{xy}$</td>
<td>$\frac{1}{4} \begin{bmatrix} 1 &amp; 0 &amp; -1 \ 0 &amp; 0 &amp; 0 \ -1 &amp; 0 &amp; 1 \end{bmatrix}$</td>
</tr>
</tbody>
</table>

Tomasi and Kanade developed a feature tracker in [Tom-91] based on a previous work of [Luc-81]. Defining a good feature as ‘the one that can be tracked well’, a feature is detected if only the two eigenvalues of an image patch are smaller than an empirically computed threshold.

Most of the detectors discussed above are relatively simple and successful applied in the small baseline cases only. As pointed out by Triggs in a tutorial talk in 2003, the automatic stereo systems in the 1990’s are very much like buildings on the sands without a strong basis which is very prone to difference between different image frames.

The last decade sees the progress of the detection of image regions invariant under certain geometric transformations. The methods for detecting scale-invariant regions were presented in [Lin-98], [Kad-04], [Jur-04], [Low-04], [Lei-04]. Generally these techniques assume that the scale change is constant in every direction and search for local extrema in the 3D scale-space representation of an image (x, y and scale). In particular, the DoG (Difference of Gaussian) detector described in [Low-04] shows high repeatability under various tests. It selects blob-like structures by searching for scale-space maxima of a DoG.
Affine-invariant region detector is a generalization of the scale-invariant detector, because under an affine transformation, the scale change can be different in different directions. Therefore shapes deformation is not uniform with respect to affinities. As summarized by the vision group of oxford on a website devoted to affine invariant detectors\(^2\), the most often applicable affine region detectors are:

- The Harris-affine detector: the Harris-Laplace detector is used to determine localization and scale and then the second moment of the intensity gradient determines the neighborhood (See [Mik-04] for details).
- The Hessian-affine detector: key points are detected using the Hessian matrix and the scale is selected based on the Laplacian; the elliptical regions are estimated with the eigenvalues of the second moment of the intensity gradient (See [Mik-04] for details).
- The MSER (Maximally Stable Extremal Region) detector: it extracts regions closed under continuous transformation of the image coordinates and under monotonic transformation of the intensities (See [Mat-02] for details).
- The Salient Regions detector: regions are detected by the entropy of pixel intensity histograms (See [Kad-04] for details).
- The Edge-Based Region detector: regions are extracted combining key points detected by the Harris operator and edges extracted by a Canny operator (See [Tyu-04] for details).
- The Intensity extrema-Based Region detector: affine-invariant regions are extracted by evaluating the image intensity function and its local extrema (See [Tyu-04] for details).

In this research, we will use the SIFT detector and descriptor as the major feature detection method because it have been the most important method and proven to be applicable in our data sets. It has been proven to be the most effective in many cases according to a performance study by Mikolajczyk and Schmid ([Mik-05b]). The problem we intend to

\(^2\) Please check out \text{http://www.robots.ox.ac.uk/~vgg/research/affine/}. The information is retrieved in November 2011.
solve in this research is that all the above feature detectors are local. In many cases, there will be matching problems if repeated patterns exist in the images, which are very common because many objects are more or less symmetric and have repeated patterns, as we will see in the data sets of the following sections.

3.2 The SIFT Feature Detection and Description

In this research, we propose a novel algorithm to match two feature descriptor sets using the intrinsic geometric and structural information within the feature data sets. We will use the SIFT features as the inputs for the matching algorithm. This section will give a detailed description of the detection and description of SIFT features. The following account is based on Lowe’s original paper [Low-04] and sample images are used to illustrate the intermediate results. SIFT detector and descriptor can be summarized as Harris like key points with descriptors of histogram of gradient, thus scale, location and orientation information is integrated in the SIFT key points with its descriptors.

We can divide the whole processing into two parts with seven steps:

1. Key-points Detection
   1.1. Construct Scale Space;
   1.2. Take Difference of Gaussian;
   1.3. Locate DoG Extrema;
   1.4. Sub Pixel Locate Potential Feature Points;
   1.5. Filter Edge and Low Contrast Responses;

2. Descriptor Building
   2.1. Assign Keypoint Orientation;
   2.2. Build Keypoint Descriptor.

The Keypoint Detection

Multi-scale representations of an image were already used in 1970s and the notion of scale-space representation was introduced in [Wit-84], [Koe-84] and later [Lin-93] established
that the only possible scale-space kernel is the Gaussian kernel function, under reasonable assumptions. A Gaussian pyramid is a scale-space representation because it comprises a continuous scale parameter and the spatial sampling is preserved at each scale.

A scale-space of an image is defined as a function $L(x, y, \sigma)$, where $\sigma$ is the scale factor. A Gaussian scale-space for an input image $I$, is defined by

$$L(x, y, \sigma) = G(x, y, \sigma) \ast I(x, y)$$ (3-5)

where $\ast$ represents the convolution operation in both $x$- and $y$- direction and the filter $G(x, y, \sigma)$ is defined as

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$ (3-6)

The scale-space is defined as a continuous function of $\sigma$, but in practice, we need to discretize it. At the same time, to maintain the uniformity of the information change between two levels of resolution the scale factor is designed to be distributed exponentially. Thus only the levels $L_n$ is kept, defined by $L_n(x, y, \sigma), \sigma_n = k^n\sigma_0$. Here, $\sigma_0$ is the initial scale factor and $k$ is the factor of scale change between successive levels.

Laplacian of Gaussians as described in equation (3-5) has been proven to be extremely useful because it is stable computationally and gives important information about the scale of regions of an image. But it is computationally expensive, so in practice, we often use difference of Gaussian as an approximation to it. Actually,

$$\sigma \nabla^2 G(x, y, \sigma) = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$$ (3-7)

So we have the equation,

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma)$$ (3-8)
This processing can be illustrated as in Figure 3-2.

![Diagram of Gaussian Pyramid and DOG detector](image)

Figure 3-2. Gaussian Pyramid and DOG detector (Source:[Low-04]).

The key points are then located at the extreme of the DOG. Practically, we will look for the minimal and maximum point in the neighboring points (including the scale spaces), that is 26 comparisons. To enhance the accuracy of the localization, the zero crossing of the Taylor series expansion for the DOG is estimated.

\[
D(\mathbf{x}) = D_0 + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}
\]  

Or set \( \mathbf{x} \) to the value in (3-10), to get the location and scale.

\[
\hat{\mathbf{x}} = - \left( \frac{\partial^2 D}{\partial \mathbf{x}^2} \right)^{-1} \frac{\partial D}{\partial \mathbf{x}}
\]  

To discard the key points with low contrast, the value of \( D(\hat{\mathbf{x}}) \) can be used. If this value is less than a threshold (empirically 0.03), the candidate key point is discarded. Otherwise it is kept, with final location and scale \( \sigma \), which is the original location of the key point at scale \( \sigma \). The edge points can be detected similar to the case in the Harris corner detector.
Generation of Descriptors

![Image of Keypoint Descriptor Generation](image)

Figure 3-3. Generate Keypoint Descriptor from Image Gradients (Source: [Low-04]).

Each key point can be assigned one or more orientations based on local image gradient directions. This is the key step in achieving invariance to rotation. First, the Gaussian-smoothed image $L(x, y, \sigma)$ at the key point's scale $\sigma$ is taken so that all computations are performed in a scale-invariant manner. For an image sample $L(x, y)$ at scale $\sigma$, the gradient magnitude $m(x, y)$, and orientation, $\theta(x, y)$, are calculated using pixel differences:

$$m(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x + 1, y) - L(x, y - 1)}{L(x + 1, y) - L(x, y - 1)}\right)$$  \hspace{1em} (3-11)

The magnitude and direction calculations for the gradient are done for every pixel in a neighboring region around the key point in the Gaussian-blurred image $L$. And an orientation histogram with 36 bins is formed, with each bin covering 10 degrees. Each sample in the neighboring window added to a histogram bin is weighted by its gradient magnitude and by a Gaussian-weighted circular. The peaks in this histogram correspond to dominant orientations. As shown in Figure 3-3, once the histogram is filled, the orientations corresponding to the highest peak and local peaks that is within 80% of the
highest peaks are assigned to the key point. In the case of multiple orientations being assigned, an additional key point is created having the same location and scale as the original key point for each additional orientation.

As an example, we illustrate the processing of an image of the model house as below. In Figure 3-4, the original image is first down sampling by a double, half, fourth, eighth factor. We call each row an octave. For any octave, the image is then convolved with Gaussian filter with the variance as described as $L_n(x, y, \sigma), \sigma_n = k^n\sigma_0$.

For example, the first row is octave 1, with image size 1535x1151; the five images in this row are convolved with a Gaussian with variance 0.707, 1.000, 1.414, 2.000, 2.828, respectively. The second row is of the original size of the image. The third and fourth row are subsampled with a factor 0.5 and 0.25, respectively.

![Figure 3-4. An example of Gaussian image pyramid of four scales.](image)

In Figure 3-5, the difference of Gaussian pyramid is shown. Then the location of the tentative feature points are determined by finding the local extrema of the DOG, as shown in Figure 3-6. Figure 3-7 shows the key points left after the low contrast points are removed. Furthermore, Figure 3-8 shows the key points left after the edge points are removed. Finally, the key points with dominant gradient orientation and magnitude are shown in Figure 3-9.
Figure 3-5. The Difference of Gaussian (DoG) Pyramid.

Figure 3-6. The locations of DOG extrema detected in the sample image.
Figure 3-7. The key points remain after removing low contrast extrema.

Figure 3-8. Key points remain after removing edge points.
The processing time for different steps is summarized below. As can be seen, most of the computation is devoted to the key point localization and the generation of the descriptor.

Table 3-3. A typical computation time complexity for SIFT feature detection and description.

<table>
<thead>
<tr>
<th>Step</th>
<th>Computation Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocessing</td>
<td>1.31s</td>
</tr>
<tr>
<td>Pyramid</td>
<td>1.71s</td>
</tr>
<tr>
<td>Key points</td>
<td>183.60s</td>
</tr>
<tr>
<td>Gradient</td>
<td>1.50s</td>
</tr>
<tr>
<td>Orientation</td>
<td>3.03s</td>
</tr>
<tr>
<td>Descriptor</td>
<td>72.51s</td>
</tr>
<tr>
<td><strong>Total processing time</strong></td>
<td><strong>263.66 seconds.</strong></td>
</tr>
</tbody>
</table>

3.3 A Feature Matching Algorithm with Geometric and Structural Constraints in Wide Baseline Frames
3.3.1 The Problem with the Existing Feature Matching Methods

Due to its inherent combinatorial complexity and ill-posedness, feature matching or correspondence is one of the hardest low-level image analysis tasks. The problem can be stated as finding pairs of features in two (or more) perspective views of a scene such that each pair corresponds to the same scene point.

To establish correspondence between SIFT features, Lowe used a modification of the K-D Tree algorithm called the Best-Bin-First (BBF) search method that can identify the nearest neighbors with high probability using only a limited amount of computation (See[Low-04] for details). The BBF algorithm uses a modified search ordering for the K-D Tree algorithm so that bins in feature space are searched in the order of their closest distance from the query location. This search order requires the use of a heap-based priority queue for efficient determination of the search order. The best candidate match for each key point is found by identifying its nearest neighbor in the database of key points from training images. The nearest neighbors are defined as the key points with minimum Euclidean distance from the given descriptor vector. The probability that a match is correct can be determined by taking the ratio of distance from the closest neighbor to the distance of the second closest. Lowe rejected all matches in which the distance ratio is greater than 0.8, which eliminates 90% of the false matches while discarding less than 5% of the correct matches. To further improve the efficiency of the best-bin-first algorithm search was cut off after checking the first 200 nearest neighbor candidates.

Although the SIFT descriptors capture scale, orientation and gradient magnitude information and have proven to be very useful in many applications, experiments shows that the local information within the region may not be enough to establish a good matching if there are repeated patterns in the image, which is very common in many urban scene such as buildings. And the symmetric characteristics of natural objects are also prone to mismatching. In practice, we found that there is possibility that the K-D Tree matching algorithm mismatch features with local similarity. This declaration is proved if you check out the experimental results presented in Section 3.4.
3.3.2 The Proposed Algorithm

Considering the shortcomings related to the local properties existed in the present matching algorithms, we proposed a new algorithm that makes use of the information within the pattern of the feature point set. The original idea can be trace back to the method proposed by Scott and Longuet-Higgins in [Sco-91] as to associate features of two arbitrary patterns. The algorithm exploits some properties of the singular value decomposition (SVD) to satisfy both the exclusion and proximity constraints within the data set. The original algorithm works well for narrow baseline stereo matching (See [Pil-97] for an extension of the original idea of Scott and Longuet-Higgins) and we expand it so that it can handle the projective distortion exists in the wide baseline cases. The advantages of the proposed algorithm are,

- Combine the strength of global geometric constraints and local information within the feature data set;
- Avoid mismatching between repeated patterns in the image;
- Enhance the matching results when the geometric distortion is large.

The basic idea of algorithm is explained as follows. If we have any two feature sets,

\[
\{x_i\}, i = 1, ..., m, x_i \text{ are normalized row vector of dimension } d
\]

\[
\{y_i\}, i = 1, ..., n, y_i \text{ are normalized row vector of dimension } d
\]

We can generate two matrices by stacking \(x_i\)'s and \(y_i\)'s respectively,

\[
x = \begin{bmatrix} x_1 \\ \vdots \\ x_m \end{bmatrix}, y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}
\]

Where \(x_i\)'s and \(y_i\)'s can be any feature vectors. Ideally, the matching matrix \(P\) is a permutation matrix which satisfied

\[
P = \text{argmin}_J \| X - JY \|^2_F , \text{ where } J \text{ is a } m \times n \text{ permutation matrix.} \tag{3-12}
\]

And it is easy to show that, (3-12) is equivalent to (3-13),

\[
\text{argmin}_J \| X - JY \|^2_F 
\]
\[ \mathbf{P} = \arg \min_{\mathbf{P}} \| \mathbf{J} - \mathbf{XY}^T \|_F^2 \]  

(3-13) is the well known orthogonal Procrustes problem in the matrix theory. The solution to this problem is \( \mathbf{P} = \mathbf{U}^T \mathbf{V} \), where the SVD of the matrix \( \mathbf{XY}^T = \mathbf{U}^T \mathbf{D} \mathbf{V} \). For a proof of the solution of (3-13), you are referred to [Gol-96]. It is reported that this algorithm works well for image point set undergone with rigid transformation in [Pil-97]. But it fails for large baseline cases because of the existence of large perspective distortion.

To compensate the projective distortion introduced during the imaging, we introduce an affine transformation to modify the geometric localization of the feature points. To facilitate that operation, first we define the feature vectors as,

\[ \mathbf{x}_i = \{(\bar{x}_i, \bar{y}_i); \text{des}_i\} \]  

(3-14)

Here, \((\bar{x}_i, \bar{y}_i)\) is the coordinates of the \(i\)-th key point measured in pixels, which has been normalized. \(\text{des}_i\) is the normalized feature descriptor vector of the \(i\)-th feature points.

Then we can define an affine transformation matrix \(\mathbf{T}\),

\[ \mathbf{T} = \begin{bmatrix} \mathbf{A} & 0 \\ 0 & I_{d-2} \end{bmatrix} \]  

(3-15)

Here \(\mathbf{A}\) is a two dimensional affine matrix and the dimension of \(\mathbf{T}\) is \(d\). If needed, zero rows or columns are padded to \(\mathbf{T}\) to make it compatible with the matrix \(\mathbf{Y}\). Then we can consider the problem (3-16) instead:

\[ \mathbf{P} = \arg \min_{\mathbf{P}} \| \mathbf{J} - \mathbf{XT}^T \mathbf{YT}^T \|_F^2 \]  

(3-16)

If \(\mathbf{T}\) is known, then we can readily solve \(\mathbf{P}\) in (3-16) as a Procrustes problem. And symmetrically, if \(\mathbf{P}\) is known, \(\mathbf{T}\) can be determined by solving a nearest matrix problem.
In our problem, we don’t know $T$ and $P$ at the same time. So we can introduce an iterative strategy to attain solution of the problem. The algorithm can be stated formally as followed,

**Algorithm: SIFT descriptor matching with geometric and structural constraints within the point set**

**Step 0:** Preparing the data set

$$\{x_i\}, i = 1, \ldots, m$$
$$\{y_i\}, i = 1, \ldots, n$$

Where

$$x_i = \{(\bar{x}_i, \bar{y}_i); \text{des}_i\}$$

Here, $(\bar{x}_i, \bar{y}_i)$ is the location of the $i$-th key point, which has been normalized. $\text{des}_i$ is the normalized feature descriptor vector of the $i$-th feature points.

**Step 1:** generate two matrices by stacking $x_i$’s and $y_i$’s respectively.

$$X = \begin{bmatrix} x_1 \\ \vdots \\ x_\sigma \end{bmatrix}, Y = \begin{bmatrix} y_1 \\ \vdots \\ y_\sigma \end{bmatrix}$$

Where $\sigma = \max \{m, n\}$. The smaller matrix is padded with zeros.

**Step 3.** Set an initial guess of the affine transformation $T$,

$$T = \begin{bmatrix} A_0 & 0 \\ 0 & I_{d-2} \end{bmatrix}$$

$A_0$ is set to $2 \times 2$ identity matrix.

**Step 4.** (Iteration Step)

For $k=0,1,2,3$, do the following until $P$ and $T$ converge to certain matrices:

$$P_{k+1} = \arg\min \|J - XT_k^T Y_k^T\|_F^2$$

$$T_{k+1} = \arg\min \|L - Y^T P_k^T X\|_F^2$$

We then obtain the matrix $P$.

Once we obtain the matching matrix $P$, we can consider the problem of how to determine the correspondence of features based on our experiments. Let row $k$ of matrix $P$ be $P_k = (p_{k1}, \ldots, p_{kj}, \ldots, p_{kn})$, and column $j$ of matrix $P$ be $P_j^T = (p_{1j}, \ldots, p_{kj}, \ldots, p_{nj})^T$. Then we deem feature pair $(k,j)$ is a match if (3-17) is satisfied.
\[ |p_{kj}| > \mu_1 \sum_{l \neq j} |p_{kl}| \text{ and } |p_{kj}| > \mu_2 \sum_{m \neq k} |p_{mj}| \]

(3-17)

In this way, \( p_{kj} \) is a dominating element in both row \( k \) and column \( j \). Here we choose the two parameters \( \mu_1 \) and \( \mu_2 \) between 0.7~0.9 by experimental observations.

### 3.4 Experiment Results and Comparisons with K-D Tree Algorithm

To test the performance of the proposed algorithm, we choose several data sets with large base line and in some cases there are plenty of repeated patterns in the images. The data and the types of distortion existed in the data are listed in Table 3-4. The first four data sets are obtained from the oxford wide baseline test data set, which are used frequently by the researchers in this area\(^3\). The face data sets are selected from the IRIS face database.

<table>
<thead>
<tr>
<th>Test data sets</th>
<th>Types of Distortion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model House data set</td>
<td>Large rotation and view angle change</td>
</tr>
<tr>
<td>Library data set</td>
<td>Repeated Patterns and symmetry in shape</td>
</tr>
<tr>
<td>Valbonne Church data set</td>
<td>Large scale change with illumination change, occlusions</td>
</tr>
<tr>
<td>Cottage data Set</td>
<td>Large view angle change, natural scene background</td>
</tr>
<tr>
<td>Male Face data Set</td>
<td>Symmetric face data</td>
</tr>
<tr>
<td>Female Face data set with change facial expression</td>
<td>Nonlinear surface change, view angle change</td>
</tr>
</tbody>
</table>

We also compare our matching results with the results of the matching algorithm described by Lowe, which is used by many people for matching purpose. In [Low-04], Lowe used a

\(^3\) Data sets retrieved from http://www.robots.ox.ac.uk/~vgg/data/data-mview.html in January 2011.
modification of the K-D Tree algorithm that can identify the nearest neighbors with higher probability.

In Figure 3-10, two image frames extract from a video of a model house are shown. We notice that the view angles of the camera are changed so that the two views look quite different. SIFT detects 3630 features in the first image frame and 4024 in the second image frame.

In Figure 3-11, the corresponding points found by K-D Tree method are linked using line segment. It is easy to find there are about 10 bad matches out of the total 422 matches.

Figure 3-12 presents the matching result of the proposed method. There is no obvious mismatch in the image.

In Figure 3-13 are two images of a building with repeated windows and symmetric patterns (such as the windows) as well. SIFT detects 2739 features in the first frame and 2587 in the right frame.

Figure 3-14 is the matching result generated using the K-D Tree method. We found that there are at least six bad matches. In Figure 3-15, the matching of the proposed method finds 101 matches with 3 bad matches. As anticipated, the bad matches are nearer geometrically than the result of K-D Tree.

In Figure 3-16, two wide baseline images of the Valbonne Church are shown. Because of the wide view angle change and zoom changes, the matching is not an easy task. SIFT detects 3398 features in the first frame and 4205 in the second.

In Figure 3-17 we can find there are 417 matches with several obvious bad matches obtained by the K-D Tree algorithm. While in Figure 3-18, we have 209 matches; almost all of them are good. Although the proposed method detects fewer matches but the quality
is more reliable. We also notice that most of the matches lie on the wall, with similar view angle.

In Figure 3-19, two images of a cottage in the forest is shown, as can be noticed, the two images are taken from very different view angles. SIFT detects 2207 feature points in the first frame and 2082 in the second. As can be seen in Figure 3-20, these two frames are taken from such different view angles that SIFT K-D Tree method only detects 10 matches. What makes things worse is that there are repeated patterns such as the posts and windows. Only two of the matches are good. In contrast to the performance of the comparison method, the proposed method detects 54 matches and most of them are good, as shown in Figure 3-21.

In real applications, it is highly desirable to develop some robust feature matching methods so that the face can be matched even with large view angle change. Iris lab has accumulated huge database of human faces. We just choose two data sets from the database to test the performance of the proposed method.

Shown in Figure 3-22 are two frames of a male face. We want to test the performance of the face images and the possibility of applying this technique for three dimensional face reconstructions. SIFT detects 3630 features in the first image and 4024 in the second image. As shown in Figure 3-23, the K-D Tree method detects 87 matches and 2 of them are wrong. Our proposed method detects 92 matches and almost all of them are good matches (See Figure 3-24).

Shown in Figure 3-25 are two images of a female face with large facial expression change. We use this data to test the reliability of the features and matches with respect to non-perspective distortion. As shown in Figure 3-26 and 3-27, the result of K-D Tree method is 14 good matches and our proposed method detects 33 good matches.
Figure 3-10. Two frames from the model house image sequence with wide baseline.
Figure 3-11. The matches detected by the K-D Tree method.
Figure 3-12. The matches detected by the proposed method.
Figure 3-13. Two frames from the ‘library’ image sequence.
Figure 3-14. The matches detected by the K-D Tree method.
Figure 3-15. The matches detected by the proposed method.
Figure 3-16. Two frames from the Valbonne Church sequence.
Figure 3-17. The matches detected by the K-D Tree method.
Figure 3-18. The matches detected by the proposed method.
Figure 3-19. Two frames from the Cottage sequence.
Figure 3-20. The matches detected by the K-D tree method.
Figure 3-21. The matches detected by the proposed method.
Figure 3-22. Two frames of the male student face sequence.
Figure 3-23. The matches detected by the K-D Treemethod.
Figure 3.24. The matches detected by the proposed method.
Figure 3-25. Two images of a female face with changed expression.
Figure 3-26. The matches detected by the K-D Tree method.
Figure 3-27. The matches detected by the proposed method.
Finally, the experimental results are summarized in Figure 3-28, 3-29 and 3-30. In Figure 3-28, we compare the total matches obtained by the two methods. We see that in the model house and the Valbonne church cases, K-D Tree methods detects more matches than the proposed method while in other cases, the results of the proposed method is better than the methods.

In Figure 3-29, we compare the numbers of good matches detected by the two methods. Only in the model house and the Valbonne Church cases the K-D Tree method detects more good matches than the proposed method, while in all the other cases, the proposed method beat the K-D Tree method.

Comparison of the good match rate detected for the data sets by the K-D Tree method and the proposed method is illustrated in Figure 3-30, in all the cases the results of the proposed method are better than the K-D Tree method.
Figure 3-29. Comparison of the good matches’ number detected for the data sets by the K-D Tree method and the proposed method.

Figure 3-30. Comparison of the good match rate detected for the data sets by the K-D Tree method and the proposed method.
4. Rectification of Wide Baseline Images

Stereo is an important approach to 3D scene reconstruction in various computer vision applications. One of the most difficult problems in stereo analysis is dense matching. If the epipolar lines in the two images are aligned with scan lines, the complexity of dense matching can be reduced substantially to 1D search.

In the literature, there exist plenty of narrow baseline stereo dense matching algorithms that work well if this assumption is satisfied (See [Sch-02b] and the related website for a thorough survey of dense matching methods). However, in modern computer vision and robotic applications, this assumption is often violated given the dynamic nature of the imaging systems employed, especially in the wide baseline cases. Epipolar rectification is the pre-processing before dense matching, where the conjugate epipolar lines are aligned and become collinear and parallel with the x-axis. As an example, Figure 4-1 (a) and (b) shows a pair of unrectified images, in which corresponding points are not on the same scan lines. After rectification, the epipolar lines are aligned such that all the corresponding points are of the same vertical coordinates as can be seen in Figure 4-1(c) and (d).

Epipolar rectification relies heavily on the concepts of epipolar geometry and the estimation of fundamental matrix; we first introduce the estimation method of the fundamental matrix that we are using in this research in Section 4.1.

A review of the related work in the literature is stated in Section 4.2. An important open problem with image rectification is how to deal with the wide baseline cases, where the viewpoints and the focal lengths may change substantially during image acquisition. Almost all the efforts in this chapter are devoted to this problem. Based on an analysis of previous methods, we present a novel approach that utilizes the intrinsic geometry within each image of the stereo pair. The intrinsic geometries of the two rectified views are embodied in triangulation nets of feature points. By maximizing the structural congruency of the nets, similarity between the rectified images is attained. The intricacies of the
proposed method will be detailed in Section 4.3. Then the method is applied to stereo pairs collected with wide baselines. Various test images and quantitative evaluations are presented in Section 4.5.

Figure 4-1. An example of image rectification. On the top row, two input images are not rectified. On the bottom row, the two images are rectified using the proposed method.
(Source of the input images: http://cvlab.epfl.ch/~tola/daisy.html)

4.1 The Estimation of Fundamental Matrix

In the last two decades, many methods to estimate the fundamental matrix have been proposed and can be classified into two major categories: linear methods and iterative methods (See [Har-04] and [Sze-11] for details).
The linear methods are closed formed but the accuracy may be rather poor in the presence of noise. Based on the discussion in Section 2.6, we know that the fundamental matrix is defined by the equation,

\[ \mathbf{m}'^T \mathbf{F} \mathbf{m} = 0 \quad (4-1) \]

For any pair of matching points \( \mathbf{m} \leftrightarrow \mathbf{m}' \). If we have matching point pairs \( \{ \mathbf{m}_i = (x_i, y_i, 1) \leftrightarrow \mathbf{m}'_i = (x'_i, y'_i, 1), i = 1, ..., n \} \) and assume that \( \mathbf{F} = (f_{ij})_{3 \times 3} \), then we can construct an over-determined linear system of the form

\[
\begin{pmatrix}
    x'_1 x_1 & x'_1 y_1 & x'_1 & y'_1 & x_1 & y_1 & 1 \\
    \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
    x'_n x_n & x'_n y_n & x'_n & y'_n & x_n & y_n & 1
\end{pmatrix}
\mathbf{f} = \mathbf{0}
\]

(4-2)

Because \( \mathbf{F} \) has a degree of freedom 7, if we have \( n \geq 7 \) pairs of corresponding points, we can generate an estimation of the fundamental matrix \( \mathbf{F} \).

In practices, the coordinates of the feature points are not exact because of the measure and computing error, the rank of matrix \( \mathbf{A} \) will be full as 9. \( \mathbf{F} \) is defined up to a scaling, so we can reduce the problem in (4-2) as equivalent to minimize \( \| \mathbf{A} \mathbf{f} \| \) subject to the condition \( \| \mathbf{f} \| = 1 \). This problem has a closed form solution as the singular vector corresponding to the smallest singular value of \( \mathbf{A} \), i.e., the last column of the matrix \( \mathbf{V} \) in the SVD decomposition of \( \mathbf{A} = \mathbf{U} \mathbf{D} \mathbf{V}^T \).

This straightforward linear solution generally does not satisfy the rank 2 constraint, to make the rank of the fundamental matrix be 2, we can replace the \( \mathbf{F} \) obtained above by the matrix \( \mathbf{F}' \) that minimizes the Frobenius norm \( \| \mathbf{F} - \mathbf{F}' \|_F \) subject to the condition that \( \det(\mathbf{F}') = 0 \). This matrix can be find by modify the SVD of \( \mathbf{F} \). Let \( \mathbf{F} = \mathbf{U} \mathbf{D} \mathbf{V}^T \), where \( \mathbf{D} = \text{diag}(r, s, t) \), \( r \geq s \geq t \). Then \( \mathbf{F}' = \mathbf{U} \text{diag}(r, s, 0) \mathbf{V}^T \) will be the matrix we deduce. This algorithm will be applicable when more than 8 feature points’ pairs are available, which is generally satisfied in practice.
According to Hartley [Har-97], this algorithm can perform extremely well if appropriate normalization if the input data is conducted. The suggested normalization is a translation and scaling of each image so that the centroid of the feature points set is send to the origin of the coordinate and the RMS distance of the points from the origin is equal to $\sqrt{2}$. Given $n > 8$ image point correspondences $\{m_i = (x_i, y_i, 1) \leftrightarrow m'_i = (x'_i, y'_i, 1), i = 1, \ldots, n\}$, the normalized 8-point algorithm to estimate the fundamental matrix $F$, such that $m'^T_iFm_i = 0$, is as followed,

<table>
<thead>
<tr>
<th>Normalized 8-point algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i) Normalization: Transform the image coordinates according to $\hat{m}_i = Tm_i$ and $\hat{m}'_i = T'm'_i$, where $T$ and $T'$ are normalizing transformations.</td>
</tr>
<tr>
<td>(ii) Find the fundamental matrix $\tilde{F}'$ corresponding to the matches $\hat{m}_i \leftrightarrow \hat{m}'_i$ by</td>
</tr>
<tr>
<td>a) Linear solution: Determine $\hat{F}$ from the singular vector corresponding to the smallest singular value of $A$, where $A$ is composed from the normalized matches as in (4-2).</td>
</tr>
<tr>
<td>b) Constraint enforcement: Replace $\hat{F}$ by $\hat{F}'$ such that $\det(\hat{F}') = 0$ using the SVD, as explained before.</td>
</tr>
<tr>
<td>(iii) Denormalization: Finally, $F = T'T\hat{F}' T$ is the fundamental matrix corresponding to the original data set.</td>
</tr>
</tbody>
</table>

In practice, we can also introduce robust statistics strategy to iteratively enhance both the matching results and the estimation of fundamental matrix. Three robust methods: M-Estimators, Least-Median-Squares (LMedS) and Random Sampling (RANSAC) are widely used in practice (See [Har-04] for details). The M-estimators try to reduce the effect of outliers by weighting the residual of each point. The LMedS and RANSAC techniques are very similar. The RANSAC calculates the number of inliers for each $F$ and the chosen $F$ is the one that maximizes it. Once the outliers are eliminated, $F$ is recalculated with the aim of obtaining a better approach.
In this research, we will use automatic fundamental matrix estimation algorithm with RANSAC. The outline of the algorithm is as follow:

**Automatic fundamental matrix estimation algorithm using RANSAC**

(i) Feature points Detection: Compute feature points in each image.
(ii) Putative correspondences: Compute a set of feature point matches.
(iii) RANSAC robust estimation: Repeat for $N$ samples,
   (a) Select a random sample of 8 correspondences and compute the fundamental matrix $F$ as described in the normalized 8 points algorithm.
   (b) Calculate the distance $d_{\pm}$ for each putative correspondence.
   (c) Compute the number of inliers consistent with $F$ by the number of correspondences for which $d_{\pm} < t$ pixels.
   (d) If there are three real solutions for $F$ the number of inliers is computed for each solution, and the solution with most inliers retained. Choose the $F$ with the largest number of inliers.
(iv) Non-linear estimation: re-estimate $F$ from all correspondences classified as inliers by minimizing a cost function.
(v) Guided matching: Further feature point correspondences are now determined using the estimated $F$ to define a search strip about the epipolar line.

### 4.2 Related Work

Epipolar rectification is a necessary step in stereovision analysis. The aligning of epipolar lines allows subsequent dense matching algorithms to search along scan lines, rather than in a two dimensional projective space. In uncalibrated cases, the depth information can be reconstructed up to a projective transformation (See [Har-04] for more information). Once the calibration information is available, the Euclidean reconstruction of the scene is possible. On the other hand, many applications require relative depth measures, such as view morphing described in [Sei-97] and robotic navigation as in [Fau-01].
Figure 4-2. General set up for a stereo system and the illustration of epipolar rectification.

Figure 4-3. Rectified Images.

The literature in this research area can be divided primarily into two categories, namely Euclidian epipolar rectification ([Fau-93], [Fus-00]) and projective epipolar rectification. Once the fundamental matrix is estimated, the stereo system is termed weakly calibrated. In this research, only the projective rectification problem is considered, where the camera
parameters are not known explicitly.

One of the seminal works of projective rectification was due to Hartley ([Har-99a], [Har-04]). When applying homographies, severe distortions could be introduced if the parameters are not selected properly. To minimize projective distortions, various constraints can be included. This is the major concern of researchers in most of the previous work. Loop and Zhang consider a stratified decomposition of the rectification homographies in [Loo-99] while Gluckman and Nayar investigate the local resampling effect in [Glu-01]. More recently, Mallon explores the local behavior of the Jacobian matrix of the transformations he introduced in [Mal-05]. The efforts of Wu and Yu described in [Wu-05] are useful in preserving orthogonality and aspect ratio of the geometric elements of the objects in the image.

In the category of Euclidean epipolar rectification, [Fus-00] provides an efficient rectification method for calibrated cameras and the solution is in closed form and can be applied in the cases that the cameras’ internal and external parameters are known beforehand. In [Fus-08], they extend the original method by introducing quasi-Euclidean constraints thus less information is needed.

While minimizing the projective distortions can be beneficial in many narrow baseline cases, it may not be suitable for wide-baseline cases. For instance, the size and shape of the Facade is quite different in the two images shown in Figure 4-1 (a) and (b), due to changing viewpoints and the perspective distortion. There is no point maintaining the shape difference in the rectified images. Indeed, the outputs of the rectification, which are the outputs to the dense matching algorithms, should be made as similar as possible to each other in shape.

This inspires us to propose a new approach that maximizes the structural congruency of resultant images rather than minimize the distortion led by the homographies. We believe that favorable deformations can be introduced for easier dense matching in wide-baseline cases. The proposed method is explained in detail in Section 4.3.
4.3 Uncalibrated Image Rectification using structural congruency

In this section, we will explain the basic idea of the proposed rectification method and state it formally.

4.3.1 Basic Ideas

The basic idea is illustrated in Figure 4-4 and Figure 4-5. Let’s say we take two images of a triangle. Because of the wide baseline properties of the imaging system, we may get two images like what are shown in Figure 4-4(a) and (b). Notice that the bottom side of the triangle may of very different look in the two views because of the perspective distortion.

Figure 4-4. Image rectification with distortion minimized. By applying homographies H and H' to (a) and (b) respectively, we obtain image (a') and (b'). The corresponding pixels are on the same scanline after the operations but it is still very difficult to establish pixel to pixel dense matching for the two triangle shapes in the rectified images.
The traditional method of rectification will minimize the local distortion introduced by the rectified homographies, so that a possible result may be look like in Figure 4-4(a') and (b'). The two images are rectified with no doubt. But it is impossible for us to get a fine dense matching for the two images because the sizes of the triangles in the two images are quite different. So we can try applying an affine transformation after the two images are rectified, as shown in Figure 4-5.

Figure 4-5. Using proposed method, we apply one more affine transformation A on intermediate result (a') to maximize the shape congruency of the scenes and objects in the two frames. In this way, the task of dense matching is made easier.
We use affine transformation here because it will change the x-coordinate of the rectified image and thus will not introduce severe distortions that are not welcome here. Using optimization method, we can find the appropriate affine transformation so that the transformed triangle could be look like the other one as much as possible.

### 4.3.2 The Delaunay Triangulation

Based on the previous analysis, we naturally think about using the feature points detected in the input images to generate a network of control triangles. And then we can consider the net shape difference for the triangle network. Luckily, there is a readily tool to generate triangle network, i.e., the Delaunay triangulation. This is a well-developed research topic in computational geometry; for the details of how to construct a triangle network from a point set, you are referred to [Ber-08]. For a sample Delaunay triangle network, see Figure 4-6. All the triangles are identified by a list of its vertex. Such as in Figure 4-6, the triangles identified by yellow deltas are (62,8,21), (87,20,62) respectively.

![Figure 4-6. The Delaunay triangle network generated using the feature points detected on an image of a silo.](image)
4.3.3 The Proposed Algorithm for Wide Baseline Stereo Rectification

Based on the discussion in the previous section, it is clear that epipolar rectification and distortion compensation are needed before dense matching can be applied on wide baseline stereo image pairs. Epipolar rectification can simplify the dense matching task by reducing 2D searching to 1D searching while distortion compensation compensates for the perspective distortion and made the two views look similar in shape and pose. We define wide baseline image rectification as “epipolar rectification with shape difference minimization”. Now we propose our new algorithm that combines the epipolar rectification and shape distortion compensation. The epipolar geometry of a wide baseline stereo system can be illustrated as in Figure 4-7.

To rectify the epipolar lines, two homographies $H$ and $H'$ can be applied on $I_1$ and $I_2$, respectively. To make the conjugate epipolar lines ($\overline{me}$ and $\overline{m'e'}$ in Figure 4-7) parallel with x-axis and collinear, the equalities $He = (1,0,0)^T$ and $H'e' = (1,0,0)^T$ are necessary. The fundamental matrix for the rectified stereo pair is given by $\overline{F} = [H'e']_x = [(1 0 0)^T]_x$, which is a 3 x 3 skew symmetric matrix. A major constraint on the two rectifying homographies is given by,

$$H^T\overline{F}H = 0.$$  \hspace*{1cm} \text{(4-3)}
, where \( \mathbf{F} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \).

This equation imposes no constraint on the first row of \( \mathbf{H} \) and \( \mathbf{H}' \), which allows certain degrees of freedom of choosing \( \mathbf{H} \) and \( \mathbf{H}' \) to attain more objectives.

The outline of the proposed rectification method with maximized structural congruency is as follows:

**Proposed Algorithm for Wide Baseline Stereo Rectification**

1. Detect and establish the matches between two \( n \)-feature point sets, \( \{ \mathbf{m}_i \} \leftrightarrow \{ \mathbf{m}'_i \} \), \( i = 1, 2, \ldots, n \). Here \( \{ \mathbf{m}_i \} \subseteq I_1 \) and \( \{ \mathbf{m}'_i \} \subseteq I_2 \).
2. Estimate the fundamental matrix \( \mathbf{F} \) using the matching point sets. Solve for the epipoles \( e \) and \( e' \).
3. Apply a homography \( \mathbf{H}' \) on \( I_2 \) and get the image \( \overline{I}_2 \), such that the epipole \( e' \) is sent to infinity.
4. Generate the 2D triangulation net \( N_2 \) for \( \overline{I}_2 \) using the vertices \( \{ \mathbf{H}' \mathbf{m}'_i \}, i = 1, 2, \ldots, n \), and construct a lookup table of triangles of the net.
5. Apply a compatible homography \( \mathbf{H}_0 \) on \( I_1 \) such that \( \mathbf{H}_0 \mathbf{e} = (1,0,0)^T \).
6. Apply an optimized affine transformation \( \mathbf{A} \) on \( \mathbf{H}_0 I_1 \) to maximize the structural congruency between the triangulation nets.

Essentially, image \( I_2 \) is transformed by some quasi-rigid transformation to \( \overline{I}_2 \). The inherent structure of \( \overline{I}_2 \) is represented by the triangulation net \( N_2 \), with the feature points as its vertices. \( N_2 \) is then used as a reference of the rectified structure. Finally, the homography applied on \( I_1 \) is optimized to drive the structural congruency between the rectified images \( \overline{I}_1 \) and \( \overline{I}_2 \).
Typically, we can choose $\mathbf{H}'$ and $\mathbf{H}_0$ such that Equation (4-3) is satisfied. As the first step of our exploration, we adopt the choices described in [Har-99] and [Har-03] in our experimentation.

$\mathbf{H}'$ is the resultant of three sequential operations. Initially, the origin of the image coordinate system is sent to the center of $I_2$ by a translation matrix $\mathbf{T}$. After that, a rotation $\mathbf{R}_\varphi$ can be applied to relocate the epipole $\mathbf{e}' = (e'_x, e'_y, 1)^T$ onto the x-axis with a rotation angle $\varphi = -\tan^{-1}(e'_y / e'_x)$. Finally, a quasi-rigid projective transformation $\mathbf{G}$ is applied to send $\mathbf{e}' = (\tilde{e}'_x, 0, 1)^T$ to $(1,0,0)^T$. Or equivalently we can state that $\mathbf{H}' = \mathbf{GR}_\varphi \mathbf{T}$, where

$$
\mathbf{G} = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
-1/\tilde{e}'_x & 0 & 1
\end{bmatrix}
$$

(4-4)

Once $\mathbf{H}'$ is determined, $\mathbf{H}_0$ can be chosen according to (4-3) as $\mathbf{H}_0 = \mathbf{H}'([\mathbf{e}']_x \mathbf{F} + \mathbf{e}' \mathbf{v}^T)$, where $[\mathbf{e}']_x$ is the skew symmetric matrix such that $[\mathbf{e}']_x \mathbf{e}' = \mathbf{0}$, $\mathbf{v}$ is an arbitrary 3-vector.

The crux of the proposed method lies in maximizing the structural congruency between the rectified images. To achieve this, we first need to characterize the intrinsic geometry of a view. To obtain a reliable representation, 2D Delaunay Triangulation ($DT$) is adopted. $DT$ is considered the de facto standard for triangulation in computer geometry. Computationally, a $DT$ net is represented by a lookup table. Each entry in the table is a vector containing the index of the three vertices of a triangle. As described earlier, we first apply $\mathbf{H}'$ on image $I_2$ and obtain rectified image $\tilde{I}_2$. Then we generate the $DT$ net $N_2$ using the feature points in $\tilde{I}_2$. The structure of $N_1$ is controlled by an affine transformation $\mathbf{A}$ such that it is congruent to $N_2$ as much as possible. Now we establish the cost function which minimize the shape difference between $N_1$ and $N_2$, which is equivalent to maximizing the structural congruency. To illustrate the idea, let's take one sample triangle $A'B'C'$ in $N_2$ and its corresponding triangle $ABC$ in $N_1$ as examples, as shown in Figure 4-8.
Figure 4-8. One sample triangle \( A'B'C' \) in net \( N_2 \) and its corresponding triangle \( ABC \) in net \( N_1 \).

For this sample triangle pair, the structural difference \( d \) between \( ABC \) and \( A'B'C' \) can be defined as,

\[
d = |(X_A - X_B) - (X_{A'} - X_{B'})|^2 + |(X_C - X_B) - (X_{C'} - X_{B'})|^2
\]  

(4-5)

Here \( X_{(.)} \) are the horizontal coordinates of the vertices of the triangles. Considering all the triangles in the net, the total structural difference is defined as

\[
\Sigma_d = \sum_i \left| \left( X_A^{(i)} - X_B^{(i)} \right) - \left( X_{A'}^{(i)} - X_{B'}^{(i)} \right) \right|^2 + \left| \left( X_C^{(i)} - X_B^{(i)} \right) - \left( X_{C'}^{(i)} - X_{B'}^{(i)} \right) \right|^2
\]

(4-6)

Here \( i \) is the index of the triangles in the lookup table. An example of comparison of the shape of two triangle networks is illustrated in Figure 4-9. It is easy to infer from the figure that, although the two triangle nets are of the same topology, the corresponding meshes may differ in size.
Minimizing (4-6) is a linear least squares problem and a closed form solution can be obtained. Notice that the affine transformation $A$ is of the form

$$A = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Parameters $a, b$ in (4-7) is used to control the shape of $N_I$, while parameter $c$ does not affect the shape. Subsequently, $c$ can be chosen to center the rectified image at an appropriate position to get the best view portion.

### 4.4 Experimental Results and Discussions

The proposed method is devised to handle wide-baseline datasets. To evaluate the performance of the proposed method, we compare our methods to Hartley’s approach described in [Har-04], the most applicable image rectification method in the literature.

To rectify the data sets, reliable feature points are initially detected using the SIFT key points detection algorithm described in [Low-04] and then the descriptor vectors of the key points are matched using the proposed matching method described in Chapter 3, which can
guarantee the accuracy of the matching. The fundamental matrix can then be estimated using the normalized 8-point algorithm together with the RANSAC method, as described in Section 4.1. The inliers of the RANSAC are then used to generate the Delaunay triangle nets.

Four data sets are used to evaluate the proposed methods:

- Model house;
- Silo;
- The 'Façade' image sequence (four image pairs with changing baselines);
- The 'Fountain' image sequence (four image pairs with changing baseline).

The Façade and Fountain image sequences are used to evaluate the performance of the proposed method with respect to the changing baselines.

In Figure 4-10, two images of a model house extracted from a video sequence are shown. The original images with triangle nets are shown in Figure 4-11. In Figure 4-12 (a) and (b), the results generated by Hartley's method are shown. We notice that the corresponding points are on same scan lines but the rectified images look dissimilar to each other. For example, the two roof edges are of very different lengths. In Figures 4-12 (c) and (d), the results of the proposed method are shown. The optimized Delaunay triangulation nets are shown overlaid on the outputs in Figures 4-12(e) and (f). As anticipated, the corresponding points are collinear. In addition, we notice that the outputs appear to have been acquired with narrow baseline due to the maximization of the structural congruency between the rectified pair.

---

4 The model house image pair is obtained from [http://www.robots.ox.ac.uk/~vgg/data/](http://www.robots.ox.ac.uk/~vgg/data/). The Silo image pair are from the IRIS surveillance image database. The Façade image sequence and the Fountain image sequence are obtained from [http://cvc.epfl.ch/~tola/daisy.html](http://cvc.epfl.ch/~tola/daisy.html).
Figure 4-10. Two wide baseline frames of a model house from a video sequence. (Source: http://www.robots.ox.ac.uk/~vgg/data/data-mview.html)

Figure 4-11. Two images with triangle nets.
Figure 4-12. (a)(b), the rectification results by Hartley’s method. (c),(d), the results of proposed methods. (e) and(f), the optimized and reference triangulation nets.
The second data set we use to test the performance of the proposed method is the silo data set from the IRIS surveillance image database, as shown in Figure 4-13 (a) and (b).

Figure 4-13. Rectification of wide-baseline stereo, where (a) and (b) are input images. (c) and (d) are results of Hartley’s method. (e) and (f) are results of the proposed methods.

Notice the size difference in the two images shown in Figure 4-13(a) and (b). The regions of interest bounded by yellow boxes are images of the same part on the silo. The size is very different so that a direct dense matching is impossible. Figure 4-13 (c) and (d)
presents the rectification results of Hartley’s method. Figure 4-13 (e) and (f) presents the rectification results of the proposed method with the triangle nets illustrated as blue meshes on the images. It is easy to find that the rectified images look similar to each other.

To evaluate the performance of methods quantitatively, we generate a comparison of the shape differences defined by Equation (4-6). The results are listed in Table 4-1. The numbers listed in the table are the square roots of the $\sum d$ defined in (4-6) for each triangle net, so the unit of the numbers is pixels.

<table>
<thead>
<tr>
<th>Model House</th>
<th>Silo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Images</td>
<td>319.6</td>
</tr>
<tr>
<td>Hartley’s Results</td>
<td>536.1</td>
</tr>
<tr>
<td>Our Results</td>
<td>282.4</td>
</tr>
</tbody>
</table>

To evaluate the performance of the proposed methods with respect to the change of baseline, we use a five-image Façade data set with changing baselines, as shown in Figure 4-14. Image pairs 1-2, 1-3, 1-4, 1-5 are used for the purpose of test.

Figure 4-14. Five images of a façade taken by cameras with changing viewpoints. (Data obtained from http://cvlab.epfl.ch/~tola/daisy.html)

Figure 4-15 (a) and (b) are image 1 and 2 from the 'Façade' image sequence. We notice that corresponding points are not on the same scan line and there is severe perspective
distortion between the two views. In the same figure, (c) and (d) are the images with Delaunay triangle nets generated from the control points detected by SIFT and matched using the proposed method in Chapter 3. It can be noticed that the two triangle nets are of same topology but the size of the corresponding meshes in the two images maybe different from each other. Figure 4-16 (a),(b) are the rectification results of Hartley’s method, with rectified triangle nets. While (c) and (d) are the rectification results by the proposed method, with rectified triangle nets also. (e) and (f) are results of the proposed method without triangle nets. It is easy to check that the corresponding points are on the same scan line and the shape of objects in the rectified images are similar.

Figure 4-15. (a) and (b) are frame 1 and frame 2 of the ‘Façade’ sequence. (c),(d) are frame 1 and 2 with triangle nets.
Figure 4-16. (a),(b) the rectification results by Hartley’s method, with rectified triangle nets. (c) and (d) the rectification results by proposed method, with rectified triangle nets. (e) and (f) results of the proposed method.
Figure 4-17 (a) and (b) are image 1 and 3 from the 'Façade' image sequence. We there is even severe perspective distortion between the two views than in the previous data set. In the same figure, (c) and (d) are the images with Delaunay triangle nets generated from the control points detected by SIFT and matched using the proposed method. The two triangle nets are of same topology but the size of the corresponding meshes in the two images are different from each other. Figure 4-18 (a), (b) are the rectification results of Hartley’s method, with rectified triangle nets. While (c) and (d) are the rectification results of the proposed method, with rectified triangle nets also. (e) and (f) are results of the proposed method without triangle nets. It is easy to check that the corresponding points are on the same scan line now and the shape of objects in the rectified images is similar.

Figure 4-17. (a) and (b) are frame 1 and frame 3 of the ‘Façade’ sequence. (c),(d) are frame 1 and 3 with triangle nets.
Figure 4-18. (a),(b) are the rectification results by Hartley’s method, with rectified triangle nets. (c) and (d) the rectification by proposed method, with rectified triangle nets. (e) and (f) results of the proposed method.
Figure 4-19 (a) and (b) are image 1 and 4 from the 'Façade' image sequence. There is more severe perspective distortion between the two views than image 1 and 3. In the same figure, (c) and (d) are the images with triangle nets generated from the control points detected by SIFT and matched using the proposed method. It can be noticed that the triangle nets are of same topology but the size of the corresponding meshes in the two images are different from each other. Figure 4-20 (a),(b) are the rectification results by Hartley’s method, with rectified triangle nets. While (c) and (d) are the rectification results by the proposed method, with rectified triangle nets also. (e) and (f) are results of the proposed method without triangle nets. It is easy to check that the corresponding points are on the same scan line now and the shape of objects in the rectified images is similar.

Figure 4-19. (a) and (b) frame 1 and frame 4. (c),(d) frame 1 and 4 with triangle nets.
Figure 4-20. (a),(b) are the rectification results by Hartley’s method, with rectified triangle nets. (c) and (d) are the rectification results by proposed method, with rectified triangle nets. (e) and (f) are results of the proposed method.
Figure 4-21 (a) and (b) are image 1 and 5 from the 'Façade' image sequence. The viewpoints change is more severe between the two views than image 1 and 4. In the same figure, (c) and (d) are the images with triangle nets generated from the control points detected by SIFT and matched using the proposed method. It can be noticed that the triangle nets are of same topology but the size of the corresponding meshes in the two images are different from each other. Figure 4-22 (a),(b) are the rectification results by Hartley’s method, with rectified triangle nets. While (c) and (d) are the rectification results by the proposed method, with rectified triangle nets also. (e) and (f) are results of the proposed method without triangle nets. It is easy to check that the corresponding points are on the same scan line now.

Figure 4-21. (a) and (b) frame 1 and frame 5. (c),(d) frame 1 and 5 with triangle nets.
Figure 4-22. (a),(b) the rectification results by Hartley’s method, with rectified triangle nets. (c) and (d) are the rectification results by the proposed method, with rectified triangle nets. (e) and (f) are results of the proposed method.
The second image sequence used to evaluate the performance of the proposed methods with respect to the change of baseline is the five-image 'Fountain' data set with changing baseline, as shown in Figure 4-23. Image pairs 1-2, 1-3, 1-4, 1-5 are used to test performance of the proposed algorithm.

Figure 4-23. Five images of a fountain taken by cameras with changing viewpoints. (Data obtained from http://cvlab.epfl.ch/~tola/daisy.html)
Figure 4-24 (a) and (b) are image 1 and 2 from the 'Fountain' image sequence. The original images are not rectified and there is pose difference between the two views to some degree. In the same figure, (c) and (d) are the images with triangle nets generated from the control points detected by SIFT and matched using the proposed method. The triangle nets are of same topology but the size of the corresponding meshes in the two images may different from each other. In Figure 4-25, (a),(b) are the rectification results of Hartley’s method, with rectified triangle nets. While (c) and (d) are the rectification results of the proposed method, with rectified triangle nets also. (e) and (f) are results of the proposed method without triangle nets. Please notice the effects of pose adjustment in the rectified images.

Figure 4-24. (a) and (b) are frame 1 and frame 2 of the ‘Fountain’ sequence. (c),(d) are frame 1 and 2 with triangle nets.
Figure 4-25. (a),(b) the rectification results by Hartley’s method, with rectified triangle nets. (c) and (d) are the rectification results by the proposed method, with rectified triangle nets. (e) and (f) are results of the proposed method.
Figure 4-26 (a) and (b) are image 1 and 3 from the 'Fountain' image sequence. There is even more pose difference between the two views. In the same figure, (c) and (d) are the images with triangle nets generated from the control points detected by SIFT and matched using the proposed method. The triangle nets are of same topology but the size of the corresponding meshes in the two images is different from each other. In Figure 4-27, (a), (b) are the rectification results of Hartley’s method, with rectified triangle nets. While (c) and (d) are the rectification results of the proposed method, with rectified triangle nets also. (e) and (f) are results of the proposed method without triangle nets. Please notice the effects of pose adjustment in the rectified images.

Figure 4-26. (a) and (b) are frame 1 and frame 3 of the ‘Fountain’ sequence. (c),(d) frame 1 and 3 with triangle nets.
Figure 4-27. (a),(b) the rectification results by Hartley’s method, with rectified triangle nets. (c) and (d) are the rectification results by the proposed method, with rectified triangle nets. (e) and (f) are results of the proposed method.
Figure 4-28 (a) and (b) are image 1 and 4 from the 'Fountain' image sequence. It is easy to notice that the corresponding points are not on the same scan line and there is severe perspective distortion between the two views. In the same figure, (c) and (d) are the images with triangle nets generated from the control points detected by SIFT and matched using the proposed method. It can be noticed that the triangle nets are of same topology but the size of the corresponding meshes in the two images are different from each other. Figure 4-29 (a),(b) are the rectification results by Hartley’s method, with rectified triangle nets. While (c) and (d) are the rectification results by the proposed method, with rectified triangle nets also. (e) and (f) are results of the proposed method without triangle nets. It is easy to check that the corresponding points are on the same scan line now.
Figure 4-29. (a),(b) the rectification results by Hartley’s method, with rectified triangle nets. (c) and (d) are the rectification results by the proposed method, with rectified triangle nets. (e) and (f) are results of the proposed method.
Figure 4-30 (a) and (b) are image 1 and 5 from the 'Fountain' image sequence. It is easy to notice that the images are not rectified and there is severe pose difference between the two views. In the same figure, (c) and (d) are the images with triangle nets generated from the control points detected by SIFT and matched using the proposed method. It can be noticed that the triangle nets are of the same topology but the size of the corresponding meshes in the two images may differ from each other. Figure 4-31 (a),(b) are the rectification results by Hartley’s method, with rectified triangle nets. While (c) and (d) are the rectification results by the proposed method, with rectified triangle nets also. (e) and (f) are results of the proposed method without triangle nets. It is easy to check that the corresponding points are on the same scan line now.

Figure 4-30. (a) and (b) are frame 1 and frame 5. (c),(d) frame 1 and 5 with triangle nets.
Figure 4-31. (a),(b) the rectification results by Hartley’s method, with rectified triangle nets. (c) and (d) are the rectification results by the proposed method, with rectified triangle nets. (e) and (f) are results of the proposed method.
To evaluate the performance of methods quantitatively, we generate a comparison of the shape differences defined by Equation (4-6) also. The results are summarized in Table 4-2 and Table 4-3. The numbers listed in the table are the square roots of the $\Sigma_d$ defined in (4-6) for each triangle net, so the unit of the numbers in the tables is pixels.

Table 4-2. Structural difference of original image pairs and rectified pairs (in pixels).

<table>
<thead>
<tr>
<th>Façade Pairs</th>
<th>1-2</th>
<th>1-3</th>
<th>1-4</th>
<th>1-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Images</td>
<td>305.78</td>
<td>387.30</td>
<td>431.07</td>
<td>497.23</td>
</tr>
<tr>
<td>Hartley’s</td>
<td>242.68</td>
<td>313.64</td>
<td>357.56</td>
<td>446.15</td>
</tr>
<tr>
<td>Proposed</td>
<td>215.34</td>
<td>258.41</td>
<td>300.51</td>
<td>443.28</td>
</tr>
</tbody>
</table>

Table 4-3. Structural difference of original image pairs and rectified pairs (in pixels).

<table>
<thead>
<tr>
<th>Fountain Pairs</th>
<th>1-2</th>
<th>1-3</th>
<th>1-4</th>
<th>1-5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Images</td>
<td>532.36</td>
<td>608.11</td>
<td>636.87</td>
<td>698.31</td>
</tr>
<tr>
<td>Hartley’s</td>
<td>407.77</td>
<td>465.83</td>
<td>500.51</td>
<td>489.32</td>
</tr>
<tr>
<td>Proposed</td>
<td>351.65</td>
<td>401.58</td>
<td>454.07</td>
<td>471.75</td>
</tr>
</tbody>
</table>

As shown in Table 4-2 and 4-3, we can easily notice that the shape difference getting larger and larger between the image pairs. The proposed method can always reduce the shape difference between the original image pairs. Because of the optimized properties, it also performs better than Hartley's method.
5. Wide Baseline Dense Matching

5.1 A Survey of Stereo Dense Matching Algorithms

The stereo dense matching problem can be defined as "given two images of the same scene or object, compute a representation of its depth information". A most often used representation of the depth information is the disparity map, which is the difference between two corresponding points in two views along the epipolar line measured in pixels, as has been discussed in Chapter 1.

A most exhaustive survey of various stereo correspondence algorithms can be found in [Sze-01]. They also set up a website as a test bed for new algorithms, which is readily available at the WWW web\(^5\). Up to the year 2011, there are hundreds of algorithms available on the site and the software or source code of some algorithms is available also. The assumption of all these algorithms is that the input images are rectified, i.e., the corresponding points should be on the same scan line. Another implicit assumption is that the view points of the two images should be similar or the pose of the objects in the images should be similar to each other. To attain rectification, we can conduct image rectification as discussed in detail in Chapter 4. Another type of methods to rectify images is plane sweep. The advantage of plane sweep is that it can be applied in many different kinds of problems while at the same time is very efficient. Plane sweep was proposed first in [Col-96] and later extended in [Sze-99] and [Sai-99]. Recently, it has been extended to cylindrical surfaces as in [Zhe-07].

According the taxonomy proposed by Szeliski and Scharstein, most of stereo dense matching algorithms include a subset or all of the following four components: Matching cost computation; Cost (support) aggregation; Disparity computation and optimization and Disparity refinement.

\(^5\) The URL of the middlebury stereo vision website is: http://vision.middlebury.edu/stereo/.
A certain algorithm may not have all of the components, but by using these building blocks, many algorithms will be implemented readily in short time. Another advantage of this strategy is that the various algorithms can be tested not only as a whole, but also component-wise.

To establish a reliable matching between pixels, we need a similarity measure in the first place. The most simple similarity measures include sums of squared intensity differences (SSD) and absolute intensity differences (SAD). More recently, robust measures are proposed to limit the influence of mismatches. For a survey of the robust measures you are referred to [Vai-06]. More detailed studies of these matching costs can be found in [Hir-09].

To compute the disparity map from the established costs, certain kinds of optimization or selection strategies should be applied. All the methods can be divided into local and global categories. Two detailed survey of the local methods can be found in [Gon-07] and [Tom-08].

In contrast to the sliding window strategy used in the local methods, global optimization frameworks takes an image as an a two dimensional gridded random filed. Then the computation of the disparity map is transformed to a global optimization problem of find the best disparity field. In this category, the dynamic programming methods (See for instance, [Kol-06]), the graph cuts methods (See for instance, [Boy-01]) and the belief propagation methods (See for instance, [Sun-03]) are most applicable.

5.2 Wide Baseline Stereo Dense Matching

5.2.1 Previous Work
The dense matching problem of narrow baseline stereo has been studied in depth for the last several decades while the research on dense matching problem of wide baseline stereo just began in the last decade.

Based on our survey, Strecha of KU Leuven is one of the first researchers in the wide baseline dense matching area. In [Str-03], he proposed a PDE-based method for the reconstruction of precise 3D models. Basically, he started with a very sparse set of initial depth estimation for the seed points, and then developed an inhomogeneous time diffusion algorithm such that the dense matching is obtained. As declared, the algorithm can deal with occlusions, light changes, ordering changes along epipolar lines and extensive changes in scale. The short coming of the method is the computation complexity. As reported, the matching of four average size images needs 15 minutes to obtained.

Strecha also tried the probabilistic approach to the wide baseline stereo problem in [Str-04]. First, images are modeled as noisy measurements of an unknown irradiance image-function, and the dominant diffuse and i.i.d. pixel-color distributions are assumed. Then a smoothness regularizer was introduced to give shape to the prior beliefs about the world. This method is also expensive in terms of time and memory when high accuracy is needed.

The SIFT feature detector and descriptor proposed by Lowe in [Low-04] leads a major leap in the research on feature matching. Tola proposes a local descriptor named DAISY and fulfill the task of dense matching for wide baseline stereo in [Tol-10].

Quasi-dense wide baseline matching method is proposed in [Kan-07] by Kannala, in which he used a match propagation strategy over a set of seed points to generate a denser but not completely dense pixel correspondences. In this way, the computational cost is reduced.

5.2.2 The Proposed Method
As suggested in Section 1.5, we propose an efficient diagram for the purpose of wide baseline dense matching. The crucial step of the method is the image rectification that minimize the shape difference proposed and discussed in Chapter 4.

Through the introduction of the affine transformation acting on the stereo image, the shape difference is greatly reduced, which can be noticed in the experimental results in Section 4.4. We then obtain two rectified image with epiploar lines aligned with corresponding scan lines and what is more important is that the resultant is nearly narrow baseline to some extent.

In this research, we can then use any dense matching algorithm to process the rectified images obtained in Chapter 4. In Section 5.3, we will conduct a series of dense matching experiments using the graph cuts method implemented in [Boy-01].

The benefit of the proposed method is that,

- It is computationally cheap compared to the previous methods;
- It is flexible in that any dense matching algorithms can be applied;
- The range of the disparity is relatively larger than near baseline cases.

Based on our best knowledge, this is one of the first research work that approach the wide baseline stereo dense problem using the rectified image. And we also study the quality of dense matching with respect to the changes of baselines, as will be shown in Section 5.3.

5.3 Experimental Results and Discussions

The first data set we used is the synthetic “Corridor” data set from the University of Bonn (See [Fro-96] for detailed information), which are shown in Figure 5-1. The ground truth disparity images are shown in Figure 5-2. And a typical result generated by the graph cuts method is shown in Figure 5-3.
Figure 5-1. Left: The left stereo image, Right: the right stereo image.

Figure 5-2. Left: the ground truth Disparity image for the left stereo image. Right: the ground truth Disparity image for the right stereo image.

Figure 5-3. Typical output disparity images.
We also test the case when the original stereo pair is contaminated by noise of variance 100 (See Figure 5-4). The result (See Figure 5-5) is badly contaminated if compared with Figure 5-3.

![Figure 5-4. The contaminated stereo images with Gaussian Noise (Var = 100).](image)

Then we use the rectified image pairs generated in the 'Facade' image sequence and the 'Fountain' image sequence as the inputs for graph cuts methods to generate disparity maps. The inputs and results of image pairs 1-2, 1-3, 1-4 and 1-5 of the 'Facade' data set are shown in Figure 5-6 through 5-13. The results of Hartley's method are also used for comparison. The inputs and results of image pairs 1-2, 1-3, 1-4 and 1-5 of the 'Fountain' data set are shown in Figure 5-14 through 5-21.
Figure 5-6. (a),(b) are frame 1 and 2 of the 'Facade' sequences. (c),(d) are rectified images of proposed method. (e),(f) are rectified images of Hartley's method.
Figure 5-7. (a) is the disparity map generated from the rectified image pair of the proposed method. (b) is the color-coded version of (a). (c) is the disparity map generated from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).
Figure 5-8. (a),(b) are frame 1 and 3 of the 'Facade' sequences. (c),(d) are rectified images of proposed method. (e),(f) are rectified images of Hartley's method.
Figure 5-9. (a) is the disparity map generated from the rectified image pair of the proposed method. (b) is the color-coded version of (a). (c) is the disparity map generated from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).
Figure 5-10. (a), (b) are frame 1 and 4 of the 'Facade' sequences. (c), (d) are rectified images of the proposed method. (e), (f) are rectified images of Hartley's method.
Figure 5-11. (a) is the disparity map generated from the rectified image pair of the proposed method. (b) is the color-coded version of (a). (c) is the disparity map generated from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).
Figure 5-12. (a), (b) are frame 1 and 5 of the 'Facade' sequences. (c), (d) are rectified images of the proposed method. (e), (f) are rectified images of Hartley's method.
Figure 5-13. (a) is the disparity map generated from the rectified image pair of the proposed method. (b) is the color-coded version of (a). (c) is the disparity map generated from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).
Figure 5-14. (a),(b) are frame 1 and 2 of the 'Fountain' sequences. (c),(d) are rectified images of proposed method. (e),(f) are rectified images of Hartley's method.
Figure 5-15. (a) is the disparity map generated from the rectified image pair of the proposed method. (b) is the color-coded version of (a). (c) is the disparity map generated from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).
Figure 5-16. (a),(b) are frame 1 and 3 of the 'Fountain' sequences. (c),(d) are rectified images of proposed method. (e),(f) are rectified images of Hartley's method.
Figure 5-17. (a) is the disparity map generated from the rectified image pair of the proposed method. (b) is the color-coded version of (a). (c) is the disparity map generated from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).
Figure 5-18. (a),(b) are frame 1 and 4 of the 'Fountain' sequences. (c),(d) are rectified images of proposed method. (e),(f) are rectified images of Hartley's method.
Figure 5-19. (a) is disparity map generated from the rectified image pair of the proposed method. (b) is the color-coded version of (a). (c) is disparity map generated from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).
Figure 5-20. (a),(b) are frame 1 and 5 of the 'Fountain' sequences. (c),(d) are rectified images of proposed method. (e),(f) are rectified images of Hartley's method.
Figure 5-21. (a) is the disparity map generated from the rectified image pair of proposed method. (b) is the color-coded version of (a). (c) is disparity map generated from the rectified image pair of Hartley's method. (d) is the color-coded version of (c).

From the results of the experiments, we can see that the proposed diagram can produce better results from the proposed rectification method when compared with the results generated by Hartley's method. In almost all the cases, our proposed method can produce disparity map of better details and larger range, thus can produce more details in 3D model.

If we carefully compare the disparity maps with respect to the changes of baseline within both of the image sequences, we notice that the quality is degraded gradually when the baseline gets longer.
6. Conclusion and Future Work

6.1 Summary of the Work

In this research we focus our efforts on the wide baseline stereo problem which has getting more and more attention in the computer vision community in recent years. To solve the problem, we proposed an efficient diagram and developed a software system which can generate a disparity map from just two wide baseline images and no more other information is needed beforehand.

Throughout all phases of the research, we make use of various geometric information and structural characteristics of the feature point sets.

An innovative feature matching algorithm with geometric and structural constraints is proposed and tested using the SIFT descriptors. The major advantage of the proposed feature matching method is that it combines the power of spectral analysis and affine transformation, thus avoids mismatches of feature points in images of repeated or symmetric patterns. The testing results show a performance enhancement when compared to a state-of-the-art matching method. Another advantage of the proposed method is its independence of the construction of feature vectors so that it can be applied to other kinds of feature descriptors. It can also be used separately for applications such as object identification and recognition in images with wide baselines.

Once a reliable feature matching is obtained, we can estimate the epipolar geometry within two uncalibrated images and then fulfill image rectification. In this research, we proposed a new definition of image rectification that combines the concept of epipolar rectification and minimization of shape difference. To compensate the shape difference introduced by perspective distortion that is ubiquitous in various wide baseline applications, we proposed a new method based on the analysis of the structural characteristics of triangle nets. The
triangle nets are built as Delaunay triangle network from the control points generated by the proposed matching method.

By choosing appropriate criterions of optimization, we can reduce the shape and pose difference in the images. Several data sets was used to test the performance of the proposed method and comparisons to the state-of-the-art Hartley's method is provided. In almost all the cases, our proposed method can produce better results if judged using the shape minimization criterion.

Dense matching for rectified stereo images is then possible and then tested. Test results show that the pipeline is practical for many wide baseline set up. The proposed diagram and software system may be deployed and tested in wide baseline real world applications.

To the best of our knowledge, this is the first research that makes use of rectified image for disparity estimation. And the performance of the dense matching of rectified images with respect to the changing baseline is also tested.

6.2 Future Work

Wide baseline stereo is a new research area that attracts much attention from the computer vision community in the last decade or so. There are many possible ways to extend the ideas and methods proposed in this research.

For feature points matching or sparse matching, we may want to enhance the proposed method so that it can be used to detect more matches in difficult situations. A possible way is to employ local image normalization. If an image can be segmented into several parts of continual depth changes, we can apply different affine transformations on different areas of interest, which will make the algorithm more adaptive and thus combine the advantage of global topological analysis and local geometric analysis.
For the projective rectification part, the topology of the triangle net can be optimized by appropriately selection of the distribution of the feature points. The experiments show that the results of rectification are affected by the topology of the triangle net. If we can select the control points before the construction of the triangle net, the performance of rectification may be further improved.

Dense matching may be enhanced with the introduction of local descriptor matching rather than just the matching of intensity value. This will remove some ambiguity and give more accurate details in the disparity map.

As a whole, the system can be deployed in more real world wide baseline applications such as three-dimensional wide area surveillance for further test and enhanced.

The disparity maps can be used to generate 3D model of the scene if calibration information can be obtained. This is a very interesting topic for further study.
Bibliography


Vita

Wei Hao was born in Hubei province, China. He attended Tianjin Polytechnic University where he majored in Electrical Engineering and received a Bachelor of Science degree in 1997. After that, he continued to pursue his Master of Science degree in Tianjin University with a major in Pattern recognition and Intelligent System. After year 2000, he worked as a software engineer in the telecommunication and Internet industries. He came to University of Tennessee in fall 2003 and joined the Imaging, Robotics, and Intelligent Systems Laboratory afterward, where he completed his Doctor of Philosophy degree in 2011.