Locally Adaptive Stereo Vision Based 3D Visual Reconstruction

by

Jinjin Li

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved January 2017 by the
Graduate Supervisory Committee:

Lina Karam, Chair
Chaitali Chakrabarti
Nital Patel
Andreas Spanias

ARIZONA STATE UNIVERSITY

May 2017

ABSTRACT

Using stereo vision for 3D reconstruction and depth estimation has become a popular and promising research area as it has a simple setup with passive cameras and relatively efficient processing procedure. The work in this dissertation focuses on locally adaptive stereo vision methods and applications to different imaging setups and image scenes.

Solder ball height and substrate coplanarity inspection is essential to the detection of potential connectivity issues in semi-conductor units. Current ball height and substrate coplanarity inspection tools are expensive and slow, which makes them difficult to use in a real-time manufacturing setting. In this dissertation, an automatic, stereo vision based, in-line ball height and coplanarity inspection method is presented. The proposed method includes an imaging setup together with a computer vision algorithm for reliable, in-line ball height measurement. The imaging setup and calibration, ball height estimation and substrate coplanarity calculation are presented with novel stereo vision methods. The results of the proposed method are evaluated in a measurement capability analysis (MCA) procedure and compared with the ground-truth obtained by an existing laser scanning tool and an existing confocal inspection tool. The proposed system outperforms existing inspection tools in terms of accuracy and stability.

In a rectified stereo vision system, stereo matching methods can be categorized into global methods and local methods. Local stereo methods are more suitable for real-time processing purposes with competitive accuracy as compared with global methods. This work proposes a stereo matching method based on sparse locally adaptive cost aggregation. In order to reduce outlier disparity values that correspond to mis-matches, a novel sparse disparity subset selection method is proposed by assigning a significance status to candidate disparity values, and selecting the significant disparity values adaptively. An adaptive guided filtering method using the disparity subset for refined cost aggregation and disparity calculation is demonstrated. The proposed stereo matching algorithm is tested on the Middlebury and the KITTI stereo evaluation benchmark images. A performance analysis of the proposed method in terms of the l_0 norm of the disparity subset is presented to demonstrate the achieved efficiency and accuracy.

ACKNOWLEDGMENTS

Pursuing a Ph.D. in the past years has been a treasured and adventurous journey during the best time in my life. I have received endless support and encouragement from a great number of individuals. I would like to express utmost gratitude to my advisor Lina Karam for her constant support, understanding, encouragement and inspiration during my Ph.D. study. She has been a tremendous mentor for me. Her kind patience and guidance have been the strong back-support for me to concur all the difficulties and proceed throughout my doctoral program and research projects. Her enthusiasm and serious attitude in research have inspired me and motivated me to do my best. I feel so fortunate to become her student and work with her in the past years.

I would like to express my sincere gratitude to members of my Ph.D. committee, Prof. Chaitali Chakrabarti, Dr. Nital Patel and Prof. Andreas Spanias for their kind guidance and constructive feedback. I enjoyed the opportunity to work closely with them on some of my past research projects and courses. Their willingness to share ideas and raise insightful discussions has been of great help to my research and study. I would like to thank Ms. Bonnie Bennett for being a great colleague and project manager during the Intel research project.

I would like to thank the Electrical Engineering department and the ECEE Graduate Program Chair Dr. Joseph Palais for giving me the opportunity to work as a teaching assistant. I would like to thank the ECEE graduate advisors Ms. Toni Mengert, Ms. Darleen Mandt, Ms. Ginger Rose and Ms. Jenna Marturano for their advising on my Ph.D study plan, graduate timeline and graduation preparation. I would like to thank Mr. Clayton Javurek for being such a great and helpful lab manager for EEE120.

As a member of the IVU lab at Arizona State University, I would like to extend my thanks to the fellow lab members for their support. I would like to thank Qian Xu for her inspring research advice, understanding and valuable friendship. Many thanks to Milind Gide for being a very helpful and supportive fellow TA as well as an excellent friend. I would like to thank Alireza Golestaneh for all the help and discussions about both research and life. I would also like to thank my other lab mates, Bashar Haddad, Samuel Dodge, Juan Andrade Rodas, Srenivas Varadarajan, Asaad Said and Vinay Kashyap for their kind help and companionship throughout the past years.

I am indebted to my dearest friends. Your continuous warmth and friendship have been a great power to me through these difficult years. I would like to thank my best friend Yuan Yuan for her side-by-side companionship, listening and the every-day tiny joy she brought to me during my first four years of Ph.D studies. I would like to thank my best friend Xue Zhang for always encouraging me and helping me overcome setbacks and stay sane during the past two years. I would like to extend my thanks to my dearest friends, Li Ji, Yu Dong, Jing Li, Xiaoyan Deng and Hualin Lv for always being there for me when I needed a companion, listening and sharing.

Lastly and most importantly, I would like to thank my family, especially my parents and my grand-parents, who raised me with unconditional love and supported me in all my pursuits. My graduate education at Arizona State University would not be possible without the numerous sacrifices and support by my father. He is my life mentor and idol. I would like to thank my dearest mother, who has been always supportive, encouraging and joyful to me throughout these years.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Chapter 1

INTRODUCTION

This chapter presents the motivation behind the work in this dissertation and briefly summarizes the contributions and organization of this dissertation.

1.1 Motivation

Three-dimensional reconstruction and modeling is one of the major areas in computer vision and computer graphics. 3D reconstruction is a process that estimates the 3D structure and appearance of objects or scenes. In recent decades, people are paying more and more attention to obtaining 3D information using 3D reconstruction due to its wide range of applications in various fields. 3D reconstruction plays important roles in automatic target recognition and tracking [2–4], medical diagnostic and surgery [5–7], large scene reconstructions [8], remote sensing and global information systems (GIS) [9, 10], teleconferencing [11, 12], commercial 3D television and 3D gaming, 3D cinema creation, industrial automated assembly and machine vision for product inspection and quality control [13, 14].

In different applications, the 3D reconstruction can be done with various technologies, including mechanical measurement of objects using depth gauge, radiometric scanning methods, including time-of-flight lasers [15], microwave or ultrasound [16], and camera-sensor based methods using images and videos. 3D reconstruction can be realized directly using 3D cameras such as time-of-flight cameras and structured light systems [17] that generate the range information of the 3D scene by projecting laser beams on to the scene and computing the delay of the reflected laser rays from scene objects. In camera-sensor based methods, two-dimensional (2D) images and video frames are the 2D projection of 3D objects on the camera sensors. Thus by extracting features from 2D images/videos, the 3D structure of the scene can be reconstructed. 3D reconstruction using image and video processing techniques is attracting a lot of attention because of the readily available non-intrusive image and video capture and storage resources, and is perceptually motivated due to the ability to achieve 3D reconstruction from 2D visual data as demonstrated by the Human Visual System. Furthermore, recent image/video processing techniques demonstrated relatively good 3D reconstruction results for various applications. Using image and video for 3D reconstruction is also more efficient in terms of implementation, computation and transmission, as compared to other methods that rely on high-maintenance equipment such as X-ray microscopy [18] and 3D laser scanners [15, 17]. Using passive 2D camera sensors, 3D reconstruction can be achieved using a single 2D image, a stereoscopic imaging system, or multiple images or video frames. The stereoscopic method is one of the most popular 3D reconstruction methods because of the relative simple image setup, reliable 3D estimation results and efficient computation procedure.

Stereo vision plays a significant role in many 3D reconstruction applications, and the binocular camera setup shares similar concepts as the human vision for 3D perception. 3D vision for humans is caused by the fact that the projected points of the same point in space on the two human eyes are located at different distances from the center of focus (center of fovea). The difference between the distances of the two projected points, one on each eye, is called disparity. The disparity information is processed by high levels of the human brain to produce a feeling of the distance of objects in 3D space. Stereoscopic vision methods adopt a similar idea to obtain stereo 2D images of the same scene with two cameras that have a small rotation or shift in their relative locations. The 3D information of the image scene is represented using the depth value or disparity value in stereo vision methods. The depth value of an image point is the distance between the 3D point corresponding to the image point and the camera projection center. The disparity corresponds to the coordinate difference between the two projected 2D image points in stereo images. The goal of stereo matching is to estimate the depth or disparity information using either a sparse feature point set or dense image pixels.

According to different applications of stereo vision, the stereo vision setup can be categorized into two different classes: angled stereo camera setup with unrectified stereo images and horizontally planar aligned stereo camera setup with rectified stereo images. In both stereo vision setup schemes, the main procedure, called stereo matching, is to first find matching 2D points, in stereo images, that correspond to the same 3D point in the image scene. Using the displacement of the matching 2D points, the depth or disparity information can be estimated. In the angled stereo camera setup, since the poses of two cameras include both a rotation and translation, the corresponding 2D image points in the stereo views may not be located on the same horizontal axis. In the horizontally aligned stereo setup, the stereo cameras are pointing to the image scene with the same angle and there is only a horizontal translation between stereo cameras. Thus the stereo camera planes are co-planar, and the matching 2D image points in the stereo views are on the same horizontal axis, reducing the search of matching point to the horizontal axis only. It is possible to transform the angled stereo cameras to be horizontally aligned with co-planar camera planes, and align the matching pixels to be on the same horizontal axis in stereo images. This linear transformation is called image rectification [19], and it is usually applied before performing the stereo matching procedure for horizontally aligned images.

Although various stereo matching methods have been proposed, there are still many challenges in stereo vision for different applications. In angled stereo vision schemes, the matching points in the unrectified stereo views are typically found using feature detection and matching methods. The feature detection and matching methods generate sparse matching pairs between the stereo views. Existing feature point detection methods such as Harris corner detection [20], SIFT [21], SURF [22], BRISK [23] and FAST [24] are able to generate reliable and scale-invariant feature points in natural images with various textures and features, luminance changes and object discontinuities. These feature points usually include image corners, edges and texture points. With a sufficient number of matching feature points in the stereo views, the 3D scene can be reconstructed. The depth information at the locations of the detected feature points is calculated using the detected and matched 2D feature points in the stereo views and using the relative stereo camera positions. This type of stereoscopic system has been widely used in object recognition and tracking, large scene reconstruction [9] and intelligent vehicle systems such as advanced driver assist systems [25]. However, there are some applications for which the imaged objects and scenes lack sufficient texture and features such as edges and corners. The aforementioned feature detection methods are not suitable for these applications. Furthermore, passive stereo vision is rarely explored for such applications. One such application area is the industrial automated machine vision for product inspection and quality control.

In semiconductor manufacturing and quality control, the inspection of defective solder joints on Ball Grid Array (BGA) for defect detection is important in both manufacturing and in the post-inspection process because defective solder joints can cause problems in semi-conductor products, including non-wets and infant mortality resulting in failed parts. The uniformity of solder ball heights and coplanarity across the unit surface affects the reliability of the BGA package significantly. Traditional solder ball height and coplanarity measurement methods include 3D X-ray laminography [26], laser scanning [27], confocal microscopy [28] or machine vision approaches with prior assumptions or prior references [29–32]. These methods estimate ball height information with high-cost equipment, complicated setup, expensive computation load and slow speed. Stereo vision methods for ball height and coplanarity estimation are rarely explored due to the characteristics of solder ball images such as textureless, edgeless and smooth surfaces with less color change. Due to the simpler setup and more efficient processing nature of stereo vision, an efficient image feature detection and matching method is needed to extract 2D features in solder ball images. With such extracted features, the reconstruction of solder ball height and coplanarity can be implemented efficiently using stereo vision for real-time inspection and measurement purposes.

For rectified and aligned stereo vision fields, dense depth or disparity maps are generated as the output using stereo images and various stereo matching algorithms. Dense stereo matching methods have attracted a lot of attention from researchers in computer vision and computer graphics because the dense disparity maps computed using stereo matching can be applied to areas such as teleconferencing, depth-image-based-rendering in multiview displays and coding, to name a few. Existing stereo matching approaches are summarized and compared based on the Middlebury image database in [33]. The stereo matching algorithms are categorized into two major classes: global methods and local methods. Both the accuracy and computation complexity of the computed dense disparity map are crucial in most stereo matching algorithms. Global methods using energy optimization methods such as belief propagation [34, 35], graph cut [36] and dynamic programming [37] produce accurate disparity maps, but the computation load and memory consumption are high as compared to local methods. Local stereo methods are more suitable for real-time applications because the computation complexity is relatively low and they can be more parallelized for faster processing. Recently proposed local stereo methods produce competitive disparity estimation results in terms of accuracy as compared to the global methods, and the computation is efficient using the integral image technique or hardware optimization.

Although the recent local stereo matching methods are able to compute accurate disparity maps with a relatively low computational complexity, the task of efficiently estimating dense disparity maps using stereo methods is still challenging. The captured stereo images are disturbed by surrounding environmental noise such as sensor noise and illumination changes. In addition, image scenes with textureless regions, slanted surfaces and occlusions make the stereo matching problem more complicated. Recently researchers are seeking possibilities to improve the performance of local stereo matching methods in order to achieve a higher accuracy and faster computation speed for real-time applications with various possible image scenes. In addition, there is a need to improve the performance of local stereo methods in the presence of noise, uniform areas, depth discontinuities and occlusions.

1.2 Contributions

In Chapter 3, a novel stereo vision system is proposed for solder ball height and ball grid array (BGA) coplanarity estimation. The proposed method enables in-line real-time automatic product 3D characterization and inspection. We propose a novel iso-contour-based feature detection and matching algorithm for textureless objects. Compared to other existing BGA inspection systems, the proposed method has benefits in mainly three aspects: 1) the proposed image processing and machine vision method is computationally efficient compared to other image processing techniques for solder ball height detection; 2) the solder ball height and package coplanarity results by the proposed method show high accuracy and reliability compared to other far more sophisticated methods; 3) the imaging setup procedure and equipment calibration and adjustment of the proposed method is much simpler than other existing methods. The proposed stereo setup using two area-scan cameras captures enough details of the solder balls with diameters around 200-400 μm. The proposed stereo vision based method is able to reconstruct the 3D structure of the solder balls without any prior height and coplanarity information. Furthermore, with the novel iso-contour tree structure based matching scheme, the feature points of the textureless solder ball images can be accurately located and matched between stereo views. We show that the proposed system is capable to estimate the solder ball height in different height ranges. Both the ball height results and coplanarity results show high correlation and stability compared to the existing confocal inspection method and laser scanning method.

In Chapter 4, a novel stereo matching method using sparse locally adaptive cost aggregation is proposed to compute a more accurate disparity map with less complexity and redundancy. The proposed local stereo matching method consists of a fast initial cost aggregation stage followed by a refined cost aggregation that is only performed over a sparse subset of disparities. In the proposed method, the cost aggregation is performed in a locally adaptive manner by adapting the support region to the local image intensity and structure. In order to reduce outlier disparity values that correspond to mis-matches, a novel sparse disparity subset selection method is proposed by assigning a significance status to candidate disparity values, and selecting the significant disparity values adaptively. A novel adaptive guided filtering method using the disparity subset for refined cost aggregation and disparity calculation is demonstrated. The disparity maps are refined through occlusion handling and post-processing steps using localized-support-region-based propagation and weighted propagation. We show that the proposed sparse adaptive guided filter produces accurate dense disparity results using the Middlebury stereo matching database version 2 and version 3 [38] and the KITTI 2015 [39] stereo database, and that the proposed method outperforms previous popular local methods [40–42] and some semi-global stereo matching methods [43]. We also conduct a performance analysis by varying the size of the sparse disparity subset, and show that using a sparse disparity subset for cost aggregation helps in removing matching ambiguities and in improving the disparity estimation accuracy, while preserving the computational efficiency.

1.3 Organization

This dissertation is organized as follows. Chapter 2 presents background material on concepts that are related to the work in this dissertation. Chapter 3 presents the proposed stereo vision based automated solder ball height and coplanarity detection method. The method is tested on different BGA packages and experimental results are evaluated using the measurement capability analysis (MCA) to demonstrate the accuracy, repeatability and reproducibility of our proposed method. Chapter 4 presents a novel local stereo matching method using sparse locally adaptive cost aggregation. The performance analysis and results of the proposed method are obtained based on stereo images from the Middlebury [38] and the KITTI [39] benchmark databases. Finally, Chapter 5 summarizes the contributions of this work and discusses future research directions.

Chapter 2

BACKGROUND

This chapter gives some background knowledge on 3D reconstruction and stereo vision. In Section 2.1, the camera geometry and the pinhole camera model, which are basic for the analysis of 3D systems, are illustrated. In Section 2.2, the two-view geometry is discussed in two aspects: the epipolar geometry and the disparity calculation. The epipolar geometry between two views and the fundamental matrix are introduced in Section 2.2.1 for further use. In rectified stereo images, the relationship between disparity and depth is illustrated in Section 2.2.2.

2.1 Camera Geometry

In computer vision, homogeneous representations of lines and points are described as follows. A line passing through the point $(x, y)^T$ can be described as:

$$ax + by + c = 0 \tag{2.1}$$

So, the vector $\boldsymbol{l} = (a, b, c)^T$ is the homogeneous representation of a line. Alternatively, the line can be described using the vector $(a, b, c)^T$. Therefore, equation (2.1) can be written in the form of two inner products as follows:

$$\boldsymbol{l^T} \cdot \boldsymbol{x} = (a, b, c) \cdot (x, y, 1)^T = 0 \tag{2.2}$$

According to (2.2), a 2D point can be expressed using a three-dimensional vector $(x, y, 1)^T$, whose third element serves as a scale factor. For a more general case, the homogeneous representation $(x, y, z)^T$ of a point denotes the point $(x/z, y/z)^T$ in 2D-vector form. Similarly, the three-dimensional point $\boldsymbol{X} = (x, y, z)^T$ can be represented using the homogeneous notation as $\boldsymbol{X} = (x, y, z, 1)^T$ and the plane $\boldsymbol{\pi}$ on which $\boldsymbol{X}$ lies is represented in homogeneous form as

$$\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3, \pi_4)^T \tag{2.3}$$

A 3D point $\boldsymbol{X}$ lying on the plane $\boldsymbol{\pi}$ satisfies:

$$\boldsymbol{\pi^T} \cdot \boldsymbol{X} = (\pi_1, \pi_2, \pi_3, \pi_4) \cdot (x, y, z, 1)^T = 0 \tag{2.4}$$

A basic camera model is the projective pinhole camera geometry as shown in Fig. 2.1. It is assumed that the camera center is the origin of a Euclidean coordinate system. The camera center $\boldsymbol{O_C}$ is also called the optical center. The image captured by the camera is typically projected on the camera plane (also called the focal plane) behind the camera center, with a negative focal length $-f$ on the z-axis. In addition, according to the imaging mechanism of cameras, the image on the camera image plane is upside-down with respect to the real scene. In the model in Fig. 2.1, the image plane is placed to be in front of the camera center, and the distance from the image plane to the center point is the focal length f. In this latter case, the image does not have to be inverted. The plane, which passes through the camera center and is parallel to the image plane, is denoted as the principal plane. The line, passing through the camera center and perpendicular to the image plane, is called the principal axis. The intersection of the principal axis with the image plane is a point called the principal point $\boldsymbol{PP}$.

In this camera model (Fig. 2.1), a 3D point in space at a position $\boldsymbol{X} = (x, y, z)^T$ can be mapped to the image plane by forming a line starting at the camera center to the point $\boldsymbol{X}$, and the intersection of this line with the image plane is a 2D point $\boldsymbol{x}$ lying on the image plane. Using the similar triangles, the position of $\boldsymbol{x}$ with respect to the camera center can be represented in homogeneous coordinates as $(\frac{f \cdot x}{z}, \frac{f \cdot y}{z}, f, 1)^T$

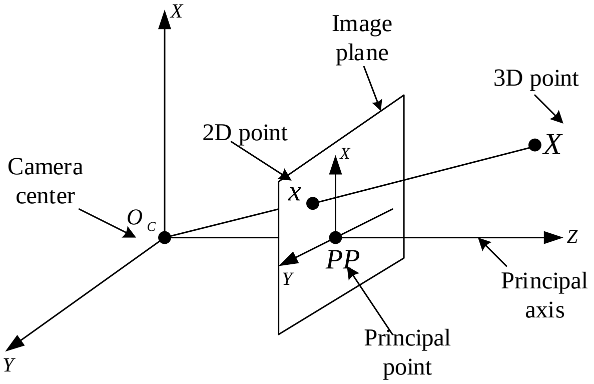


Fig. 2.1: Pinhole Camera Model.

in the 3D space. The homogeneous representation of the 2D point $\boldsymbol{x}$ on the image plane is $(\frac{f \cdot x}{z}, \frac{f \cdot y}{z}, 1)^T$, while the origin of the image plane is the same as the principal point $\boldsymbol{PP}$. A projective camera [44] is modeled though the projection equation as

$$\boldsymbol{x} = P\boldsymbol{X} \tag{2.5}$$

where $\boldsymbol{x}$ represents the 2D point in homogeneous representation, that is, it is a 3×1 dimensional vector. P is a 3×4 projection matrix. $\boldsymbol{X}$ stands for the 3D point in homogeneous representation, and it is a 4×1 dimensional vector.

The Projection matrix P can be represented as

$$P = K[\ R \mid \boldsymbol{t}\ ] \tag{2.6}$$

where R is a 3×3 rotation matrix representing the orientation of the camera coordinates with respect to the world coordinates, $\boldsymbol{t}$ is a 3×1 translation vector which shifts the camera center $\boldsymbol{O_C}$ with respect to the world coordinate system, and $\boldsymbol{t}$ is given by

$$\boldsymbol{t} = -R \cdot \boldsymbol{O_C} \tag{2.7}$$

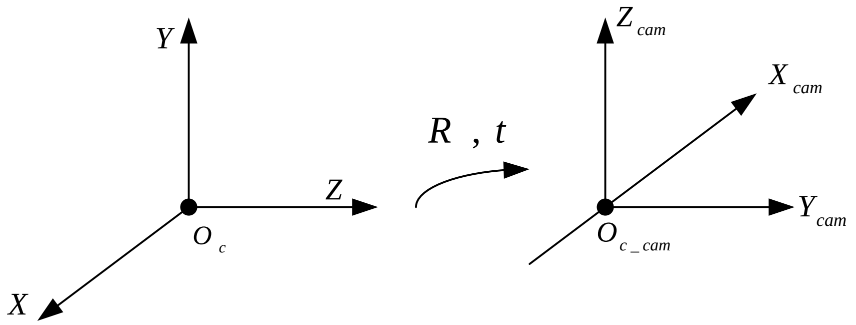


Fig. 2.2: Rotation and Translation between Two Coordinates.

The transformation (including rotation and translation) between different coordinates is shown in Fig. 2.2. In (2.6), K is the intrinsic camera matrix, called the camera calibration matrix and is given by

$$K = \begin{bmatrix} f & s & u \\ 0 & \alpha f & v \\ 0 & 0 & 1 \end{bmatrix} \tag{2.8}$$

where f is the focal length of the camera, α is the aspect ratio of the pixel size on the image plane in the x and y directions, (u, v) represents the coordinates of the principal point with respect to the left bottom corner of the image plane, and s is the skew factor which is non-zero if the x and y axes of the image coordinates are not perpendicular.

2.2 Two-View Geometry

2.2.1 Epipolar Geometry and Fundamental Matrix

The geometry between two views of the same scene can be represented using the epipolar geometry. The epipolar geometry is illustrated in Fig. 2.3. Suppose a 3D point $\boldsymbol{X}$ in space is projected into two views to generate 2D points $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$,

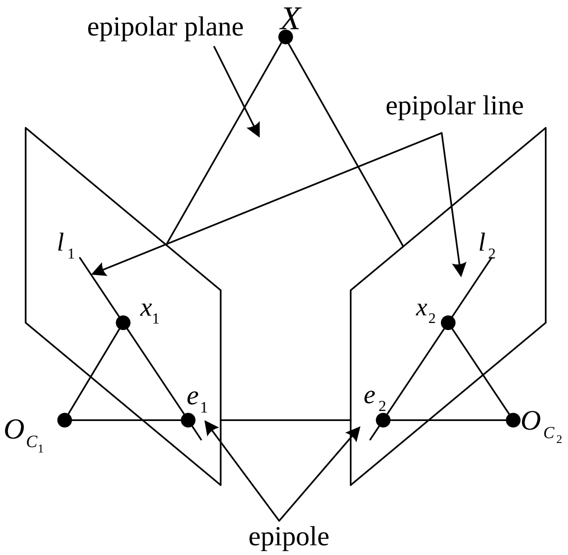


Fig. 2.3: Epipolar Geometry between Two Views.

respectively. As three points form a plane, $\boldsymbol{x_1}$, $\boldsymbol{x_2}$ and $\boldsymbol{X}$ would lie on a common plane $\boldsymbol{\pi_E}$. The plane $\boldsymbol{\pi_E}$ is denoted as the epipolar plane. The line connecting the two camera centers is the baseline between two views, and it also lies on the epipolar plane. The intersection points of the baseline with the two views are the epipoles denoted by $\boldsymbol{e_1}$ and $\boldsymbol{e_2}$, one in each view. The line, which connects the 2D point and the corresponding epipole on the same plane, is called the epipolar line. The epipolar line $\boldsymbol{I_2}$ in the second view is parallel to the ray through $\boldsymbol{x_1}$ and the camera center $\boldsymbol{O_{C_1}}$, and $\boldsymbol{I_2}$ is the projected image in the second view of that ray. Since the 3D point $\boldsymbol{X}$ lies on the ray through $\boldsymbol{x_1}$ and camera center $\boldsymbol{O_{C_1}}$, the projected 2D point $\boldsymbol{x_2}$ of the 3D point $\boldsymbol{X}$ in the second view must be lying on the epipolar line $\boldsymbol{l_2}$.

From the above discussion, any point $\boldsymbol{x_2}$ in the second image that matches the point $\boldsymbol{x_1}$, must lie on the epipolar line $\boldsymbol{l_2}$, and the epipolar line $\boldsymbol{l_2}$ in the second view is the mapped image of the ray through $\boldsymbol{x_1}$ and camera center $\boldsymbol{O_{C_1}}$. So, there is a mapping between the 2D point in one view and the epipolar line in the other view. The fundamental matrix F_{12} is defined to represent this mapping relationship $\boldsymbol{x_1} \xrightarrow{F_{12}} \boldsymbol{l_2}$. Similarly, the fundamental matrix F_{21} represents the mapping between $\boldsymbol{x_2}$ and $\boldsymbol{l_1}$. The fundamental matrix is the algebraic representation of the epipolar geometry, and it is a 3×3 matrix with a rank of 2.

The epipolar line $\boldsymbol{l_2}$ corresponding to the 2D point $\boldsymbol{x_1}$ is represented by

$$\boldsymbol{l_2} = F_{12}\boldsymbol{x_1} \tag{2.9}$$

The fundamental matrix relates to the corresponding epipoles $\boldsymbol{e_1}$ and $\boldsymbol{e_2}$ as follows:

$$\boldsymbol{e_2^T} F_{12} = 0 \tag{2.10}$$

$$F_{12}\boldsymbol{e_1} = 0 \tag{2.11}$$

From (2.10) and (2.11), the epipole in the first view $\boldsymbol{e_1}$ is the right null-space of F_{12}, and the epipole in the second view $\boldsymbol{e_2}$ is the left null-space of F_{12}. Epipoles for two views can be computed from the fundamental matrix using the singular value decomposition (SVD). Suppose M is a $m \times n$ matrix; the singular value decomposition of M is in the form of

$$M = U\Sigma V^T \tag{2.12}$$

where U is a $m \times m$ unitary matrix, Σ is a $m \times n$ diagonal matrix and V is a $n \times n$ unitary matrix. The diagonal entries of Σ are the singular values of M. The column vectors of U are the left-singular vectors of M and the column vectors of V are the right-singular vectors of M. That is, the relationship of the corresponding left-singular vector $\boldsymbol{u}$, right-singular vector $\boldsymbol{v}$, and the singular value $sigma$ can be represented as

$$\boldsymbol{u^T} M = \sigma \boldsymbol{v^T} \tag{2.13}$$

$$M\boldsymbol{v} = \sigma \boldsymbol{u} \tag{2.14}$$

Since the rank of the fundamental matrix is 2, in the SVD of F the third singular value in the diagonal matrix is zero. According to (2.13) and (2.14), if U and V are, respectively, the left-singular and right-singular matrices in the SVD of the fundamental matrix F, the third column of the left-singular matrix U and the third column of the right-singular matrix V would correspond to the left null vector and the right null vector of the fundamental matrix F, respectively, and would satisfy (2.10) and (2.11). Thus, the epipole in the second view is computed from the third column of the left-singular matrix U in the SVD of the fundamental matrix F and the epipole in the first view is given by the third column of the right-singular matrix V in the SVD of the fundamental matrix F.

As stated in [44], the two 2D points $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$, corresponding each to the projection of the 3D point $\boldsymbol{X}$ into two different views, are related as follows:

$$\boldsymbol{x_2^T} F_{12} \boldsymbol{x_1} = 0 \tag{2.15}$$

and

$$\boldsymbol{x_1^T} F_{21} \boldsymbol{x_2} = 0 \tag{2.16}$$

From (2.15) and (2.16), the fundamental matrices, F_{12} and F_{21}, can be related as

$$F_{21} = F_{12}^T \tag{2.17}$$

In this dissertation, the fundamental matrix F_{12} is denoted as F for simplicity.

2.2.2 Rectified Two-View Geometry and Disparity

The depth values of a 3D scene correspond to the distances of the 3D points to the camera center, while the disparity values in rectified stereo vision represent the differences between the coordinates of matching points in stereo views. There is an implicit relationship between the disparity and the depth. If the disparity shift of a

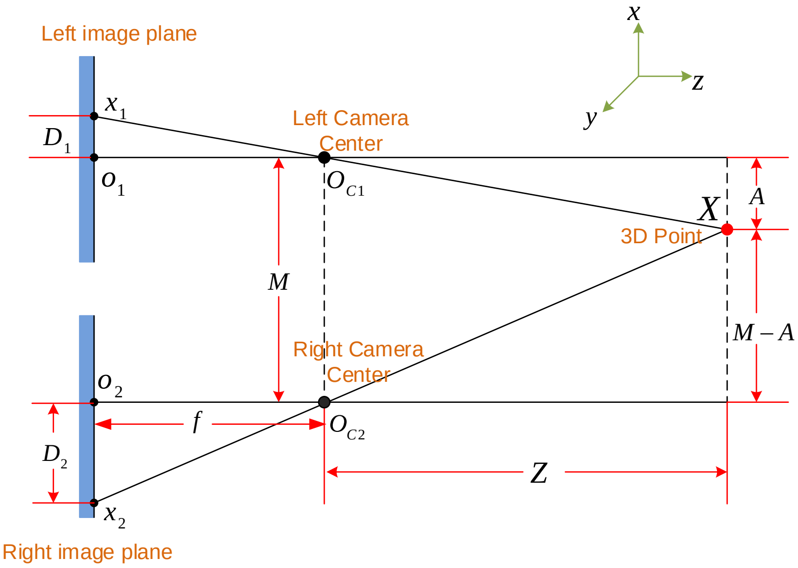


Fig. 2.4: Geometry of the Disparity vs. Depth Relationship for a 3D Point.

3D point is known, the corresponding depth value can be estimated, with the prior knowledge of the camera movement and the focal length of the camera [45, 46]. The geometry of the disparity versus depth relationship is shown in Fig. 2.4.

In Fig. 2.4, $\boldsymbol{x_1}$ is the projected 2D point of the 3D point on view 1, and $\boldsymbol{x_2}$ is the projected 2D point of the 3D point $\boldsymbol{X}$ on view 2. $\boldsymbol{O_{C1}}$ and $\boldsymbol{O_{C2}}$ are the stereo camera centers with a horizontal camera shift of M. The line connecting camera centers $\boldsymbol{O_{C1}}$ and $\boldsymbol{O_{C2}}$ is called baseline. The perpendicular distance from the 3D point $\boldsymbol{X}$ to the baseline is the depth value of the 3D point. This depth is denoted as Z. The distance from the baseline to the image plane is the focal length of the camera, denoted as f. The distance between $\boldsymbol{x_1}$ and the principal point $\boldsymbol{O_1}$ is denoted as D_1, and the distance between $\boldsymbol{x_2}$ and the principal point $\boldsymbol{O_2}$ is denoted as D_2. Using the triangular geometry, the disparity of the 3D point $\boldsymbol{X}$ on the stereo 2D views, can then be presented as

$$D = D_1 + D_2 = f \cdot \frac{M}{Z} \tag{2.18}$$

From the analysis above, the disparity D of corresponding 2D points in stereo views is the inverse of the depth Z with a scalar of the product of the known focal length f and the camera shift M.

Chapter 3

STEREO VISION BASED AUTOMATED SOLDER BALL HEIGHT AND SUBSTRATE COPLANARITY INSPECTION

3.1 Introduction

For unrectified stereo images, the 3D structure can be estimated with the prior knowledge of stereo camera positions and enough feature correspondences of 2D points between stereo images. Stereo vision through camera calibration and feature detection has been widely used in various areas such as target recognition [47], vehicle navigation and obstacle detection [48] and robotics 3D reconstruction [49]. Sparse features such as corners and edges are usually detected using feature detectors including SIFT [21], SURF [22], etc. However, for stereo images that contain objects with less texture and features, these popular feature detectors fail to produce enough matching correspondences, thus the depth information of stereo images can not be estimated efficiently. For applications that has object with textureless regions, the stereo vision method is rarely used for 3D reconstruction and depth estimation. One of such applications is the industrial automated machine vision for product inspection and quality control. In this chapter, we propose a stereo vision method with novel feature detection method for the solder ball height and coplanarity estimation in industrial product inspection.

Defective solder joints on BGA (Ball Grid Array) can cause problems in semiconductor products, including non-wets and infant mortality resulting in failed parts. In order to reduce the potential for late detection of warped or defective parts resulting in potential added cost for a defective unit and escapee to a customer, the inspection of solder joints of BGA is an important process in manufacturing. The solder joint bonding ability and reliability is highly dependent on the uniformity of solder ball heights and coplanarity across the unit substrate. Non uniform solder ball heights can result in non-wets which cause connectivity failures and result in failed units. Warpage can also cause connectivity failures as well as result in infant mortality due to connectivity failures at joints with minimal or weak connectivity. Thus the inspection of solder ball heights is essential in the detection and identification of defects in a timely manner before defective units escape to the customer.

Numerous methods have been developed in recent years for solder ball height and coplanarity measurements. Traditional methods such as visual inspection and in-circuit functional tests are not able to analyze the solder ball layout and height information, and they are usually time-consuming and produce variable results. Nowadays, automated methods are developed to produce more reliable and accurate ball height and coplanarity results. These methods include 3D X-ray laminography [26], laser scanning [27], Moiré projection (Fringe Moiré and shadow Moiré) [50], confocal microscopy [28], shadow graph [51], machine vision methods [29–32] and hybrid methods combining structured light and machine vision [52].

A common factory floor tool for warpage inspection uses a model-based method which requires a-priori reference and calibration by a microscope tool tested on several hundred BGA units for each type of product. The sampling and testing procedure of the reference microscope tool is time-consuming. This factory floor tool assumes uniform ball height from the model height, and is not able to compute the absolute ball height for each solder ball. If the incorrect model ball heights are used, there will be mis-detection of warpage and potential for defects and escapees due to incorrect ball size not being detected. The machine vision method in [29] acquires images of the package using two cameras with directional lighting and, for each image, the ball peaks are located by determining the position of the brightest point in each solder ball region, However, the assumption that the ball peaks correspond to highest intensity points is not valid in practice as shown later in this chapter. In addition, the method of [29] just calculates the position of the ball peaks with respect to the image origin and not ball peak heights. And it assumes that the majority of ball peaks are co-planar, which implies that most of the balls have the same height, and uses this assumption to determine a linear planar transformation (homography) describing the relation between the ball peak positions in the two views. The computed transformation is then applied to the positions of ball peaks in one view. The deviation between the resulting transformed positions and the actual positions in the second view is computed and used to flag balls with large deviations as defective. Similar to the method in [29], the machine vision methods described in [30–32] use two area-scan cameras at different angle setups and different lighting conditions to calculate the solder ball height and coplanarity. But they all have several deficiency and drawbacks: the method in [30] assumes that the substrate points are lying on the same plane, and it calculates the individual ball height from the fitted substrate plane and an estimated average ball height value; the methods in [31] and [32] require a complex calibration process of the camera poses, multiple images for the whole package inspection, and more importantly, they all require a reference ball with known ball height in the field of view for each image. Thus these machine vision methods are not sufficient to detect the true ball height and coplanarity without any prior knowledge and not suitable for real-time computation.

The method in [52] combines structured light and machine vision methods to estimate the 3D shape of mirror surfaces in applications of modeling specular weld pool surfaces. This method projects a structured laser dot pattern onto the mirror surface, and uses three cameras in addition to non-reflective planes to capture the reflected laser patterns. The three captured laser pattern images are used to reconstruct a sparse set of 3D points on the measured surface through camera calibration and homography calculation. However, the method in [52] is used to reconstruct relatively large surfaces in the millimeter range, and it is not applicable to model micrometer surfaces as considered in our application. Additionally, the method of [52] assumes that the 3D surface to be reconstructed is convex and can be estimated from the sparse set of 3D points through a second-degree polynomial fitting. This assumption is not generally applicable to real-world solder ball surfaces.

Some of the existing automated ball height and coplanarity detection methods [26–28, 50, 51] are able to provide relatively more accurate results as compared to the aforementioned existing machine vision methods [29–32], but they usually require high-cost equipment and complicated setup, and the measuring speed is slow. The laser scanning method [27] uses a line-structured laser vision sensor to obtain the range image of the BGA package and determine the orientation of the BGA from the substrate surface region. Each individual solder ball is located from semi-circles in the line-scan profile, and the ball surface is modeled using Bezier curves. Then the ball height is computed from the modeled ball surface profile. Although the laser scanning method reconstructs the ball surface with relatively high accuracy, it suffers from slow speed and complicated setup and cannot be used as part of an in-line inspection system. The Moiré projection method [50] projects the periodic fringe patterns on the package surface and generates absolute ball height and coplanarity from the deformation of the projected waveforms on the solder ball packages. But the phase unwrapping in the Moiré projection methods can produce inaccuracy in the ball height results, and it is computationally intensive and time consuming. The motion control system and integrated workstation of the Moiré projection methods are usually expensive and require complex training. Another height inspection method [51] calculates absolute ball height from the shadowgraph of balls in the images generated by an oblique collimated light source. But this method is only used on wafers, and might not be usable for solder balls on BGA packages. The collimated light setup requires a number of expensive optical lenses. The size of the whole setup is large and is not suitable for in-line manufacturing ball inspection operations. Due to the set up complexity and limited execution speed, existing automated solder ball height and coplanarity measuring methods are not suitable for a real-time inspection process. Therefore, a reliable, fast, in-line ball height and coplanarity measurement method is needed for inspecting units undergoing assembly.

Existing stereo vision measurement techniques determine the height and depth of objects by detecting corresponding feature points in two views of the same scene taken from different viewpoints. The images in stereo matching research are usually taken from a natural scene or manmade objects, which have distinct features for each object in the scene for matching, such as color and gradient. There are various methods proposed for stereo matching [53–55], but a common issue with existing techniques is that they rely on the presence of edges, corners and surface texture for the detection of feature points. Therefore, these techniques cannot be applied to the measurement of solder ball height due to the textureless, edgeless, smooth surfaces of solder balls.

In this chapter, an automatic, stereo vision based, in-line ball height and coplanarity inspection method is presented. The method proposed in this chapter is computationally efficient compared to other image processing techniques for solder ball height detection and is shown to exhibit high accuracy, repeatability and reproducibility. Additionally, the imaging setup procedure and equipment of the proposed method is much simpler than other existing methods. The proposed method includes an imaging setup together with a computer vision algorithm for reliable, in-line ball height measurement. The imaging set up consists of two different area-scan cameras mounted at two opposing angles with ring lighting around each camera lens which allows the capture of two images of a semi-conductor package in parallel. The lighting provides a means to generate features on the balls which are then used to determine height. The computer vision algorithm consists of calibration of stereo cameras, segmenting individual balls, detecting substrate feature points and ball peak feature points in stereo views, triangulation of corresponding points and calculating ball height and substrate coplanarity. The camera parameters, including intrinsic parameters and extrinsic parameters, are calculated in the calibration process. The segmentation of each individual ball is achieved using histogram thresholding and a boundary circle-fitting algorithm. The substrate feature points are detected in the segmented individual ball region, and the ball peak feature points are determined by grouping points with the same intensity on the ball surface, which allows the formation of curves, also known as iso-contours, that are then matched between the two views. Finally, an optimized triangulation is performed to determine feature point depth and ball height, and coplanarity is calculated from the determined substrate depth.

The proposed ball height calculation method was tested on three different types of BGA products, which have different ball size, ball surface appearance, ball pitch and layout. The results are evaluated in a measurement capability analysis (MCA) procedure and compared with the ground-truth obtained by an existing laser scanning tool and an existing confocal inspection tool. The laser scanning tool and the confocal tool are primarily suitable for sampling measurements due to slower speed and lengthy calibration process. The accuracy of ball height of the proposed method is compared with the ball height of the laser scanning tool, and a correlation of 0.94 is achieved. The coplanarity of the BGA package is determined from the computed substrate depth results in the proposed algorithm. The results show that the proposed method is capable of calculating the ball height and warpage on BGA packages, and that the produced results are comparable to the results produced by other methods that require significantly more expensive equipment and complicated processing software.

This chapter is organized as follows. The stereo camera setup and camera calibration process are presented in Section 3.2. The proposed feature detection algorithm and ball height and coplanarity calculation algorithms are presented in Section 3.3. The experimental results and performance analysis with existing schemes are presented in Section 3.4. Conclusions are drawn in Section 3.5.

3.2 Imaging Setup and Camera Calibration

Since the size of solder balls on the BGA package is relatively small (the ball diameter is typically 200-400μm for different types of BGA packages), in order to obtain repeatable and reliable ball height results, the setup of the stereo cameras is required to be precise and stable when imaging such small-size objects. The imaging setup in this work uses two cameras mounted at two opposite angles with a ring light around each camera lens, which allows the capture of two images of a semi-conductor package in parallel. In our setup, we used two Adimec OPAL-8000 cameras and two AI standard working distance 3” LED white color ring lights. The BGA package is placed on a tray holder under both cameras’ field of view. The ring light around each camera generates straight light beams that shine on top of the solder ball surface. The surface of the solder balls is reflective to directional light due to the specular nature of ball surface. With the previously described setup, the ball peak points on the ball surface that have the same surface normal with that of the substrate surface will reflect the illumination into the camera [29], as shown in Fig. 3.1. Thus these ball peak points will appear bright in the captured images. Other points, whose surface normal vectors have a direction that is different from that of the normal of

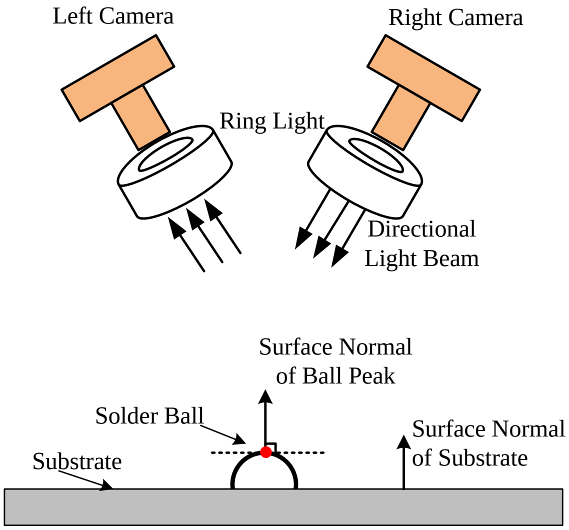


Fig. 3.1: Illustration of the Ball Peak Point Reflection in the Imaging Setup. It Should be Noted that the Ball Size was Enlarged for Illustration Purposes and that the Actual Ball Size is Significantly Smaller than Shown in the Figure.

the substrate surface, will reflect the directional light to other directions that cannot be captured by the CCD camera, and thus appear darker in the captured images. In the stereo images captured by two cameras at opposite angles, the bright regions near the center of each individual ball contain the ball peak area of solder balls. In the proposed camera and ring light setup process, the position of the BGA package and the lights are adjusted so that maximum brightness is achieved in an area surrounding the ball peaks in both views.

In stereo vision algorithms, the stereo camera parameters consist of intrinsic parameters, such as focal length and principal points, and extrinsic parameters, such as the rotation matrix and translation vectors between stereo cameras. In the pinhole camera model as illustrated in Chapter 2.1, the 3×4 camera matrix can be represented using the calibrated camera parameters [44]. In order to estimate the camera matrices of our stereo camera setup, the camera calibration method used in this chapter makes use of the MATLAB Camera Calibration Toolbox [56][57], which computes the camera parameters from the feature points of a planar calibration pattern in multiple views through a closed-form solution and a minimum mean-squared error (MSE) based nonlinear refinement procedure. The planar calibration pattern used in this chapter is a circular pattern as shown in Fig. 3.2. The small dark circles are printed on a ceramic board, are uniformly spaced and of the same size. A total of 25 pairs of images were captured with the calibration pattern board placed in 25 different positions. For each position, a pair of images is recorded using the left camera and right camera in the stereo setup. Then the camera calibration is implemented using the captured image pairs and a stereo calibration method [56] to obtain the camera parameters of the left and right cameras.

3.3 Proposed Automated Ball Height Inspection Method

In this section, we present more details about the proposed ball height detection algorithm which is capable of calculating accurate solder ball heights on different products. Fig. 3.3 shows the flowchart of the proposed ball height inspection method. The block diagram summarizes the steps of the proposed method including individual solder ball segmentation and matching, substrate feature point detection, ball peak feature point detection and matching, triangulation and ball height calculation and substrate coplanarity calculation. More details about ball segmentation, feature point matching, triangulation and coplanarity estimation are provided in Sections 3.3.1 to 3.3.4 below.

3.3.1 Individual Ball Segmentation and Matching

The accuracy of the matching process of ball feature points is one of the main factors that affect the accuracy of the final height results. In the feature matching

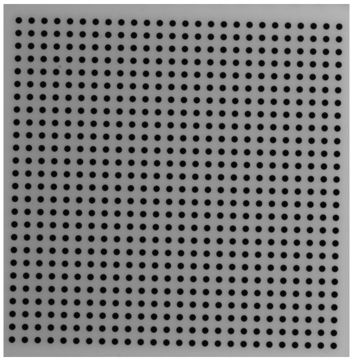


Fig. 3.2: Planar Calibration Pattern Board.

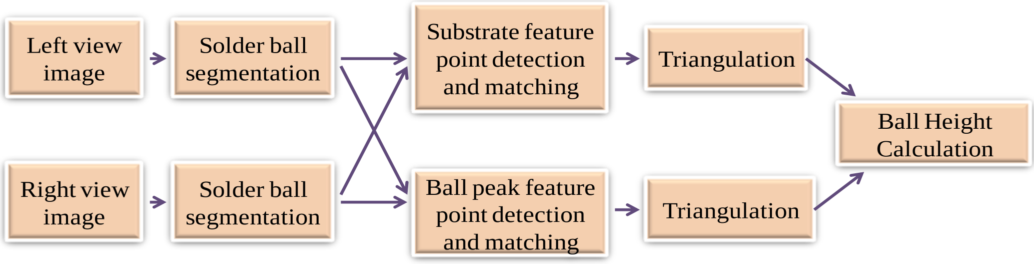


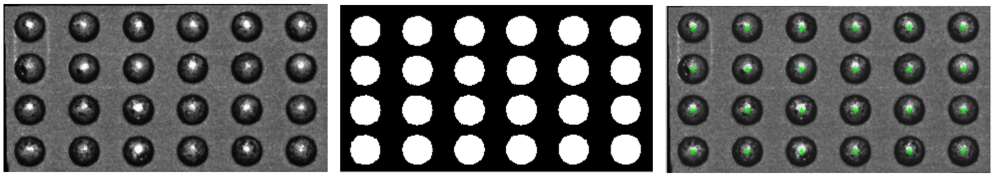
Fig. 3.3: Diagram of the Proposed Solder Ball Height Inspection Method.

process, the first basic step is to match the individual balls in the left view and right view correctly. For this purpose, the individual balls are segmented and labeled in a row-wise order. A ball with the same label number in the left and right views correspond to the same imaged physical ball. The segmentation of BGA images segments out two regions, the balls region and the substrate region. Several automatic thresholding methods were proposed to segment the gray-scale images into binary masks [52, 58]. The automatic segmentation method proposed in [58] uses two thresholds to model fuzzy edge regions that occur in metal transfer images, and assigns a probability to pixels in fuzzy regions to determine the object boundary. The thresholding method in [52] was proposed to segment the captured images of a structured laser dot pattern, which contain unevenly distributed background intensities. A two-dimensional band-pass filter is designed based on the known laser dot pattern structure and is used to filter the laser dot pattern image. A prefixed global threshold is then used to generate the segmentation mask. For our application, as shown in Fig. 3.4a, the captured BGA images exhibit strong boundaries between the individual balls and the background, which can thus be easily segmented using efficient histogram-based thresholding methods.

The segmentation method used in this work consists of an adaptive thresholding method based on histogram analysis [59]. The histogram of the image can be represented using two Gaussian distributions with different variance and different mean values. Using the automatic threshold calculation algorithm in [59] and morphological opening operations, the round-shape ball regions for each individual ball are segmented. The boundary of each individual ball mask is refined using circle fitting. An example of left view and right view images of a solder ball is shown in Fig. 3.4a, and the corresponding segmented ball mask is shown in Fig. 3.4b. In addition, for the same imaged physical ball, the highest intensity region containing or surrounding the ball peak occurs at different locations in the left and right views due to the opposite imaging angles of the left and right cameras, as shown in Fig. 3.5a. Therefore, template matching is performed for each matching pair of solder balls, using the segmented ball in one view as the template, in order to correct for this deviation. The bright region aligned images in the left view and right view are shown in Fig. 3.5b. This alignment enables a more accurate matching of the locations of the corresponding ball peaks as described in Section 3.3.2.

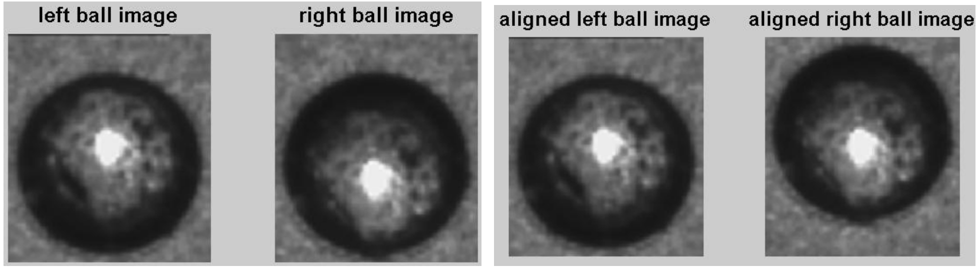
3.3.2 Feature Point Detection

After the balls are segmented and matched in the two camera views, ball peak points as well as substrate points need to be localized and matched in the left and right camera views in order to determine the ball heights. In order to measure the



(a) **(b)** **(c)**

Fig. 3.4: (a) An Example of a BGA Image in the Left View. (b) The Segmented Ball Mask of the BGA Image Shown in (a). (c) Substrate Feature Points Marked as Green Stars in the BGA Image Shown in (a).



(a) **(b)**

Fig. 3.5: (a) Original Ball Image in Left View and Right View. (b) Ball Image After Bright Region Alignment.

ball height, the goal is to find the bottom substrate point and the peak point of each solder ball from the features in the stereo 2D images. However, the main difficulty of the feature detection is that the surfaces of the solder balls are textureless and edgeless, which makes the popular feature detection algorithms, such as the Canny edge detector and the SIFT (Scale Invariant Feature Transform) [21] not suitable for finding the correct matching features. This is also the main reason why few stereo vision methods are used in the solder ball height inspection area.

The bottom points of each solder ball are the points that lie on the same surface as the substrate of the BGA package. Since the solder balls are placed on the substrate using the paste or fluxing technique, the circle-shape boundary where each individual solder ball touches the substrate can be used to generate the bottom points for each solder ball. In the 2D image captured by the CCD camera, the boundary between a solder ball and the substrate surface is the ball mask boundary of the considered ball. For each individual ball, the centroid of the boundary points represents the imaginary point lying under the solder ball surface on the substrate surface. Thus, for each pair of matching solder balls in the stereo images, the ball mask boundary and its corresponding centroid are calculated for each ball in the pair. The computed centroid points corresponding to a matched solder ball pair in the left and right views are taken as the matched feature points on the substrate. The substrate feature points of the BGA image shown in Fig. 3.4a are plotted as the green star points in Fig. 3.4c.

Compared to the substrate feature point detection, the feature point detection and matching for ball peaks is more intricate. In our proposed method, we exploit the fact that, in the captured pair of images, the ball peaks should belong to areas of high intensities due to the employed imaging and lighting set-up as described in Section 3.2. The lighting provides a means to generate intensity based features on the ball. According to the imaging setup procedure, since the directional illumination and the area scan camera are placed at the same angle but on opposite sides, the ball peaks belong or are surrounded by bright regions in the captured image pair.

In real-world manufacturing environments, not all solder balls have ideal surfaces; some ball surfaces may be slightly scratched or worn off. Therefore, in order to account for this, there are several different types of ball surfaces that are considered as part of this work. Fig. 3.6 shows different types of solder ball surfaces with different reflective characteristics. Fig. 3.6a, the normal ball surface exhibits a concentrated high-intensity round bright region around the ball peak. Another type of surface exhibits a bright region with a relatively lower intensity and larger diffused area than the one resulting from the normal surface, as shown in Fig. 3.6b. Some ball surfaces

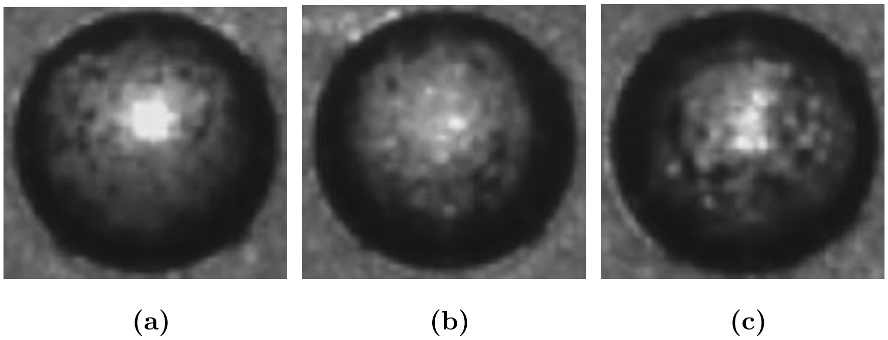
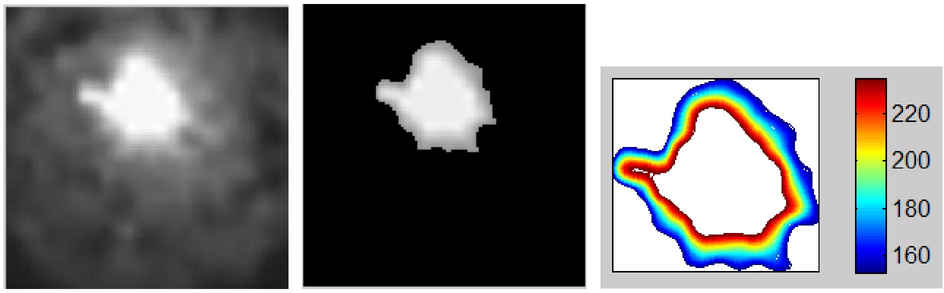


Fig. 3.6: Different Types of Ball Peaks. (a) Ball Peak with a Concentrated High-Intensity Round Bright Region. (b) Ball Peak with a Diffused Low-Intensity Bright Region. (c) Ball Peak with Several Separate Distinct Bright Regions.

result in reflected bright intensities forming several separated distinct bright areas instead of a single bright region, as shown in Fig. 3.6c. In addition, the imaged surface of a solder ball results in different bright regions in the left and right view due to local variations in the reflective surface characteristics of the solder ball as shown in Fig. 3.9c.1 and Fig. 3.9c.2. Due to the difference in the formed bright regions in the two views and to the variations in the reflective surface characteristics of solder balls, a robust matching algorithm suitable for the different types of solder ball surfaces is needed. In this chapter, an iso-contour based matching algorithm is proposed and is applied to detect the matching bright regions between stereo views.

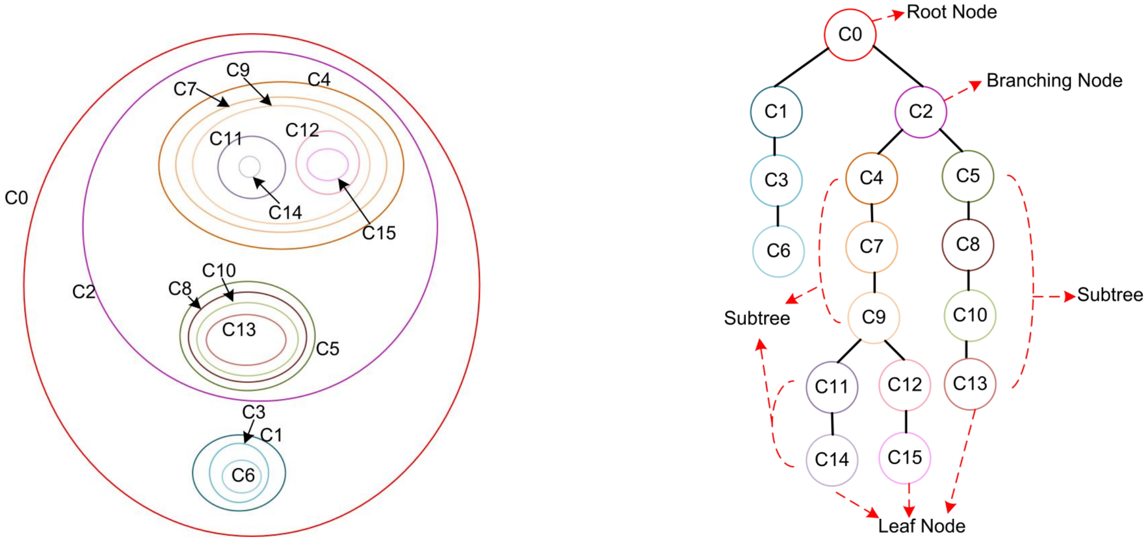
In our proposed method, points with the same intensity are determined on the ball surface and are grouped together to form curves of similar intensities, also known as iso-contours, which are then matched between the two views. For a given intensity, the iso-contour can be obtained by computing the locations of points having the considered intensity and connecting the located points together. In each individual ball region, multiple iso-contours are contained in a bright region intensity range. The lower threshold of the intensity range is determined using the adaptive thresholding method based on histogram analysis [59], and the higher threshold is the highest gray-scale level in the ball region. In order to obtain iso- contours with smooth curves, the discrete-domain image is transformed into an image on a fine dense grid approximating the continuous domain using bilinear interpolation. In this approach, the intensities of pixels at non-integer coordinates can be approximated from the surrounding pixels at integer coordinates, and the desired continuous iso-contour curves can be approximated. The iso-contours of various intensities in the bright region reflect the characteristic of the ball peak surface, such as the shape of the formed bright region and the structure of the bright region. The iso-contours inside a single concentrated bright region usually follow similar shapes as the bright region's boundary, and the iso-contours are nested from highest intensity to lowest intensity as shown in Fig. 3.7. Fig. 3.7c illustrates the iso-contours corresponding to the bright regions shown in Fig. 3.7b. In order to match the iso-contours in the left and right views, the iso-contours of each view are represented using a graph structure, which casts the iso-contour matching problem into a graph matching problem.

For a set of intensities ranging from a minimum value to a maximum value, the nesting relationship of iso-contours can be represented effectively using a tree graph structure, called the inclusion tree structure [60], as illustrated in Fig. 3.8. An iso-contour curve C_1 is defined as included inside another iso-contour curve C_2 if all the points along C_1 are located inside C_2. Equivalently, C_2 is said to enclose C_1. This inclusion relationship is defined mathematically as $C_1 \subset C_2$, if $C_1 \subset Int(C_2)$, where $Int(C)$ represents the interior region of the curve C according to the Jordan curve theorem [60]. As shown in Fig. 3.8b, each contour curve corresponds to a node in the inclusion tree structure. The outermost iso-contour corresponds to the root node (for example, contour C0 in Fig. 3.8. A branching node in the tree structure corresponds to an iso-contour that encloses two or more non-nested iso-contours (for



(a) **(b)** **(c)**

Fig. 3.7: (a) An Example of Solder Ball Bright Region. (b) Bright Region After Thresholding. (c) Enlarged Iso-Contour Map for the Right Region in (b).



(a) **(b)**

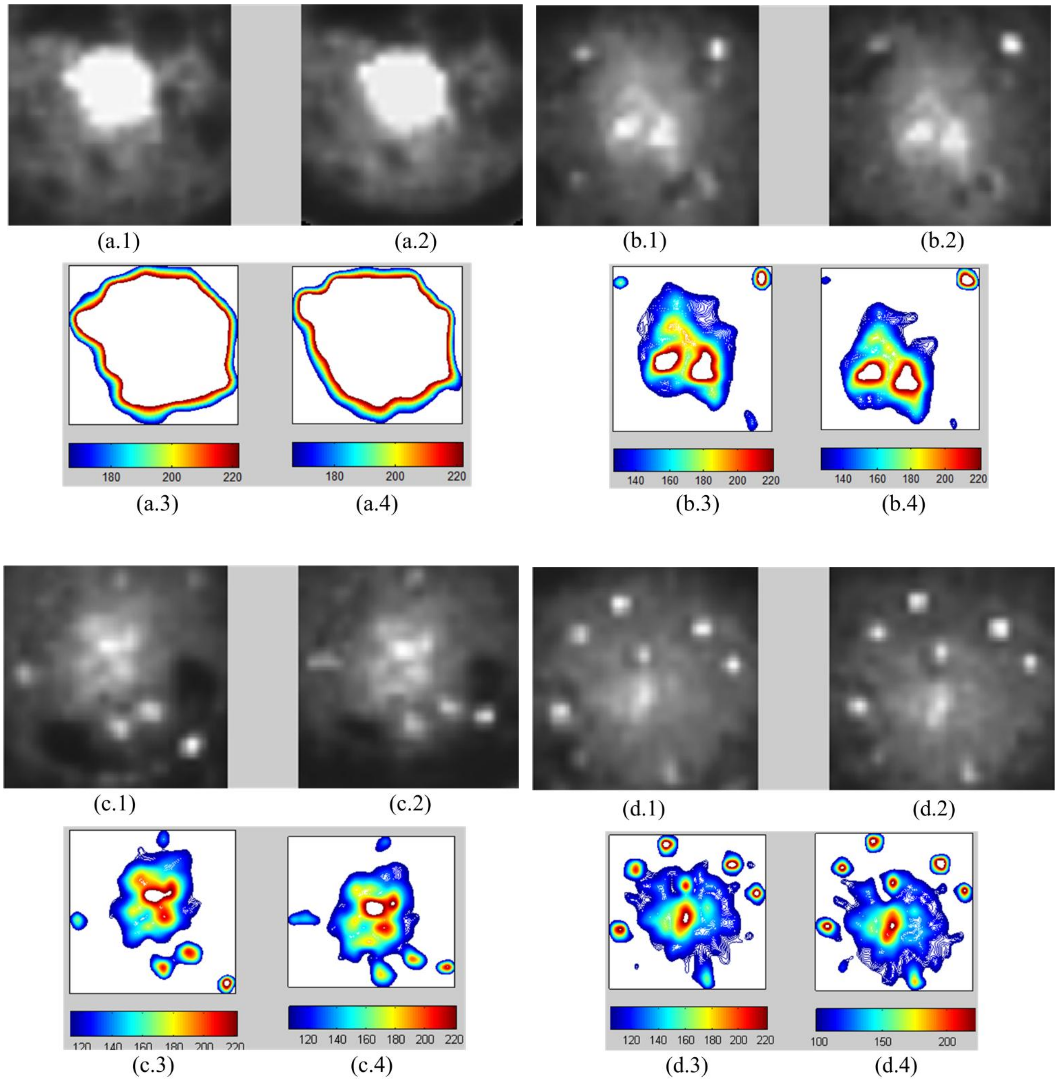
Fig. 3.8: (a) An Example of Contour Map. (b) Inclusion Tree of the Contour Map in (a).

example, contour C2 and C9 in Fig. 3.8), and the resulting branches are called each a subtree. The end nodes of the tree structure, also known as leaf nodes, correspond each to an innermost iso-contour (for example, contour C6, C13, C14 and C15 in Fig. 3.8). Fig. 3.9 illustrates different types of bright regions and their associated iso-contours. For example, in Fig. 3.9a, there is a single contiguous bright region and its corresponding iso-contours can be represented using a tree with a single branch.

Fig. 3.9b to Fig. 3.9d illustrate cases when there are several distinct bright regions and their associated iso-contours can be represented by a tree with multiple subtrees. It should also be noted that the number of subtrees representing the iso-contours in the left view can be different from the number of subtrees in the right view.

Once the tree structure is formed for each view, the iso-contour matching algorithm starts by detecting and removing the scattered outliers of sub-tree/tree regions in the left and right views' bright regions. This outlier region removal is done by locating the bounding box of iso-contour region and removing scattered bright regions near the bounding box. The bounding box of the bright region is determined as the bounding box of the root iso-contours with relatively large areas. If there is only one iso-contour in the root node of the tree structure, the bounding box is the bounding box of the root iso-contour. If multiple separate iso-contours exist at the root node level of the tree structure, the areas of each of these iso-contours are sorted in descending order, and the area ratio of the j^{th} area over the $(j+1)^{th}$ area is calculated. The first index of the area ratio that is larger than a threshold of 2 is detected, and all iso-contours in the sorted array before the threshold index are used to compute the rectangular bounding box enclosing these iso-contours. Once the bounding box of the bright region is detected, for each sub-tree/tree region, the distance between the centroid of the outermost contour and the centroid of the bounding box is calculated, and the sub-trees/trees regions that result in a distance that is greater than twice the standard deviation of all the computed distances are removed as outliers. This ensures the removal of the scattered bright regions with small areas.

After the removal of the outliers scattered bright regions, for each subtree/tree region in one view, a matching subtree/tree region is localized in the other view, and matching feature points are computed from the matching subtree/tree regions. A flowchart in Fig. 3.10 shows the main steps of the matching procedure.


(a.1) (a.2) (b.1) (b.2)

(a.3) (a.4) (b.3) (b.4)

(c.1) (c.2) (d.1) (d.2)

(c.3) (c.4) (d.3) (d.4)

Fig. 3.9: Examples of Different Iso-Contour Maps of Solder Ball Peak Regions. (x.1) Left Ball Peak Image. (x.2) Right Ball Peak Image. (x.3) Iso-Contour Map in Left. (x.4) Iso-Contour Map in Right. (x = a,b,c,d).

For each subtree/single-branch tree in the left view, referred to as reference subtree/tree, a set of candidate matching subtrees/single-branch trees in the right view is formed. The set of matching candidates is formed based on three features: the overlap ratio between the areas covered by the outermost contour of the considered subtree/tree in the left view and the candidate subtree/tree in the right view, the

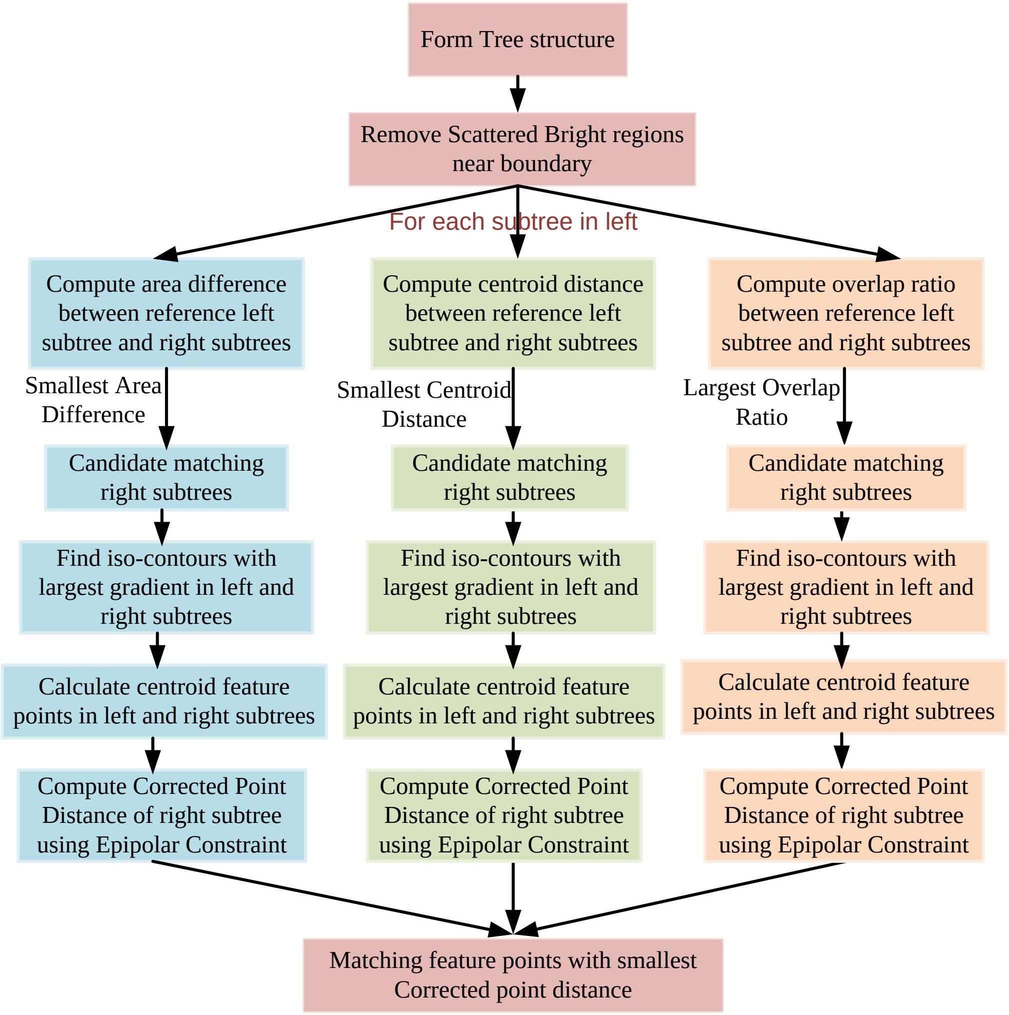

Form Tree structure
Remove Scattered Bright regions near boundary
For each subtree in left
Compute area difference between reference left subtree and right subtrees
Compute centroid distance between reference left subtree and right subtrees
Compute overlap ratio between reference left subtree and right subtrees
Smallest Area Difference
Smallest Centroid Distance
Largest Overlap Ratio
Candidate matching right subtrees
Candidate matching right subtrees
Candidate matching right subtrees
Find iso-contours with largest gradient in left and right subtrees
Find iso-contours with largest gradient in left and right subtrees
Find iso-contours with largest gradient in left and right subtrees
Calculate centroid feature points in left and right subtrees
Calculate centroid feature points in left and right subtrees
Calculate centroid feature points in left and right subtrees
Compute Corrected Point Distance of right subtree using Epipolar Constraint
Compute Corrected Point Distance of right subtree using Epipolar Constraint
Compute Corrected Point Distance of right subtree using Epipolar Constraint
Matching feature points with smallest Corrected point distance

Fig. 3.10: Flowchart of the Ball Peak Feature Point Detection.

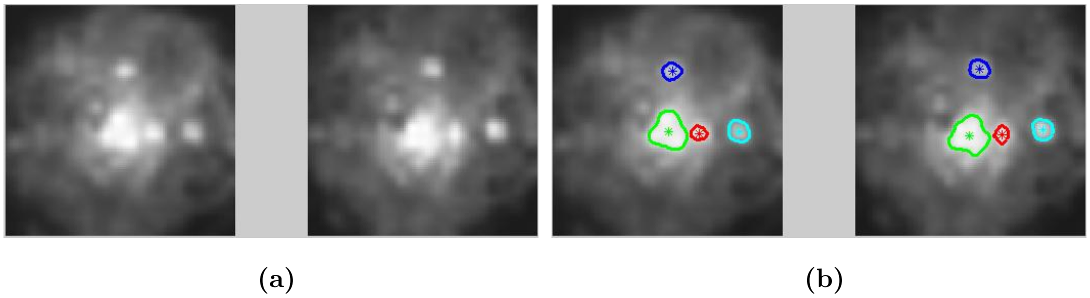


Fig. 3.11: (a) An Example of Ball Image in Left View and Right View. (b) Multiple Pairs of Matching Iso-Contours and Matching Centroid Points (Same Color Illustrates the Matching).

area difference and the centroid distance of the outermost iso-contour of the considered subtree/tree in the left view and the candidate subtree/single-branch tree in the right view. The right-view subtrees/single-branch trees with the largest overlap ratio, the smallest area difference or smallest centroid distance are selected as matching candidates. As a consequence, for each subtree/single-branch tree in the left view, there are at least one matching candidate and at most three matching candidates in the right view. If the formed set of candidates contains all the subtrees of a tree, an additional candidate representing the whole tree is also included as part of the formed candidate set.

Once the matching candidates of the subtree/tree region are formed, the next step is to locate the matching ball peak feature points in the matching subtree/tree regions in the left and right views. For this purpose, for each candidate matching pair of subtree/tree regions between the left and right view, the iso-contour with the largest average intensity gradient magnitude values in each view is chosen as the candidate matching iso-contour curve, and the centroid points of the matching iso-contours in both views are calculated as a candidate matching pair of feature points. Among the matching candidate feature point set, the best matching subtree/tree region is determined using the epipolar geometry between their relative locations. The details of the epipolar geometry will be discussed later in Section 3.3.3. The epipolar constraint between two corresponding 2D feature points $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ is represented using the fundamental matrix F as $\boldsymbol{x_2^T} F \boldsymbol{x_1} = 0$. Due to the presence of noise and pixel quantization error, the detected feature points may not satisfy the epipolar constraint. While fixing the feature point in the left view, the nearest point to the original feature point in the right view which satisfies the epipolar constraint can be located. The Euclidean distance of the corrected point and the original feature point in the right view can be used as a measure of the quality of the original matching feature points. If a pair of candidate matching points results in a large Euclidean distance for the corrected point, the candidate matching is problematic. For each subtree/single branch tree of the left view, among the candidate matching feature points in the right view, the matching feature point that results in the smallest corrected Euclidean distance is chosen as the best matching feature point, and the corresponding matching subtree/tree region is selected as the best matching subtree/tree region.

For multiple pairs of matching subtrees/trees, the corresponding multiple pairs of centroid points are used as the matching feature points for the considered ball peak between the left view and right view. An example of the matching iso-contour curves and centroids for multiple matching bright regions are shown in Fig. 3.11. These multiple pairs of centroid points are further used to calculate candidate ball height using triangulation as discussed in Section 3.3.3. For a given solder ball, candidate ball heights that deviate from other ball height values by twice of the standard deviation are eliminated, and the ball height value with the smallest corrected point distance among the remaining candidate ball heights is selected as the final solder ball height.

3.3.3 Triangulation, and Ball Height Calculation

Once the corresponding feature points of substrate and solder ball peak are located in the stereo images, the triangulation method is used to obtain the 3D reconstruction of these feature points. As illustrated in Fig. 2.3, ideally, in the 3D space, the intersection of the two lines, which are formed by connecting each of the matching 2D points and their corresponding camera centers, can be computed to get the corresponding 3D point in space. But due to the presence of noise and digitization errors, it is possible that the intersection of these two rays does not exist in the 3D space. In this chapter, the epipolar-based optimized triangulation [44] is applied to the matched pair of feature points in order to calculate the coordinates of the corresponding 3D points.

As the feature points are detected in the images according to local iso-contour information, the detected matching feature points might not satisfy the epipolar constraints between the stereo views due to noise and digitization error. In linear triangulation, the 3D points calculated using the 2D feature points that do not satisfy epipolar constraints will be inaccurate. Thus a corrected set of 2D matching feature points satisfying the epipolar constraint needs to be calculated near the original 2D feature points detected in images. The method in [44] uses a MSE-based method to localize a pair of corrected 2D feature points $\hat{\boldsymbol{x}}_{\boldsymbol{L}}$ and $\hat{\boldsymbol{x}}_{\boldsymbol{R}}$ that minimize the Euclidean distance between the corrected points and the original noisy 2D points $\boldsymbol{x}_{\boldsymbol{L}}$ and $\boldsymbol{x}_{\boldsymbol{R}}$ in two views, which can be represented mathematically as

$$\min\, d(\boldsymbol{x}_{\boldsymbol{L}}, \hat{\boldsymbol{x}}_{\boldsymbol{L}})^2 + d(\boldsymbol{x}_{\boldsymbol{R}}, \hat{\boldsymbol{x}}_{\boldsymbol{R}})^2 \text{ subject to } \hat{\boldsymbol{x}}_{\boldsymbol{R}}^{\boldsymbol{T}} F \hat{\boldsymbol{x}}_{\boldsymbol{L}} = 0 \tag{3.1}$$

According to the epipolar geometry introduced in Chapter 2.2.1, any pair of corrected points satisfying the epipolar constraint must lie on a pair of corresponding epipolar lines in the two images; alternatively, any pair of points lying on the corresponding epipolar lines will satisfy the epipolar constraint. Thus localizing corrected 2D points which minimize the Euclidean distance from the original points is equivalent to localizing a pair of corrected epipolar lines in both views that minimize the Euclidean distance between the original 2D points and the corrected epipolar lines in both views. This can be represented as

$$\min\, d(\boldsymbol{x}_{\boldsymbol{L}}, \hat{\boldsymbol{l}}_{\boldsymbol{L}})^2 + d(\boldsymbol{x}_{\boldsymbol{R}}, \hat{\boldsymbol{l}}_{\boldsymbol{R}})^2 \tag{3.2}$$

where $\hat{\boldsymbol{l}}_{\boldsymbol{L}}$ and $\hat{\boldsymbol{l}}_{\boldsymbol{R}}$ are the corrected epipolar lines satisfying the epipolar constraint. In order to minimize (3.2), firstly the corrected epipolar line in the left view is parameterized using a parameter t and the epipole $\boldsymbol{e}_{\boldsymbol{L}}$ calculated using the fundamental matrix F. Thus the epipolar line in the left view can be written as $\hat{\boldsymbol{l}}_{\boldsymbol{L}}(t)$. Secondly, using the fundamental matrix F, the corresponding epipolar line in the right view can be computed as $\hat{\boldsymbol{l}}_{\boldsymbol{R}}(t)$. Thirdly, the distance function in (3.2) can be expressed as a polynomial function of t, represented as $g(t)$. By computing the roots of the numerator polynomial of $g'(t)$, the solution of the parameter t is the root value which minimizes the polynomial function (3.2). Finally, using the calculated parameter t, a pair of corrected epipolar lines are solved for. In each view, the intersection point between the corrected epipolar line and the line going through the original 2D point and perpendicular to the corrected epipolar line is the corrected 2D feature point.

In this chapter, a simplified method to compute a corrected set of 2D feature points is used. Instead of computing a pair of corrected points near the original ones, the feature point in one view is fixed, and a corrected feature point in the other view near the original one is calculated. In order to locate the corrected set of 2D matching feature points at the sub-pixel level, both the left-view and right-view images are interpolated by an integer factor of 4 before feature point detection and matching. The goal of the optimized triangulation method is to fix the original 2D feature point in one view (left view), and to localize a corrected 2D feature points $\hat{\boldsymbol{x}}_{\boldsymbol{R}}$ that minimizes the Euclidean distance between the corrected points $\hat{\boldsymbol{x}}_{\boldsymbol{R}}$ and the original noisy 2D points $\boldsymbol{x}_{\boldsymbol{R}}$, in the other view (right view), which can be represented mathematically as

$$\min d(\boldsymbol{x}_{\boldsymbol{R}}, \hat{\boldsymbol{x}}_{\boldsymbol{R}})^2 \text{ subject to } \hat{\boldsymbol{x}}_{\boldsymbol{R}}^{\boldsymbol{T}} F \hat{\boldsymbol{x}}_{\boldsymbol{L}} = 0 \qquad (3.3)$$

Since the feature point in the left view is fixed, the corresponding epipolar line in the right view can be calculated using the feature point in the left view and fundamental matrix, as $\boldsymbol{l}_{\boldsymbol{R}} = F\boldsymbol{x}_{\boldsymbol{L}}$. The corrected point in the right view that minimizes the distance in (3.3) is the intersection point between the epipolar line and the line going through original 2D point and perpendicular to the epipolar line. The corrected pair of 2D feature points satisfies the epipolar constraint, and the intersection of the projected lines of 2D points in 3D space is ensured. Using the corrected matching feature points in both views and the calibrated camera projection matrix, the coordinates of the 3D point corresponding to the matched 2D points in the left and right images are computed using linear triangulation based on singular value decomposition (SVD) [61]. The Euclidean distance in (3.3) also reflects the accuracy of the matching between the original 2D feature points. If the corrected Euclidean distance is large, the original matching pair of feature points is problematic. Thus this corrected Euclidean distance is used as indicated in Section 3.3.2 to determine the best matching among multiple candidate matching sub-trees or single branch trees.

For each individual ball, the 3D point of the ball bottom on the substrate and the 3D points of ball peaks are calculated through triangulation. The Euclidean distance between the coordinates of each 3D ball peak point and corresponding 3D ball bottom point is calculated as the ball height value. As indicated previously in Section 3.3.2, for a given ball, it is possible to obtain multiple best matching pairs of ball peak subtree/tree regions. In this latter case, for the considered ball, multiple 3D ball peak points and corresponding ball heights are calculated, one for each best matching pair. Ball heights that deviate from other ball height values by twice of the standard deviation are eliminated, and the height value with the smallest corrected point distance in the remaining ball heights is selected as the solder ball height.

3.3.4 Substrate Coplanarity Calculation

The coplanarity of the BGA substrate is used to evaluate the warpage of the substrate surface. In the proposed method, coplanarity is calculated using the depth of substrate points corresponding to each individual ball. And by finding a rotation matrix and a translation vector that transform the substrate 3D points from the coordinate system of the left camera (with the origin point located at the camera center) to the coordinate system of the BGA package (with the origin point located at the top left corner point of the image package area), the depth values of the transformed substrate 3D points correspond to the coplanarity values of the BGA substrate. According to three-dimensional machine vision theory [44] and to the camera calibration method in [56], given the calibrated intrinsic camera matrix of the left camera, the 2D image points of a planar pattern on the left camera image plane and the 3D points that correspond to the same physical planar pattern with respect to the BGA package's coordinate system, the homography between 2D planar image points and planar 3D points can be calculated, and the rotation matrix and translation vector are computed by decomposing the obtained homography matrix.

The first step in the proposed method is to form a set of 3D planar points in the 3D space, and get the 3D coordinates of these points, $\boldsymbol{X_C}$ and $\boldsymbol{X_P}$ with respect to both the left camera's coordinate system and BGA package's coordinate system, respectively. Then, using the projection matrix of the left camera, P_L, and the 3D planar point's coordinates $\boldsymbol{X_C}$ with respect to the left camera's coordinate system, a set of 2D planar image points $\boldsymbol{x_C}$ on the left camera's image plane can be computed. Finally, the rotation matrix and translation vector can be computed using the 2D planar image points $\boldsymbol{x_C}$ and the set of corresponding 3D planar points' coordinates $\boldsymbol{X_P}$ in BGA package's coordinate system as described in more details below.

The procedure for warpage detection and quantification consist of the following: (1) As indicated in Section 3.3.3, for each individual ball, the 3D point of the ball bottom on the substrate and the 3D points of ball peaks are calculated through triangulation. The obtained 3D substrate points calculated with respect to the left-view camera (denoted as $\boldsymbol{X}_{C\_substrate}$) may not lie on the same plane due to warpage. In order to detect and quantify this warpage, a reference plane, with respect to which warpage is measured, needs to be determined. For this purpose, the coordinates (with respect to the left-view camera's coordinate system) of three boundary 3D substrate points in the imaged BGA package are used to form a plane. We chose the top-left, top-right and bottom-left boundary 3D substrate points for the plane calculation. For all other 3D substrate points, they are projected on this plane resulting each in a projected 3D point with the same x and y coordinates but with a z-coordinate (depth value) that is determined from the plane equation. Thus a set of 3D planar points $\boldsymbol{X_C}$ with respect to the left camera's coordinate system is formed using 3 boundary substrate points and all other projected 3D substrate points lying on the plane. The x and y coordinates of 3D planar points $\boldsymbol{X_P}$ with respect to the BGA package are formed using the same x and y coordinates of $\boldsymbol{X_C}$, and the z values are defined to be 0.

(2) The 2D image points $\boldsymbol{x_C}$ on the left-view camera's image plane corresponding to the 3D planar points $\boldsymbol{X_C}$ are calculated using the left-view camera's projection matrix P_C as $\boldsymbol{x_C} = P_L\boldsymbol{X_C}$, where $P_L = K_L[I, \bar{\mathbf{0}}]$, K_L is the camera intrinsic matrix, I is a 3×3 identity matrix, and $\bar{\mathbf{0}}$ is a 3×1 column vector with all zero elements.

(3) Finally, the 3×3 homography H between $\boldsymbol{x_C}$ and $\boldsymbol{X_P}$ is computed using the Direct Linear Transform (DLT) method [44], and $\boldsymbol{x_C} = H\boldsymbol{X_P}$. The rotation matrix R and translation vector $\boldsymbol{t}$ is computed by decomposing H [56]. Once R and $\boldsymbol{t}$ are obtained, the 3D substrate points $\boldsymbol{X}_{C\_substrate}$, calculated using triangulation with respect to the left camera's coordinate system, are transformed to 3D substrate points $\boldsymbol{X}_{C\_package}$ with respect to the BGA package's coordinate system, as $\boldsymbol{X}_{C\_package} = R^T\boldsymbol{X}_{C\_substrate} - R^T\boldsymbol{t}$. The origin point of the BGA package's coordinate system is taken to be the top-left corner point of the imaged solder ball package region. The depth values (z-coordinate value) of the transformed 3D substrate points

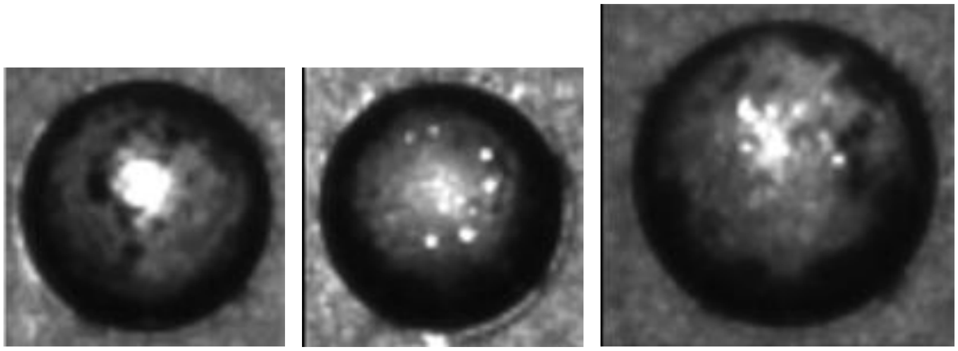


Fig. 3.12: Examples of Solder Balls of Three BGA Products. (a) Solder Ball of Product A. (b) Solder Ball of Product B. (c) Solder Ball of Product C.

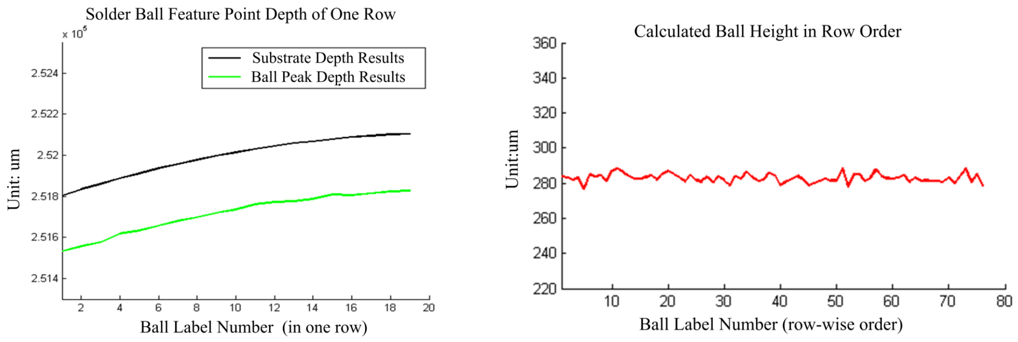
corresponding to each solder ball represent the coplanarity of the BGA substrate.

3.4 Experimental Results

The proposed method was applied to different Intel product lines: A, B and C. Each one of these product lines has different solder ball layout and different ball characteristics. The average ball height for product A, B, C is around 280μm, 280μm and 380μm respectively. Example images of solder balls, one from each of the three products, are shown in Fig. 3.12.

In order to evaluate the performance of the proposed method, the ball height and coplanarity results of the proposed algorithm were analyzed through the measurement capability analysis (MCA) procedure. The MCA procedure consists of three different aspects to prove that the proposed metrology is accurate, capable and stable under important parameters. These three analyses are accuracy, repeatability and reproducibility, and for each analysis, there is a standard evaluation metric to determine the capability of the metrology. The MCA procedure is conducted on both the ball height and coplanarity results, and details are described in the following sections.

In the MCA procedure, the ball height and coplanarity results of the proposed method are compared with those collected using two existing automated inspection tools: the laser-scanning tool and the confocal inspection tool. The laser-scanning tool scans the entire ball surface for each ball to calculate the true ball height. But the laser scanning process is quite time-consuming being extremely slow and is not suitable to measure all the testing ball samples due to the considerable amount of time it takes to profile each ball. Furthermore, laser-scanning does not provide a measure of coplanarity. As a consequence, in our MCA analysis, the ball height accuracy of the proposed method is compared with the ball height of the laser-scanning tool on a limited number of balls. On the other hand, the existing confocal inspection tool is able to profile the BGA package off-line in a shorter time than the laser scanning tool, and it can provide both ball height and coplanarity results in multiple measurements. But the confocal tool determines the solder ball height using circle-fitting techniques for ball peak detection, thus the obtained ball height values may not correspond to the true height. A detailed correlation analysis between the confocal tool ball height and laser-scanning ball height is presented in Section 3.4.1 showing that the confocal tool does not correlate well with the highly accurate laser scanning tool. As a consequence, in our MCA analysis, the ball height of the laser scanning tool is used as a reference for establishing the high accuracy of the ball height obtained using the proposed method, while ball of height of the confocal tool is used for establishing the high repeatability of the proposed method. In addition, the coplanarity results of the confocal tool are used for establishing the high accuracy and repeatability of the proposed method for coplanarity measurement and warpage detection.



(a) **(b)**

Fig. 3.13: (a) Calculated Ball Peak Depth and Substrate Depth with Respect to the Left Camera for One Row of Solder Balls. (b) Calculated Solder Ball Height Results in Row Order.

3.4.1 MCA on Solder Ball Height Results

For the solder balls in one row on the BGA package, the depth values of substrate and the depth values of ball peak feature points with respect to the left-view camera's coordinate system are plotted in Fig. 3.13a. Since both cameras are placed over the BGA package, the substrate depth values are larger (greater distance from the camera) than the ball peak depth values (less distance from the camera) when viewed from the left camera center. The ball height is obtained by calculating the Euclidean distance between the substrate 3D point and the ball peak 3D point. The ball height values of solder balls are plotted in a row-wise order, as shown in Fig. 3.13b.

In the accuracy analysis of the MCA procedure, the ball height results calculated using the proposed method are compared with the height results of laser-scanning inspection tool on product C on 17 balls. This is because product C usually produces more variance and inaccuracy in ball height than the other two products, and if the accuracy of product C is capable, it also proves the capability of the other two products. The evaluation metric of accuracy analysis is the R-squared value, which is

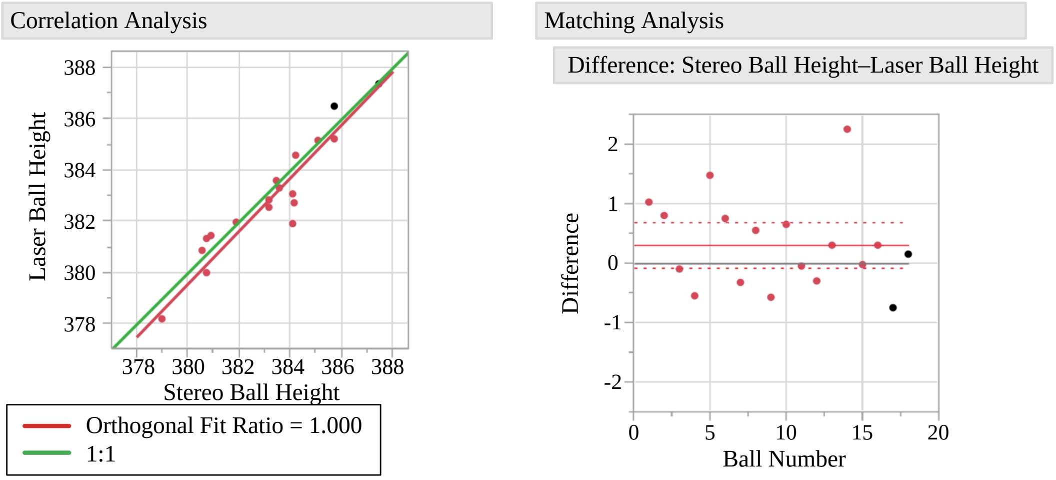


Fig. 3.14: Accuracy Analysis (Correlation Analysis and Matching Analysis) of Ball Height Results of the Proposed Method and the Laser-Scanning Method.

the square of the correlation between two comparison results. If the R-squared value of the ball height between the proposed method and laser-scanning tool is larger than 0.75, the proposed method is accurate for automated inspection. The correlation and ball height difference results of the ball height values between the proposed method and laser-scanning tool are shown in Fig. 3.14. Using the statistics analysis software JMP 7.0, the correlation value is 0.94 and the R-squared value is 0.8853, indicating that the proposed method not only satisfies but significantly exceeds the accuracy criterion. On the other hand, since the confocal tool uses circle-fitting techniques on the ball surface peak detection, the ball height detected by the confocal tool does not reflect the true ball height. In fact, the correlation of ball height between the confocal tool and the laser-scanning tool is 0.3, which does not prove high accuracy. This is the reason why the ball height of the proposed method is compared with that of the laser-scanning tool for accuracy analysis.

In the repeatability analysis of the MCA procedure, the ball height results calculated using the proposed method are compared with the height results of the confocal

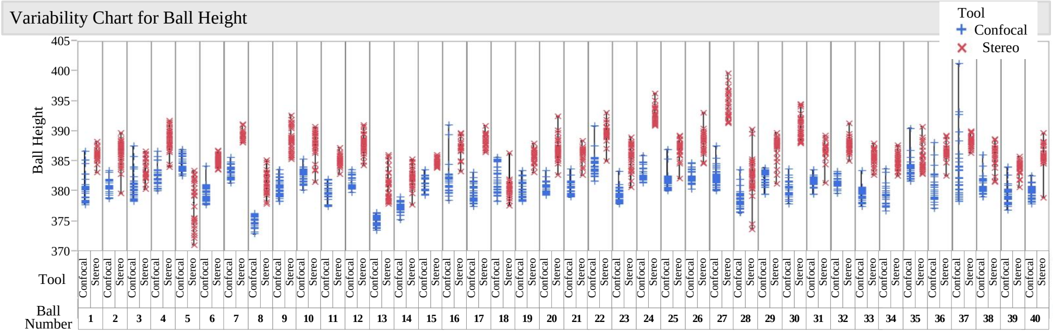
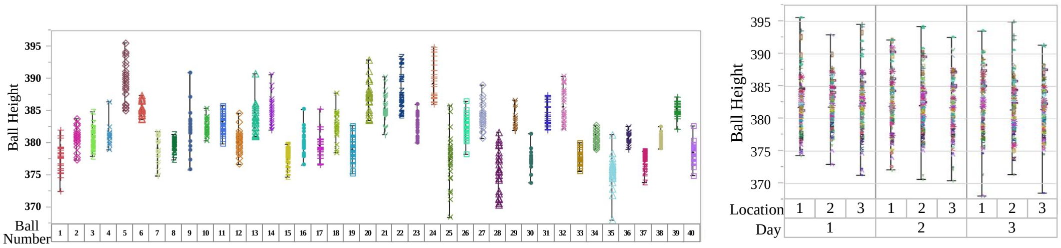


Fig. 3.15: Repeatability Analysis of Ball Height Results of the Proposed Method (Red Star) and the Confocal Method (Blue Circle). Ball Height Results of 40 Balls from Product C and 30 Measurements for Each Ball are Plotted.



(a) **(b)**

Fig. 3.16: Reproducibility Analysis of Ball Height Results of the Proposed Method (Symbols with the Same Color Represent the Ball Height of the Same Ball). Ball Height Results of 40 Balls on a BGA Package of Product C and 27 Measurements for Each Ball are Plotted. (a) Ball Height Plotted by the BGA Part Number and Ball Number. (b) Ball Height Plotted by Different Locations and Days.

inspection tool on all three products due to the established high repeatability of the confocal tool. The objective of repeatability testing is to determine whether the tool's "inherent" variation is acceptable and stable in the short term. For each product, a BGA package is randomly picked and imaged for 30 times. In the 30 repeated measurements, the BGA package is pick-up and re-fixtured in a short time on the same day. The total numbers of balls tested on the package of each product line A, B and

C are 76, 151 and 156, respectively. The evaluation metric of repeatability analysis is the P/T ratio, which is represented as

$$P/T = \frac{6\sigma_{ms}}{USL - LSL} \times 100\% \tag{3.4}$$

where σ_{ms} is the mean variance of 30 ball heights of each ball among all balls measured, and $USL-LSL$ is the tolerance range of the solder ball height in the pre-defined spec limits. The tolerance range of product A and B is 100 μm, and of product C is 120 μm. If the P/T ratio is below 20%, the repeatability of the proposed method is proved. The ball height results for tested solder balls from Product C in 30 measurements of the proposed method and the confocal method are plotted in row order in Fig. 3.15. Due to space limitation and for improved readability, repeatability results for ball heights are shown only for the first 40 of the 156 tested balls. The mean variance values of product A, B and C are 1.31μm, 1.67μm and 1.58μm, respectively. Using (3.4), the P/T ratio values of the proposed method on product A, B and C are 7.9%, 10% and 7.91% respectively. The P/T ratio values of the confocal method on product A, B and C are 5.5%, 7.6% and 8%. Both the proposed method and the confocal method have the P/T ratio well under 20%, and this proves the repeatability of the proposed method.

In the reproducibility analysis of the MCA procedure, the ball height results calculated using the proposed method are analyzed on different products under different affecting factors. The objective of reproducibility testing is to determine whether the total measurement variation is acceptable under different metrology factors. In the reproducibility testing of ball height detection method, three major affecting factors are tested: day, rotation and translation of BGA package in the field of view, and repeat of each test. The testing plan consist of 2 different products (products A and C), 40 balls on each product, 3 days, 3 rotated and shifted package locations in the camera field of view, and 3 repeats for each placement. Thus for each ball on each product, 27 measurements are taken and tested. The evaluation metric of reproducibility analysis is the P/T ratio, and if the P/T ratio is smaller than 30% in reproducibility testing, the proposed automatic tool is capable. The ball height results for the tested solder balls of product C in 27 measurements of the proposed method are plotted in row order in Fig. 3.16. Reproducibility results for ball heights are shown in Fig. 3.16 for only the first 40 of the 156 tested balls due to space limitation. The mean variance values σ_{ms} of product A and product C are 1.94μm and 1.90μm, respectively. The P/T ratio value of the proposed method on products A and C are 11.6% and 9.5%, respectively, which are significantly lower than the threshold of 30%. Thus the reproducibility analysis of the proposed ball height detection method meets and exceeds the reproducibility criterion.

From the above three MCA procedures on the proposed ball height detection method, the accuracy and stability of the proposed ball height detection method is proved, and the proposed method is capable to be used as an automated ball height detection tool.

3.4.2 MCA on Substrate Coplanarity Results

The coplanarity of the BGA substrate is used to evaluate the warpage of the substrate surface. In the considered BGA package coordinate system, the depth values of the substrate point represent the coplanarity of the package. The 3D plot of the 3D substrate points over 4 solder ball rows on the package for product C is shown in Fig. 3.17 using the proposed method (red star). For comparison, Fig. 3.17 also shows the 3D substrate points obtained using the confocal tool (blue dot). Similar results were also obtained for other tested products. The 3D substrate points are plotted in Fig. 3.17 with respect to the BGA package's coordinate system (with the

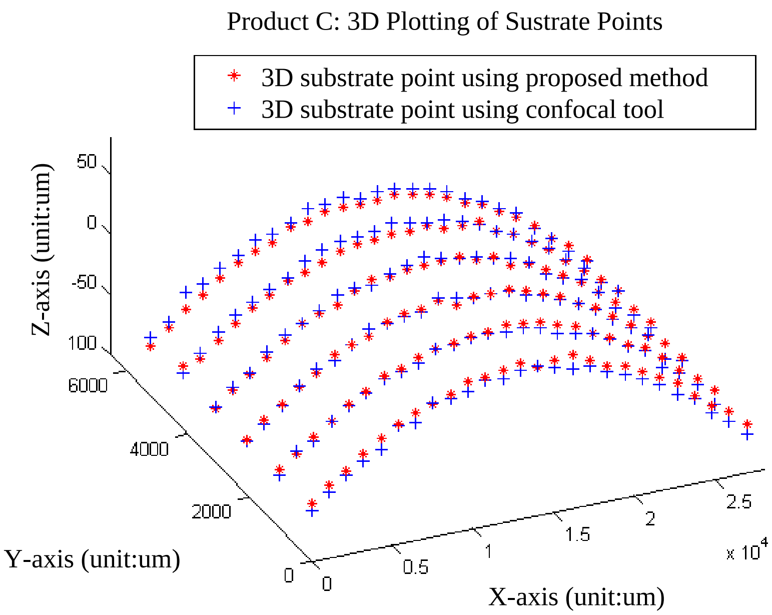


Fig. 3.17: Plotting of Substrate Points with Respect to Package Coordinates and Coplanarity Comparison with Confocal Results for Product C.

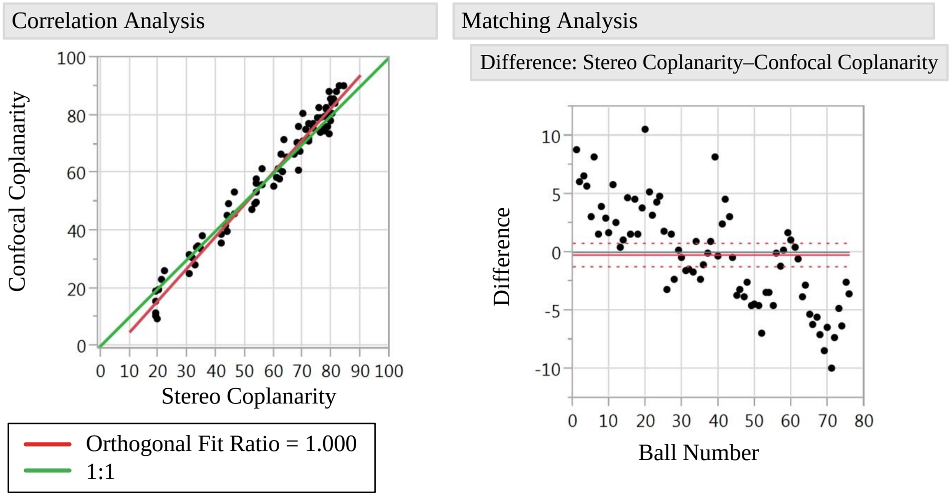


Fig. 3.18: Accuracy Analysis (Correlation Analysis and Matching Analysis) of Coplanarity Results of the Proposed Method and the Confocal Method.

origin point located at the top left corner point of the image package area). As seen in Fig. 3.17, the depth values of the ball substrate points are following a curved-shape plot, which reflects the warpage of the BGA package.

Similar to the MCA on solder ball height, the MCA procedure including accuracy, repeatability and reproducibility are tested on the coplanarity results, and compared with the confocal coplanarity results. In the accuracy analysis, the proposed method is tested on product A with 76 balls, product B with 151 balls and product C with 156 balls. The correlation and difference results in terms of coplanarity values between the proposed method and the confocal tool are shown in Fig. 3.18 for product C. The correlation values for coplanarity between the proposed method and the confocal tool on product A, B and C are 0.984, 0.979 and 0.985, respectively. Thus the R-squared values for product A, B and C are 0.9692, 0.9583 and 0.9701, respectively. Since the R-squared values for all three products are significantly larger than the threshold 0.75, the accuracy of the proposed coplanarity algorithm is proved.

In the repeatability analysis, the proposed method is tested on the same packages and balls as those used for the coplanarity accuracy analysis, and repeated for 30 measurements in a short time on the same day. The coplanarity results for the tested solder balls of product C corresponding to 30 measurements using the proposed method and the confocal method are plotted in row order in Fig. 3.19. For clarity and due to space limitation, the repeatability results for coplanarity are shown for the first 52 balls of the 156 tested balls in Fig. 3.19. The mean variance values σ_{ms} of product A, B and C are 0.8μm, 1.08μm and 0.61μm, respectively. The P/T ratio values on product A, B and C are 4.8%, 6.4% and 3%, respectively, for the proposed method, and the P/T ratio values of the confocal method is 5.5%, 7.6% and 8% respectively. For coplanarity repeatability, the P/T ratio of the proposed method is lower than that of the confocal tool. Since all the P/T ratios are significantly less than the threshold 20%, the high repeatability of the proposed coplanarity method is established.

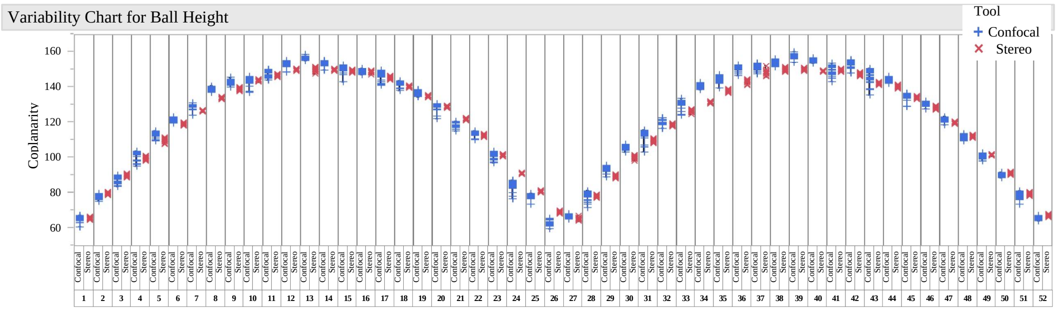
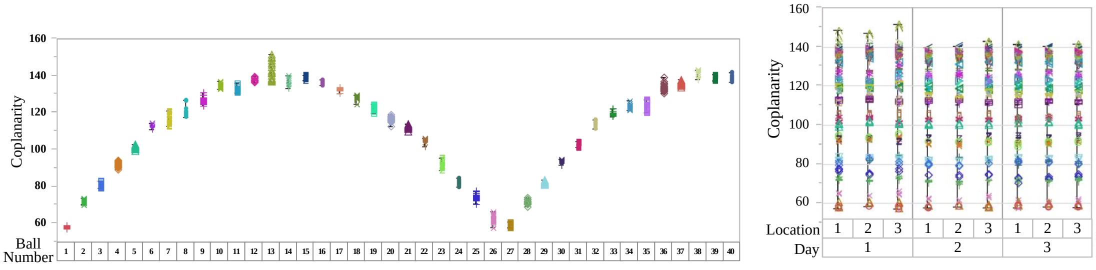


Fig. 3.19: Repeatability Analysis of Coplanarity Results of the Proposed Method (Red Star) and the Confocal Method (Blue Circle). Coplanarity Results of 52 Balls (2 Rows of Balls) on Product C and 30 Measurements for Each Ball are Plotted.



(a) **(b)**

Fig. 3.20: Reproducibility Analysis of Coplanarity Results of the Proposed Method (Symbols with the Same Color Represent the Coplanarity of the Same Ball). Coplanarity Results of 40 Balls on a BGA Package of Product C and 27 Measurements for Each Ball are Plotted. (a) Coplanarity Plotted by the BGA Part Number and Ball Number. (b) Coplanarity Plotted by Different Locations and Days.

The reproducibility analysis on the coplanarity results are tested using the same measurements as that of the reproducibility analysis on the ball height results. The coplanarity results for the tested solder balls on product C corresponding to 27 measurements using the proposed method are plotted in row order in Fig. 3.20. For clarity and due to space limitation, the reproducibility results for coplanarity are shown for the first 40 balls of the 156 tested balls in Fig. 3.20. The mean variance values σ_{ms} of product A and C are 0.82μm and 1.36μm, respectively. The P/T ratio values on product A and C using the proposed method are 4.9% and 6.8%, respectively, which are all significantly less than the threshold 30%. Thus the reproducibility of the proposed coplanarity method is validated.

3.5 Conclusion

In this chapter, a robust automatic in-line solder ball height detection scheme is presented to allow automated measurement and inspection of solder ball height. The proposed method is fully automated and can benefit the manufacturing process by measuring ball height and package coplanarity accurately. The proposed method has been implemented in software and was deployed on a standalone PC using the 2D images obtained from a simple imaging setup consisting of two cameras and directional ring lights. The proposed method can accurately and consistently compute solder ball heights and package coplanarity. The computed solder ball heights exhibit a very high correlation with the state-of-the-art expensive laser scanning technology and have been shown to be repeatable and reproducible. Compared to the existing ball height and coplanarity inspection tools, the proposed method has a low computational complexity and enables real-time in-line ball height and warpage inspection during manufacturing.

Chapter 4

SPARSE LOCALLY ADAPTIVE COST AGGREGATION FOR STEREO MATCHING

4.1 Introduction

Disparity estimation using a stereo image pair has been thoroughly studied in computer vision. The stereo image pair is taken with two cameras of the same scene at two slightly different positions. The stereo matching algorithms aim at estimating the dense correspondence between stereo images. Surveys comparing different stereo matching techniques in the last two decades can be found in [33, 62, 63]. Generally, there are two broad categories of stereo matching algorithms: global methods and local methods. Global algorithms generally define an energy model with different constraints, such as smoothness assumption and uniqueness assumption, and solve the energy model using global optimization techniques such as belief propagation [34, 35], dynamic programming [37], simulated annealing [64] or graph cut [36]. Global algorithms usually generate more accurate disparity results than local methods but the main drawback of global algorithms is that they are iterative and they incur a high computational complexity and are thus not suitable for real-time applications due to the slow speed and high memory requirements. On the other hand, local stereo matching approaches obtain the disparity map based on measuring correlations of local neighboring window pixels. According to [33], local stereo matching consists of four steps: matching cost computation, cost aggregation, disparity selection and disparity refinement. The local stereo matching methods used to be less accurate compared to global methods, but they are much faster and more suitable for real-time applications as compared to the global methods. Recent local stereo matching methods [40, 42, 65–74] use edge-preserving filters such as the bilateral filter and guided image filter in the cost aggregation step to generate more accurate disparity maps that outperform many global optimization based algorithms. Other methods use adaptive support regions to generate a more robust disparity map. Thus local stereo matching methods have been attracting more attention recently. However, in the most recent Middlebury version 3 stereo benchmark evaluation database, only 6 out of 54 submitted methods are local stereo methods, and there are no local stereo methods in the top 25 ranks, because of the various scene types and occlusion challenges in the Middlebury version 3 database. Thus there is a need to develop an efficient local stereo system that is robust to variations in stereo scenes.

In this chapter, we propose a novel Sparse Locally Adaptive Cost aggregation (SLAC) local stereo matching method. The proposed SLAC-based local stereo method achieves a higher accuracy than existing state-of-art local methods without an increase in the computational complexity. Different from most local stereo methods, except [41], that compute the cost aggregation at all disparity levels, the proposed SLAC method consists of a fast initial cost aggregation stage followed by a refined cost aggregation that is only performed over a sparse subset of disparities. In the proposed method, the cost aggregation is performed in a locally adaptive manner by adapting the support region to the local image intensity and structure. In order to reduce outlier disparity values that correspond to mis-matches, a novel sparse disparity subset selection method is proposed by assigning a significance status to candidate disparity values, and selecting the significant disparity values adaptively. An adaptive guided filtering method using the sparse disparity subset for refined cost aggregation and disparity calculation is demonstrated. Mismatched pixels in the disparity map are further refined through a sparse localized support-region-based propagation and occlusion handling. For this purpose, a novel disparity post-processing method using a multi-directional weighted propagation with a disparity and smoothness penalty is proposed. The proposed stereo matching algorithm is tested on the Middlebury version 2 and version 3 stereo evaluation benchmark images and the KITTI 2015 stereo benchmark images. We demonstrate the effectiveness of using the proposed sparse disparity subset selection method and adaptive sparse guided filtering in improving the accuracy of the disparity results.

The rest of this chapter is organized as follows. After an overview of related work in Section 4.1.1, the proposed local stereo matching method is described in Section 4.2. Section 4.3 presents the experimental results and analysis on the Middlebury version 2 and version 3 benchmark stereo dataset and the KITTI 2015 stereo benchmark dataset. Finally a conclusion is presented in Section 4.4.

4.1.1 Related Work

Adaptive Cost Weight: In local methods, cost aggregation is important in disparity estimation because it helps in removing the influence of possible noise during similarity measurements between pixels of stereo views. The cost aggregation is usually performed by summing up the (weighted) matching costs in local support windows under the assumption that the pixels in the local support region belong to the same object patch and have similar disparities. Recent local methods perform cost aggregation using locally adaptive filtering techniques with a region of support and weights that adapt to the local image characteristics. Bilateral filters are used in several popular stereo matching methods such as [65–69]. The cost in the local support region is weighted exponentially using the spatial and color differences between each surrounding pixel and the middle pixel. The main drawback of bilateral filtering methods is that the non-linear pixel-wise support weight computation consumes a large amount of time and computation load. This procedure is not amenable to be speeded up by integral image techniques. The method in [75] proposed to compute the adaptive weights using the geodesic distance to produce better performance at object borders. But the computation complexity of [75] is still high. Another method in [76] computes the adaptive weights using the information permeability and propagation method. The computation complexity is less as compared to using the geodesic distance. Using a guided filter [77] for stereo matching was first proposed in [70, 71], and was adopted in several stereo matching approaches [40, 42, 72–74]. The guided filter has better filtering effect near edges and depth discontinuities. Furthermore, the computational complexity of the guided filter is more suitable for real-time applications because of the linear computation and the speed-up implementation using integral image methods. The local multipoint filtering method in [40] extends the square-support region guided image filter to a more generalized multipoint filter using a zero-order and a first-order filtering model. The method in [42] proposed adaptive guided filtering based on the original guided image filter but with an adaptive kernel size. Recent proposed methods in [78, 79] combine the guided filtering based aggregation with semi-global optimization to achieve a better refined disparity result.

Adaptive Support Region: Besides the adaptive support weight methods such as bilateral filtering and guided filtering, the adaptive support window methods can also improve the cost aggregation accuracy, thus benefiting the disparity estimation. Instead of using fixed size rectangular windows, the size and the shape of the support window could be varied for each pixel, according to the color information in the local neighborhood. The shiftable window approach is proposed in [80] and used in [81–83] to select a proper square window size in a set of fixed-size windows for each pixel. This helps to handle the discontinuity regions. Other methods [84, 85] seek to determine the optimum non-rectangular window by optimizing over a large class of “compact” windows. The method in [86] proposed to use polygons with varying sizes to represent the arbitrarily shaped support region of each pixel. However this method does not result in an efficient image representation for cost aggregation since it requires more parameters for polygon representation. Another adaptive support window approach called the cross-based support region method is firstly proposed in [1], and has been adopted in several recent stereo matching methods [42, 72, 87–89]. The cross-based support region determines an arbitrary-shape support region using four arm lengths in four directions surrounding each pixel: left, right, up and bottom. This method handles the variable shape near depth discontinuities accurately and the cost aggregation is more efficient compared to other adaptive support window methods. The method in [40] further combines the cross-based support region with guided filtering to perform local multipoint filtering more efficiently as compared to the original guided image filtering.

Cost Aggregation Complexity Reduction: In local stereo matching methods, the most time-consuming step is the cost aggregation. It is important to reduce the computational complexity of cost aggregation for real-time purposes, while retaining the accuracy of the estimated disparities. Recently several local stereo matching schemes were proposed to reduce the computation load and the aggregation redundancy. A multiscale approach is used in [83, 90] to reduce the disparity search range using smaller support windows on the coarser image scale. A cost volume filtering method using salient subvolumes in [91] utilizes the SIFT [21] matching features to select the salient regions in each cost volume for cost volume filtering instead of filtering the full-size cost volume. This method achieves comparable results as compared to filtering the whole cost volume while accelerating the computation process. The integral image technique [92, 93] is used to accumulate the matching cost more efficiently. The cost aggregation methods based on variant support regions in [1, 40, 42] compute two integral images in the horizontal and vertical directions respectively, and the aggregated cost for each pixel is obtained by subtracting the corresponding accumulated costs in the integral images. The method proposed in [41] explores to reduce the redundancy and improve the computation efficiency in the disparity search range using a disparity subset determined jointly from the disparity histogram based on a fast box-filtered cost volume and a sampling technique in a square support region, and conducts a refined bilateral-filter-based cost aggregation using only the selected disparity candidates. But the fast box-filtered cost volume computation is noisy and inaccurate, and erroneous disparity values that correspond to mis-matches can result in inaccurate aggregated cost, which in turn can affect the disparity accuracy. Another drawback of the method in [41] is that they use the bilateral adaptive weight for refined cost aggregation with a relatively high computational complexity.

4.2 Proposed Local Stereo Matching Method

The proposed local stereo matching method follows the main steps of local stereo matching, including cost computation, cross-based support region calculation, fast and coarse cost aggregation, disparity subset selection, cost volume aggregation on the selected disparity subset using adaptive guided filtering, sparse localized support-region-based propagation, disparity calculation and disparity refinement. A block diagram of the proposed local stereo matching method is shown in Fig. 4.1. Details about each component are presented in the following subsections.

4.2.1 Cost Computation

The matching cost computation is the first step of stereo disparity estimation algorithms and it is vital to the final disparity results. The cost value is a per-pixel dissimilarity measurement between pairs of corresponding pixels in the left and right

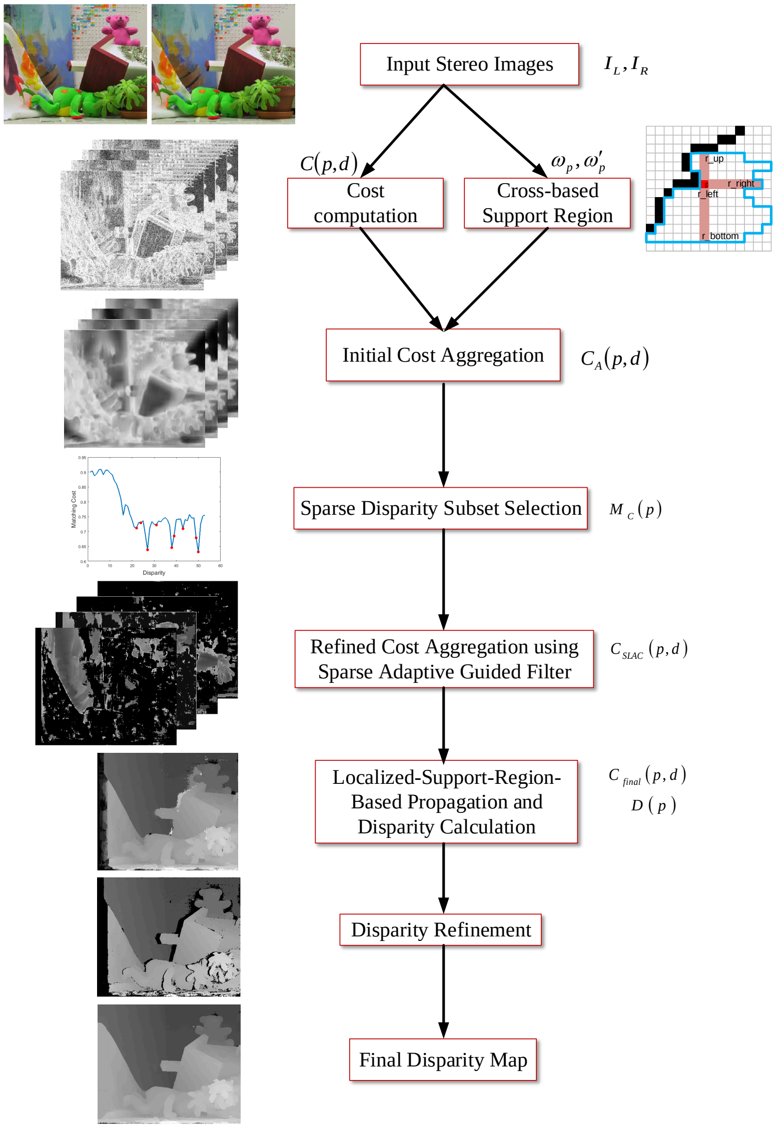


Fig. 4.1: Block Diagram of the Proposed Local Stereo Matching Method.

stereo images. Let $D = (d, 0)$. For pixel $p = (x, y)$ at disparity d, the cost value $C(p, d)$ is derived using pixel p in the left image and pixel $p - D = (x - d, y)$ in the right image. The cost volume is formed using the cost values over all pixels at all disparity values. Various cost measures were proposed for stereo matching algorithms, including absolute intensity difference (AD) [94], squared intensity differences (SD) [95–97], gradient-based measures [71, 98, 99], Birchfield and Tomasis's sampling-insensitive measure (BT) [100], mutual information [101, 102], normalized cross correlation (NCC) [103] and census [104, 105]. A comprehensive review of different matching costs is presented in [106]. Combining cost measures tend to improve the overall disparity accuracy, and different cost measures are combined and analyzed in several recent stereo matching approaches [71, 72, 107]. The census measure was shown to produce reliable dissimilarity cost and several modifications based on the census measure have been proposed [72, 105, 108]. In our stereo matching framework, we generate the matching cost by combining three measures: the Birchfield and Tomasi's sampling-insensitive measure (BT) [100], absolute horizontal gradient difference [71] and census on image intensity [104]. The Birchfield and Tomasi's sampling-insensitive measure (BT) is an essential cost measure to handle the color or gray-scale variations in the local region and is more robust to image sampling compared to the absolute intensity difference measure [100]. The Birchfield and Tomasi's sampling-insensitive measure (BT) is combined with the absolute intensity difference to enhance the robustness of the cost measure to illumination changes at depth discontinuities. The census measure encodes the local image structures with a relative ordering on pixel intensities, and was shown to be more robust to radiometric changes and image noises. The three cost measures can be represented mathematically as described below.

Suppose the pixel in the left view is denoted as $p = (x, y)$ and the disparity level is d, then the corresponding point in the right view is $p_R = p - D$, where $D = (d, 0)$. The Birchfield and Tomasi's sampling-insensitive measure (BT) of point p at disparity level d is computed as [100]:

$$C_{BT}(p, d) = \frac{1}{3} \sum_{i \in [r,g,b]} \min\Big(\bar{d}(p, p_R, I_L^i, I_R^i), \bar{d}(p_R, p, I_R^i, I_L^i)\Big) \tag{4.1}$$

where $\bar{d}(p, p_R, I_L^i, I_R^i)$ is the dissimilarity measure of how well the intensity at p in the left view fits into the linearly interpolated region surrounding p_R in the right view. $\bar{d}(p_R, p, I_R^i, I_L^i)$ is the dissimilarity measure of how well the intensity at p_R fits into the linearly interpolated region surrounding p. I_L^i denotes the color intensity value in the i^{th} RGB color channel in the left view, and I_R^i denotes the color intensity value in the i^{th} RGB color channel in the right view. According to [100], $\bar{d}(p, p_R, I_L^i, I_R^i)$ is computed as follows:

$$\bar{d}(p, p_R, I_L^i, I_R^i) = \max\Big(0, I_L^i(p) - I_{max}^{R\_i}, I_{min}^{R\_i} - I_L^i(p)\Big) \tag{4.2}$$

where

$$\begin{cases} I_{max}^{R\_i} = \max\Big(\frac{1}{2}\big(I_R^i(p_R) + I_R^i(p_R - (1,0))\big), \frac{1}{2}\big(I_R^i(p_R) + I_R^i(p_R + (1,0))\big), I_R^i(p_R)\Big) \\ I_{min}^{R\_i} = \min\Big(\frac{1}{2}\big(I_R^i(p_R) + I_R^i(p_R - (1,0))\big), \frac{1}{2}\big(I_R^i(p_R) + I_R^i(p_R + (1,0))\big), I_R^i(p_R)\Big) \end{cases} \tag{4.3}$$

Similarly, $\bar{d}(p_R, p, I_R^i, I_L^i)$ is computed as follows:

$$\bar{d}(p_R, p, I_R^i, I_L^i) = \max\Big(0, I_R^i(p_R) - I_{max}^{L\_i}, I_{min}^{L\_i} - I_R^i(p_R)\Big) \tag{4.4}$$

where

$$\begin{cases} I_{max}^{L\_i} = \max\Big(\frac{1}{2}\big(I_L^i(p) + I_L^i(p - (1,0))\big), \frac{1}{2}\big(I_L^i(p) + I_L^i(p + (1,0))\big), I_L^i(p)\Big) \\ I_{min}^{L\_i} = \min\Big(\frac{1}{2}\big(I_L^i(p) + I_L^i(p - (1,0))\big), \frac{1}{2}\big(I_L^i(p) + I_L^i(p + (1,0))\big), I_L^i(p)\Big) \end{cases} \tag{4.5}$$

The absolute horizontal gradient difference is defined as

$$C_{GD}(p, d) = \big|\nabla_x I_L\big(p\big) - \nabla_x I_R\big(p - (d, 0)\big)\big| \tag{4.6}$$

where $\nabla_x I(p)$ is the gradient value of the gray-scale intensity in the x direction at pixel p. $I_L(p)$ and $I_R(p)$ are the gray scale intensities at a pixel p in the left and right view, respectively.

To compute the census of image intensity, the gray-scale intensity at a point p in the left view is compared with the gray-scale intensity at other points in a local square window centered at p. A binary function representing the relationship between the intensity at the point p and p_n is defined as

$$\xi(p, p_n) = \begin{cases} 1, & \text{if } |I(p)| < |I(p_n)| \\ 0, & \text{otherwise} \end{cases} \tag{4.7}$$

The census transform of point p is a binary sequence formed by the concatenation of the binary function $\xi(p, p_n)$ for all neighboring p_n in the $n \times n$ square window centered at p, in a row-wise order. It can be expressed as

$$CENSUS(p) = \underset{p_n \in N_p}{\otimes} \xi(p, p_n) \tag{4.8}$$

where N_p is the $n \times n$ square window centered at p, and $\otimes$ is the concatenation operation in row-wise order. The census measure C_{CEN} between the point p in the left view and $p - (d, 0)$ in the right view is defined as the Hamming distance of two binary census strings of p in the left view and $p - (d, 0)$ in the right view, represented as

$$C_{CEN} = \text{Hamming}\Big(CENSUS^{\,left}\big(p\big), CENSUS^{\,right}\big(p - (d, 0)\big)\Big) \tag{4.9}$$

The final cost measure at each pixel is obtained as a combination of the above cost measures. However, each of the above cost measures does not carry the same significance, in the sense that increments in these cost measures do not correspond to equal increments in the perceived dissimilarity. Furthermore, an increment in a cost measure is not linearly proportional to the increment in the perceived dissimilarity, but can be related to the perceived dissimilarity through an exponential psychometric function [109, 110] as follows:

$$P(c, \lambda) = 1 - \exp\big(-\frac{c}{\lambda}\big) \tag{4.10}$$

where c is the cost measure and λ is a normalization parameter that depends on the corresponding cost measure. From (4.10), it should be noted that $0 \leq P(c, \lambda) \leq 1$, and thus $P(c, \lambda)$ provides a normalization of the cost measures so that they can be combined properly.

The final cost measure is obtained as a weighted linear combination of the normalized cost measures as follows:

$$C(p, d) = \alpha \cdot P(C_{CEN}, \lambda_{CEN}) + \beta \cdot P(C_{AD}, \lambda_{AD}) + (1 - \alpha - \beta) \cdot P(C_{GD}, \lambda_{GD}) \quad (4.11)$$

where α, β are weights for different cost components, and $0 < \alpha < 1$, $0 < \beta < 1$, $0 < \alpha + \beta < 1$.

4.2.2 Modified Cross Support Region

In the cost aggregation step, for each pixel, the cost of the neighboring pixels are aggregated in the neighborhood to reduce the matching ambiguities and generate a more reliable cost volume. The neighborhood pixels form the support region of each center pixel. It is desirable to determine the support region around a pixel p such that pixels in that region would correspond to 3D points belonging to the same object or surface as p and would thus have similar disparities. In local stereo matching algorithms, determining a proper support region for each pixel is essential to the cost aggregation step, and has a direct effect on the cost volume reliability and disparity accuracy. Several methods proposed various adaptive support region schemes to aggregate the matching cost more robustly [1, 80, 84, 86]. But the methods in [80, 84] relied on the square support regions which are not adaptive near disparity continuities. The method in [86] proposed to use polygons with varying sizes to represent the arbitrarily shaped support region of each pixel. However this method does not result in an efficient image representation for cost aggregation since it requires more parameters for polygon representation. The cross-based support region proposed in [1] is efficient to represent the adaptive support region, but it uses a fixed threshold to determine the support region arm length. Based on the cross-based support region method originally proposed by [1], we propose a locally adaptive cross-based support region calculation method that adapts to the local image characteristics by incorporating the variance of the local color change at each pixel in the color similarity threshold.

According to [1], the input color image is firstly smoothed using a 3×3 median filter to suppress the image noise. The adaptive support region of each point is determined using four cross arms in orthogonal directions based on the color similarity. The four cross arms are in the left, right, up and bottom directions. The arm length in each direction is determined by searching the largest span where all pixels covered by the span have similar colors as compared to the center pixel. For each pixel p, the cross-based support region is formed by merging the horizontal arms of all the pixels lying on the vertical arms of p, or merging the vertical arms of all the pixels lying on the horizontal arms of p. An example of the cross-based support region is illustrated in Fig. 4.2.

The original cross-based support region of [1] uses the maximum color difference in the RGB channels as the color similarity measure, and sets a fixed threshold to determine if the two pixels have a similar color. This method cannot adapt to the local characteristics of the visual content. For example, in highly textured regions with high color variations, the threshold should be higher to generate a larger support region containing the texture pixels. In textureless regions, the threshold should be relatively smaller in order to cover similar color pixels without crossing the edge boundaries.

Instead of using a fixed threshold to determine color similarity as in the method [1], in our proposed method, for each pixel p, the maximum color difference among RGB channels between p and a pixel p_i in the arm span is thresholded using an adaptive

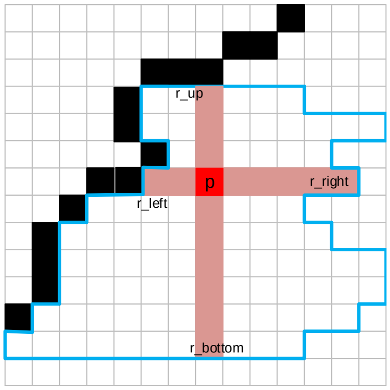


Fig. 4.2: An Example of the Cross-based Support Region of a Pixel. The Black Pixels Represent the Object Boundary. The Red Pixel is the Anchor Pixel p. For Each Pixel in the Vertical Arm Span, the Corresponding Horizontal Arm Lengths Are Determined to Form the Support Region. The Blue Contour Represents the Cross-based Support Region of p.

threshold determined by the local color standard deviation in the neighborhood of the pixel p_i.

In the horizontal arm calculation, the modified color similarity indicator function for the horizontal arm lengths $\delta(p_1, p_2)$ is defined as

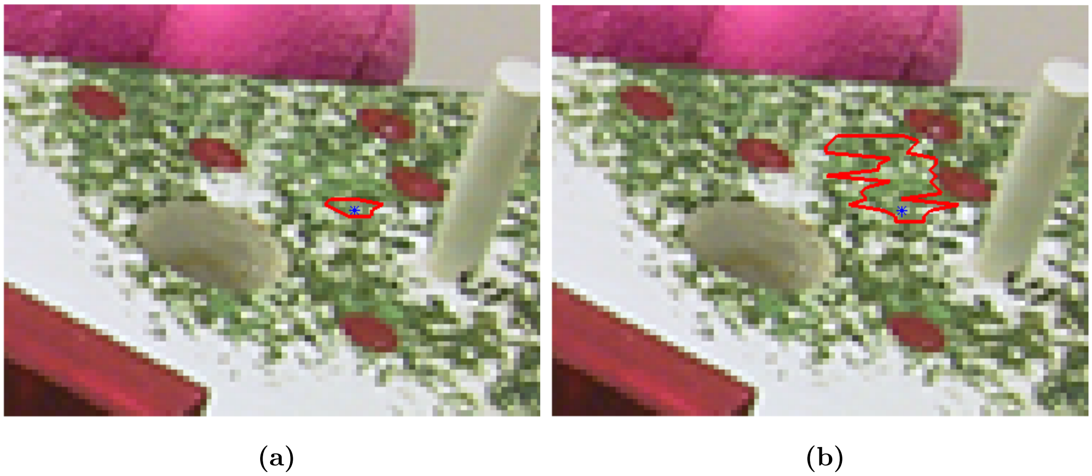
$$\delta(p_1,p_2)=\begin{cases}1, & \max\limits_{c\in[r,g,b]}\left(|I_c(p_1)-I_c(p_2)|\right)\le T_C\\ 0, & \text{otherwise}\end{cases} \tag{4.12}$$

and

$$T_C = c_1 \cdot \sigma_{Local\_H}(p_2) + c_2 \tag{4.13}$$

where c_1 is a scaling factor and c_2 is a bias term. $\sigma_{Local\_H}(p)$ is the local color change standard deviation in a neighborhood of pixel $p = (x, y)$ in the horizontal direction, and it is computed as follows:

$$\sigma_{Local\_H}(p)=\sqrt{\frac{1}{4}\sum_{k=-2}^{1}\left(\max_{i\in[r,g,b]}\left(|I_i(p+(k,0))-I_i(p+(k+1,0))|\right)-\mu_{Local\_H}\right)^2} \tag{4.14}$$



(a)

(b)

Fig. 4.3: The Boundary of the Support Region of a Considered Pixel (Blue Cross) in the Teddy Image is Shown by the Red Contour Surrounding That Pixel. (a) Original Cross-based Support Region [1] of a Pixel in the Teddy Image. (b) Modified Cross-based Support Region Generated by the Proposed Method of a Pixel in the Teddy Image.

where μ_{Local_H} is the mean value of the local color difference and is given by

$$\mu_{Local\_H} = \frac{1}{4} \sum_{k=-2}^{1} \max_{i \in [r,g,b]} \Big(\big| I_i(p + (k,0)) - I_i(p + (k+1,0)) \big| \Big) \tag{4.15}$$

The vertical arms are determined by calculating the color standard deviation in the vertical direction in a similar manner.

The proposed modified cross-based support region method adjusts in a local adaptive manner the thresholds based on the local structure and is thus able to better represent pixels belonging to the same textured regions as compared to the original method in [1]. This is illustrated in Fig. 4.3 which shows that the proposed method results in a cross support region that covers more relevant neighborhood pixels in the texture regions.

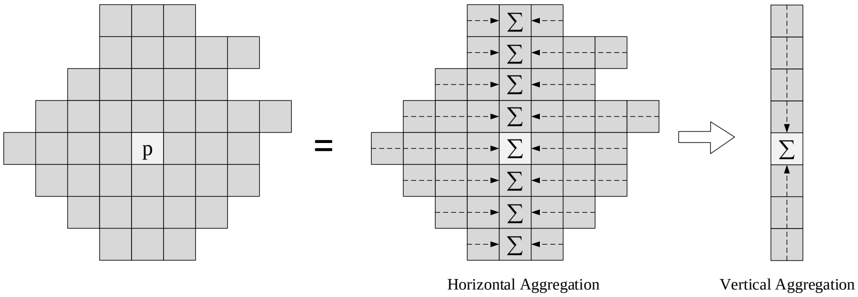


Fig. 4.4: Illustration of Cost Volume Aggregating over the Cross-based Support Region. The Cost is Aggregated First in the Horizontal Direction, and Then in the Vertical Direction.

4.2.3 Fast Coarse Cost Aggregation

After computing the per-pixel cost volume $C(p, d)$, and the adaptive cross-based support regions for all pixels, a fast cost volume aggregation is conducted to generate an initial coarse aggregated cost volume $C_A(p, d)$ for all pixels at all disparity levels. The proposed fast coarse aggregated cost volume $C_A(p, d)$ is less accurate than the refined cost aggregation using a guided filter in terms of disparity estimation, but it is faster and less noisy than the per-pixel cost calculated in Section 4.2.1. The coarse cost volume is used in Section 4.2.4 to select a sparse subset of disparity candidates with higher matching likelihood. Once the disparity subsets are estimated for all pixels, a more accurate adaptive cost aggregation is computed using the guided filter only on the disparity subset, as discussed in Section 4.2.5.

For each pixel p at disparity level d, the fast coarse aggregated cost $C_A(p, d)$ is computed as the summation of per-pixel cost $C(p, d)$ in p's support region. We use the cross-based support region ω_p as described in Section 4.2.2 for aggregation. As in [1], the cost is aggregated efficiently using two orthogonal 1-D integrations in the horizontal direction followed by the vertical direction. An illustration of orthogonal aggregation is shown in Fig. 4.4.

4.2.4 Disparity Subset Selection

As stated in Section 4.1.1, most existing edge-preserving filter based stereo methods exhaustively compute a single aggregated cost volume for all disparity values, and determine the disparity maps from the aggregated cost volume. For each pixel, the aggregated cost values are calculated for all integer values in the disparity range $d \in [0, D_{\max}]$, and the disparity value corresponding to the smallest cost value is selected as the disparity at this point. For each pixel, the matching costs with high matching values usually correspond to mis-matched disparity between stereo views. Thus there is potentially no need to calculate the aggregated cost at the mis-matched disparity levels, and this will also reduce the computational complexity in the cost aggregation step. Furthermore, it is stated in [41] that unnecessary pixel candidates with high matching cost at some disparity levels may contaminate the cost aggregation process for the neighborhood pixels, increasing the disparity ambiguity. In order to improve the computation efficiency and reduce the cost aggregation redundancy, it is possible to aggregate the matching cost using a compact representation of the per-pixel matching cost in a smaller disparity search range. Unlike the method in [41] that considers only the local minima of the cost function to form the disparity subset, we propose to adaptively select both the local minima and admissible disparity candidates with relatively small matching cost near local minima for each pixel. This is because disparity candidates with relatively small matching cost can also provide highly matching probabilities in the cost aggregation. Additionally, for each pixel, we incorporate the disparity subset candidates of the neighborhood support region pixels into the disparity subset calculation. This has the potential to reduce the noise effect in the initial cost volume estimation, and avoid missing some potential disparity candidates.

In our disparity subset calculation procedure, the initial fast and coarse aggregated cost volume $C_A(p, d)$ calculated in Section 4.2.3 is used to generate the disparity candidates corresponding to relatively lower cost values for each pixel. In the disparity subset selection, both local minima and admissible disparity candidates with relatively small matching cost are considered. For each pixel p, we assign a significance status to each of the disparity values in the disparity range $[0, D_{\max}]$, and the disparity candidates which are labeled as significant form the disparity subset.

Suppose we select a number of $N_{sub} = N_s \cdot D_{\max}$ sparse disparities candidates to generate the disparity subset $M_C(p)$, and a minimum number of M $(M < N_{sub})$ admissible disparity candidates with relatively small matching cost are enforced in the disparity subset. N_s is the ratio of the disparity subset size with respect to the total disparity range $D_{\max}$.

Firstly, for each pixel p, the initial aggregated cost $C_A(p, d)$ in the whole disparity range is normalized to $[0, 1]$, denoted as $C'_A(p, d)$, and local minima points whose cost value is smaller than 0.6 form a local minima disparity candidate set, denoted as D_{LM}. The number of selected local minima points in D_{LM} is denoted as N_{LM}.

Secondly, in order to assign a significance status to each disparity candidate, we consider the following two conditions:

(1) if the number of selected local minima N_{LM} is less than $N_{sub} - M$, all the disparity values in D_{LM} are marked as significant disparity candidates. Except for the marked significant disparity candidates, the cost values of remaining disparity candidates are sorted in ascending order, and the disparity values corresponding to the first $N_{sub} - N_{LM}$ smallest cost values are marked as significant disparity candidates;

(2) if the number of selected local minima N_{LM} is larger than $N_{sub} - M$, the cost values of local minima disparities in D_{LM} are sorted in ascending order, and the disparity values corresponding to the first $N_{sub} - M$ smallest elements are marked as

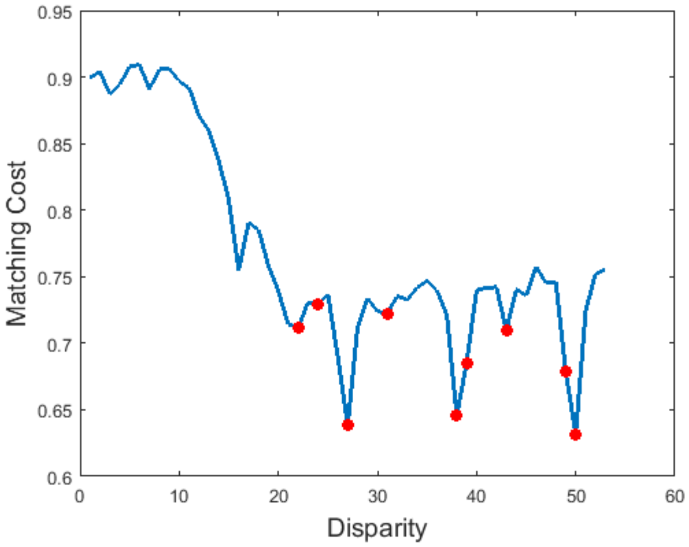


Fig. 4.5: Disparity Subset Selection From the Coarse Aggregated Cost Volume. The Red Dots Represent the Selected Disparity Candidates Including Local Minima and Admissible Disparity Candidates with Relatively Small Matching Cost.

significant disparity candidates. Except for the marked significant disparity candidates, the cost values of remaining disparity candidates are sorted in ascending order, and the disparity values corresponding to the first M smallest cost values are marked as significant disparity candidates.

Finally, the final disparity subset $M_C(p)$ for pixel p consists of all the significant disparity candidates. After the disparity subset is estimated for each pixel, we further analyze the disparity candidates of the neighborhood pixels, and add possible disparity candidates from the neighborhood. For each pixel p, we go through the disparity subset $M_C(q)$ of pixels in the neighborhood support region $q \in \omega_p$, and form a disparity histogram $H_p(d)$ of the disparity candidates in the neighborhood pixels. The histogram is normalized to $[0, 1]$. For a disparity value d_1, if $H_p(d_1)$ is larger than a threshold of $0.5/N_{sub}$, this disparity candidate is likely to correspond to true matches, thus d_1 is added to the disparity subset $M_C(p)$ of p if $M_C(p)$ does not contain d_1

already.

By selecting the disparity values of both local minima and admissible disparity candidates with relatively small matching cost and from the neighborhood pixels, a more reliable disparity subset is generated compared to only selecting the local minima candidates. An example of the initial aggregated cost $C_A(p, d)$ and the selected disparity subset (marked as red dots) is illustrated in Fig. 4.5.

4.2.5 Refined Cost Aggregation using Sparse Adaptive Guided Filter

As stated earlier, since the per-pixel matching cost is noisy and produces unstable disparity values by selecting the minimum cost in the isolated per-pixel cost among the disparity range, a refined cost aggregation is computed using the adaptive support region and disparity subset based on the per-pixel matching cost volume computed in Section 4.2.1. Since the guided filter has a good performance in preserving edges and is efficient in computation with the linear filtering model, we use the guided filter for sparse cost aggregation. For each pixel, the cost aggregation is done in its adaptive support region as discussed in Section 4.2.2, but only for the disparity values in the disparity subset $M_C(p)$. Using the disparity subset, the aggregation for each pixel in the adaptive support region for each disparity value $d \in M_C(p)$ is obtained by summing up the weighted cost value of surrounding pixels whose disparity subset contains the current disparity value d. This can be represented as

$$C'(p, d) = \begin{cases} \sum_q W_{p,q}(I) C(q, d) o(q, d), & d \in M_C(p) \\ \Gamma, & \text{otherwise} \end{cases} \tag{4.16}$$

where $C(p, d)$ is the per-pixel cost volume calculated as in Section 4.2.1, I is the color image and $o(p, d)$ is the disparity subset selection function given by

$$o(p, d) = \begin{cases} 1, & d \in M_C(p) \\ 0, & \text{otherwise} \end{cases} \tag{4.17}$$

Cost values corresponding to disparity values outside the disparity subset are set to a large number Γ in the disparity calculation. $W_{p,q}(I)$ is the adaptive weight between p and neighborhood pixel q at disparity d, expressed as follows:

$$W_{p,q}(I) = \frac{1}{|\omega_p|} \sum_{k \in \omega_p} \Big(\frac{1}{|\omega_k|} \sum_{q \in \omega_k} \big(1+(I_p-\mu_k)^T(\Sigma_k+\varepsilon U)^{-1}(I_q-\mu_k)\big)\Big), \forall q : p \in \omega_q \quad (4.18)$$

where Σ_k and μ_k are the 3×3 covariance matrix and 3×1 mean vector of the color image I in the support region ω_k of pixel k, respectively. U is a 3×3 identity matrix and ε is a smoothness parameter. $|\omega_p|$ is the pixel number in the support region ω_p of point p. According to [40], unbalanced arm lengths in the horizontal direction or vertical direction will cause the gradient reversal artifacts in guided filtering, thus we set the horizontal arm lengths symmetrically to the smaller length of r_{left} and r_{right}, and the vertical arm lengths are set similarly as the smaller length of r_{up} and r_{bottom}.

In the guided filter implementation, the weight values $W_{p,q}(I)$ do not need to be calculated explicitly. Instead, the filtered cost volume output can be calculated using the linear model definition of guided image filter as in [77]. Firstly, the input per-pixel matching cost is computed as the cost volume only containing the cost values in the selected disparity subset, represented as

$$C_D(p,d) = C(p,d) \cdot o(p,d) \quad (4.19)$$

Secondly, the filtered cost volume output $C'(p,d)$ at disparity level d is calculated as a weighted sum of the linear transform of I in the support region ω_q centered at the pixel q and $p \in \omega_q$, represented as

$$C'(p,d) = \frac{\sum_{q \in \omega_p} |\omega_q(d)| \big(a_q(d)^T I_p + b_q(d)\big)}{\sum_{q \in \omega_p} |\omega_q(d)|}, \ \forall q : p \in \omega_q, \ d \in M_C(p) \quad (4.20)$$

where $\omega_q(d)$ is the support region of q at disparity d, and $|\omega_q(d)|$ is the number of pixels with non-zero cost in the support region ω_q at disparity d. "$\forall q : p \in \omega_q$" means all the pixels q whose support region ω_q include the pixel p.

The linear coefficients 3 × 1 $a_q(d)$ and 1 × 1 $b_q(d)$ can be solved through linear regression as follows::

$$a_q(d) = \frac{1}{|\omega_q(d)|}(\Sigma_q + \varepsilon U)^{-1}\Big(\sum_{i\in\omega_q} I_i C_D(i,d) - \mu_q \bar{C}_D(q,d)\Big),\ d \in M_C(p) \tag{4.21}$$

$$b_q(d) = \bar{C}_D(q,d) - a_q(d)^T \mu_q,\ d \in M_C(p) \tag{4.22}$$

where $\bar{C}_D(q,d)$ is the mean of the per-pixel cost measure in support region ω_q. Σ_q and μ_q are, respectively, the 3 × 3 covariance matrix and 3 × 1 mean vector of the color image I in the support region ω_q centered at pixel q.

Finally, for each pixel p, the number of non-zero cost pixels in its support region at different disparity levels of the disparity subset $M_C(p)$ could be different, thus the aggregated cost using a guided filter at different disparity levels may be unstable. To address this issue, we propose to further weight the guided filter output with an exponential weight function. The final sparsely aggregated cost volume $C_{SLAC}(p,d)$ is represented as

$$C_{SLAC}(p,d) = C'(p,d) \cdot B_p(d) \tag{4.23}$$

where the exponential weight function $B_p(d)$ is defined as

$$B_p(d) = \exp\left(-\frac{|\omega_p(d)|}{\max\limits_{d\in M_C(p)} |\omega_p(d)| \cdot \lambda_B}\right) \tag{4.24}$$

where λ_B is a constant to adjust the exponential kernel shape.

4.2.6 Cost Volume Optimization and Disparity Calculation

Since the aggregated cost volume $C_{SLAC}(p,d)$ is adaptively estimated based on the local neighborhood region, there are possible matching ambiguities in the cost volume that can cause mis-matching near depth discontinuities and occluded regions. In order to further remove the matching ambiguities in the cost volume, the aggregated cost volume is optimized using a localized support-region-based propagation by considering smoothness constraints and parallelism between stereo views. The proposed propagation is similar to the semi-global optimization (SGM) method in [43, 87], but the cost volume is propagated in local support regions instead of using all image pixels. The SGM usually aggregates the cost value at each disparity level in 1D from multiple path directions. In the proposed localized support-region-based propagation, we compute the recursive optimization in four directions $\boldsymbol{r}$: left-to-right ($\boldsymbol{r} = (1, 0)$), right-to-left ($\boldsymbol{r} = (-1, 0)$), top-to-bottom ($\boldsymbol{r} = (0, 1)$) and bottom-to-top ($\boldsymbol{r} = (0, -1)$). Given a direction vector $\boldsymbol{r}$, the aggregated cost $C_{\boldsymbol{r}}(p, d)$ along the direction $\boldsymbol{r}$ at pixel p and disparity $d \in M_C(p)$ can be represented as:

$$\begin{aligned} C_{\boldsymbol{r}}(p,d) = C_{SLAC}(p,d) + \min\Big(&C_{\boldsymbol{r}}(p-\boldsymbol{r},d), C_{\boldsymbol{r}}(p-\boldsymbol{r},d\pm 1) + P_1, \\ &\min_{dd\in M_C(p-\boldsymbol{r})} C_{\boldsymbol{r}}(p-\boldsymbol{r},dd) + P_2\Big) - \min_{dd\in M_C(p-\boldsymbol{r})} C_{\boldsymbol{r}}(p-\boldsymbol{r},dd), \text{ if } (p-\boldsymbol{r}) \in \omega(p) \end{aligned} \quad (4.25)$$

$$C_{\boldsymbol{r}}(p,d) = C_{SLAC}(p,d), \text{ if } (p-\boldsymbol{r}) \notin \omega(p) \quad (4.26)$$

where $p - \boldsymbol{r}$ is the previous pixel along the direction $\boldsymbol{r}$. The cost is propagated from the previous pixel only if the previous pixel lies in the current pixel's support region, thus the propagation is computed in a localized support region instead of on all image pixels. In (4.25), P_1 and P_2 are two smoothness penalty terms for the disparity changes between neighboring pixels, and $P_1 < P_2$. The values of P_1 and P_2 are set symmetrically according to the gray-scale intensity difference D_L between pixel p and $p - \boldsymbol{r}$ in the left view and the difference D_R between the corresponding pixel $p - (d, 0)$ and $p - (d, 0) - \boldsymbol{r}$ in the right view, represented as:

$$(P_1, P_2) = \begin{cases} (\pi_1, \pi_2), & D_L < \tau_{SO}, D_R < \tau_{SO} \\ (\pi_1/4, \pi_2/4), & D_L < \tau_{SO}, D_R > \tau_{SO} \\ (\pi_1/4, \pi_2/4), & D_L > \tau_{SO}, D_R < \tau_{SO} \\ (\pi_1/10, \pi_2/10), & D_L > \tau_{SO}, D_R > \tau_{SO} \end{cases} \quad (4.27)$$

where π_1 and π_2 are constant penalty values, and τ_{SO} is the threshold on intensity difference between adjacent pixels. Note that, for each pixel p, we only aggregate the cost values corresponding to the disparity subset values in $M_C(p)$, and this helps to save the computational time of propagation.

The optimized cost volume $C_{final}(p, d)$ is obtained by averaging the propagated path costs in four directions as:

$$C_{final}(p, d) = \frac{1}{4} \sum_{\boldsymbol{r}} C_{\boldsymbol{r}}(p, d) \tag{4.28}$$

After obtaining the filtered cost volume using the proposed sparse locally adaptive cost aggregation as described above, the disparity map of each view is initially computed by selecting the disparity value corresponding to the minimal cost value in the aggregated cost volume $C_{final}(p, d)$ for $d \in M_C(p)$ using a winner-take-all (WTA) approach as follows:

$$\hat{D}(p) = arg \min_{d \in M_C(p)} C_{final}(p, d) \tag{4.29}$$

4.2.7 Disparity Refinement

Despite the effectiveness and efficiency of the local cost volume filtering and locally-support-region-based propagation, the initial estimated disparity maps contain outliers such as mis-matches and occlusion, due to the various scene structures and characteristics. The local stereo methods usually make the assumption of fronto-parallelism for object surfaces in the stereo matching process, thus they do not perform well for slanted surface regions. Another drawback of the local stereo methods is that they are not robust to large homogeneous and textureless regions since image pixels are matched in a limited neighborhood support region. In the slanted surfaces and homogeneous regions, the disparity map might have sharp discontinuities and blocking artifacts that affect the overall disparity accuracy.

Many post-processing methods were proposed to refine the disparity maps in recent years. Some disparity post-processing methods use image segmentation to detect and correct the inconsistent regions in disparity maps [72, 89, 108], but segmentation methods are usually computationally expensive. The local stereo methods in [70, 87, 105] use different interpolation methods based on the local support region to correct the mis-matched pixel regions, but the local interpolation methods can not handle the large homogeneous regions and slanted surfaces. The method in [42] proposes to detect the mis-matched homogeneous regions using the cost-ratio measure of the aggregated cost volume, and correct the large homogeneous regions using the weighted propagation method only in the horizontal directions. But it lacks the robustness to deal with homogeneous regions with disparity changes and slanted surfaces, since the weighted propagation is computed at the same disparity and only in horizontal directions.

In this section, we propose a novel disparity refinement method to handle mismatches including slanted and homogeneous regions. The refinement method firstly detects the unreliable regions in the initial disparity maps, and corrects the disparity values in these regions using a multi-direction weighted cost propagation method with disparity change and smoothness compensation terms. The remaining invalid disparity pixels in each view are further interpolated adaptively using the local support region.

A. Detection of Unreliable Disparity Regions

The unreliable disparity pixels in each view consist of two parts: the invalid disparity pixels corresponding to occlusions and unstable disparity pixels in large homogeneous regions.

Using the disparity map of the left view $\hat{D}^{left}(p)$ and the right view $\hat{D}^{right}(p)$,

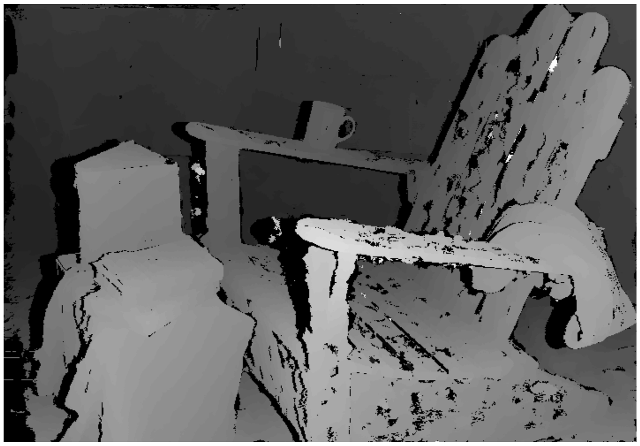


Fig. 4.6: Invalid Pixel Map of the Adirondack Dataset in the Middlebury Version 3 Database. Invalid Pixels are Marked as Black Color.

the left-right cross consistency check is implemented to detect invalid disparity pixels in each view, respectively. For a pixel p in the left view, if the disparity of its corresponding point in the right view does not agree with the disparity of p in the left view, the pixel is marked as invalid for further approximation. The invalid pixels are those which do not satisfy the following criteria:

$$\left|\hat{D}^{left}(p) - \hat{D}^{right}(p - (\hat{D}^{left}(p), 0))\right| \leq T_D \tag{4.30}$$

where $p - (\hat{D}^{left}(p), 0) = (x - \hat{D}^{left}(p), y)$, and $T_D = 1$ for integer disparity ranges. The set of invalid pixels is denoted as $P_{invalid}$. The set of invalid pixels in the right view is computed in a similar manner. These invalid pixels usually contain matching outliers and occlusion pixels between the two views. The invalid pixels are marked as black color in Fig. 4.6 for the left view of the Adirondeck image set of the Middlebury version 3 database.

In the initially computed disparity map $\hat{D}(p)$ for each view, we further identify pixels with unstable aggregated cost using both the local color standard deviation calculated using (4.14) in Section 4.2.2 and the cost ratio between the minimum cost

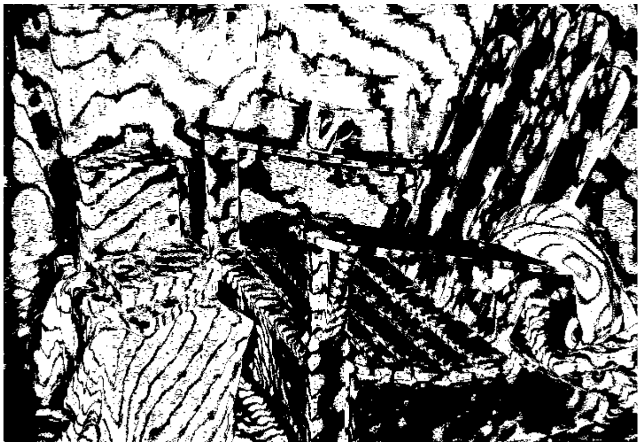


Fig. 4.7: Unreliable Pixel Map of the Adirondack Dataset in the Middlebury Version 3 Database. Unreliable Pixels are Marked as Black Color.

value and the second minimum cost value in its sparse cost volume $C_{final}(p, d)$. The cost ratio is expressed as

$$R_{cost}(p) = \frac{\left|C_{final\_1}(p) - C_{final\_2}(p)\right|}{C_{final\_2}(p)} \tag{4.31}$$

where $C_{final\_1}(p)$ is the minimum cost value for $d \in M_C(p)$, and $C_{final\_2}(p)$ is the second minimum cost value. Pixels with a local color standard deviation smaller than a threshold T_{cv} and with a cost ratio value smaller than a threshold T_{cost} are marked as unstable pixels. Both the invalid pixels and the unstable pixels are denoted as unreliable pixels. An example of the detected unreliable pixels are marked as black color in Fig. 4.7 for the left view of the Adirondeck data set in the Middlebury version 3 benchmark database.

B. Multi-Direction Weighted Propagation with Disparity Compensation

In order to normalize the cost volume $C_{SLAC}(p, d)$, an absolute difference cost volume $C_{diff}(p, d)$ is generated using the aggregated SLAC cost volume $C_{SLAC}(p, d)$ computed in Section 4.2.5 as follows:

$$C_{diff}(p,d) = \begin{cases} 0, \quad p \in P_{invalid} \\ \big|C_{SLAC}(p,d) - C_{best}(p)\big|, \quad d \in M_C(p) \end{cases} \tag{4.32}$$

where $C_{best}(p)$ correspond to the minimum matching cost in the disparity subset of pixel p.

Using the absolute difference cost volume, the cost at each disparity level in the considered disparity subset is aggregated using the weight propagation method in four directions $\boldsymbol{r}$: left-to-right $\big(\boldsymbol{r} = (1,0)\big)$, right-to-left $\big(\boldsymbol{r} = (-1,0)\big)$, top-to-bottom $\big(\boldsymbol{r} = (0,1)\big)$ and bottom-to-top $\big(\boldsymbol{r} = (0,-1)\big)$. Given a direction vector $\boldsymbol{r}$, the aggregated cost $C'_{\boldsymbol{r}}(p,d)$ along the direction $\boldsymbol{r}$ at pixel p and disparity $d \in M_C(p)$ can be represented as:

$$\begin{aligned} C'_{\boldsymbol{r}}(p,d) = C_{diff}(p,d) + \mu \cdot \min\Big(&C'_{\boldsymbol{r}}(p-\boldsymbol{r},d), C'_{\boldsymbol{r}}(p-\boldsymbol{r},d\pm 1) + H_1, \\ &\min_{dd \in M_C(p-\boldsymbol{r})} C'_{\boldsymbol{r}}(p-\boldsymbol{r},dd) + H_2\Big), \text{ if } (p-\boldsymbol{r}) \in \omega(p) \end{aligned} \tag{4.33}$$

$$C'_{\boldsymbol{r}}(p,d) = C_{diff}(p,d), \text{ if } (p-\boldsymbol{r}) \notin \omega(p) \tag{4.34}$$

In (4.33), H_1 and H_2 are smoothness penalty terms, μ is the propagation coefficient calculated using a kernel function of the intensity difference between pixel p and the previous pixel $p - \boldsymbol{r}$ along the direction $\boldsymbol{r}$, as follows [76]:

$$\mu = \exp\Big(-\frac{|I(p) - I(p-\boldsymbol{r})|}{\sigma}\Big) \tag{4.35}$$

where $I(p)$ is the gray-scale image intensity, and σ is a smoothness term [76].

The propagated cost volume is the average of the aggregated costs in all directions and is given by:

$$C_{prop}(p,d) = \frac{1}{4}\sum_{\boldsymbol{r}} C'_{\boldsymbol{r}}(p,d) \tag{4.36}$$

Using the propagated cost volume of each view, the disparity map D_{prop} is calculated using a winner-take-all operation as follows:

$$D_{prop} = arg \min_{d \in M_C(p)} C_{prop}(p, d) \tag{4.37}$$

For the detected unreliable pixels (Section 4.2.7 A), the disparity value is replaced by the disparity value in D_{prop}.

The proposed weighted propagation adapts the accumulated cost value to the disparity change in homogeneous regions by using the smoothness penalty terms, and it greatly helps to reduce the disparity error in the unreliable regions including invalid pixels and unstable pixels. The resulting disparity map using the proposed weighted propagation is shown in Fig. 4.8. Fig. 4.8 shows the effectiveness of the weighted propagation in correcting the disparity in homogeneous and slanted regions. Note that for both the proposed localized support-region-based propagation and weighted propagation, the idea is similar to a local stereo aggregation since the propagation is only computed based on the sparse disparity subset and in the local support region, while the regular SGM performs the propagation over all image points.

C. Adaptive Interpolation

Using the refined disparity maps for both views, another left-right-consistency check is computed and the detected invalid pixels are filled with reliable neighboring pixel disparities. We use an iterative cross-region-based interpolation method to approximate the disparity of invalid pixels. Starting from the invalid pixel set $P_{invalid}$ near the reliable pixels in $\hat{D}^{left}(p)$ by searching invalid pixels p whose 8-connect neighborhood includes at least one reliable pixels, the ratio $R_{pixel}(p)$ of the reliable pixel number over the total pixel number in the cross-based support region $\omega(p)$ is firstly calculated for each invalid pixel p in $P_{invalid}$. If $R_{pixel}(p)$ is larger than a threshold

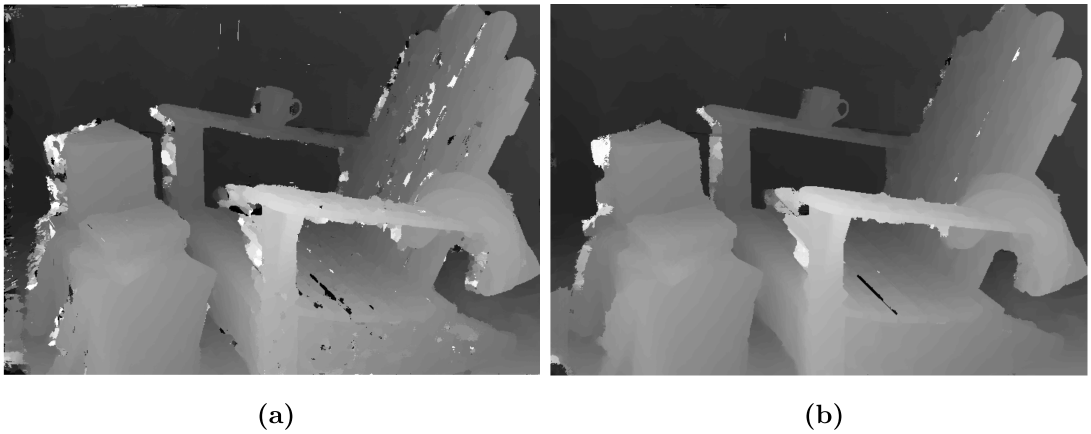


Fig. 4.8: Disparity Maps of the Adirondack Image Set in the Middlebury Version 3 Database. (a) Disparity Map Before Disparity Refinement. (b) Disparity Map After Disparity Refinement using Weighted Propagation.

T_{out}, the histogram $H'(p, d)$ is calculated for the disparity values of reliable pixels in the cross-based support region of p. The disparity of the invalid pixel p is assigned as the disparity bin corresponding to the histogram peak in $H'(p, d)$. The filled invalid pixel is marked as "reliable" pixel for the next iteration, and this filling process is repeated until no extra invalid pixels are filled.

The remaining invalid pixels are filled using the disparity of the nearest reliable pixel in the horizontal scanline. For each remaining invalid pixel p, the disparity d_L of the nearest reliable pixel to the left of p and the disparity d_R of the nearest reliable pixel to the right of p are considered, and the minimum disparity of d_L and d_R is assigned as the refined disparity for the invalid pixel p. The final disparity map after refinement is the final stereo matching result. Fig. 4.9 shows the refined disparity map of the"Adirondack" dataset in the Middlebury version 3 database.

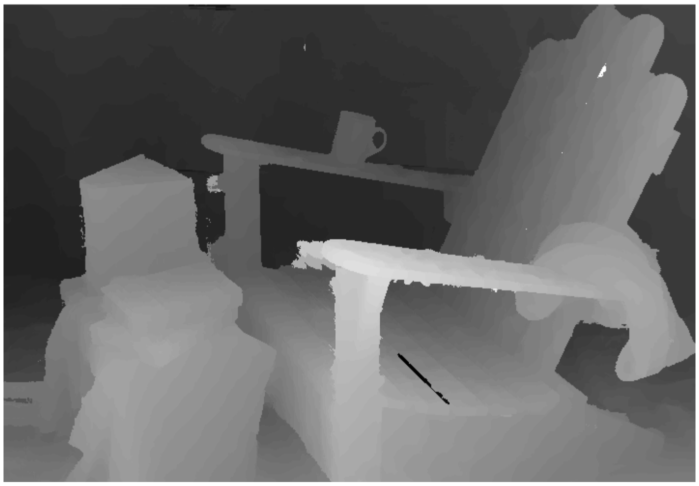


Fig. 4.9: Final Disparity Map of the Left View of the Adirondack Image Set in the Middlebury Version 3 Training Database.

Table 4.1: Parameters Used in the Proposed Stereo Matching System for All Image Sets.

λ_{CEN}	λ_{BT}	λ_{GD}	α	β	π_1	π_2	τ_{SO}	c_1	c_2	N_p
40	20	2	0.5	0.1	0.06	0.12	15	2	20	9
M	T_{cv}	T_{cost}	L_{max}	λ_B	T_D	T_{out}	H_1	H_2	σ	ϵ
2	0.001	0.1	5	4	1	0.4	0.001	0.012	0.05	1e-4

4.3 Experimental Results

The parameters used in our algorithm are summarized in Table 4.1. The parameters are tested extensively with different values and combinations using the Middlebury version 2 and version 3 training image sets. The parameters corresponding to the lowest disparity error are chosen as the system parameters. For different testing image scenes and dataset, the parameters are fixed values.

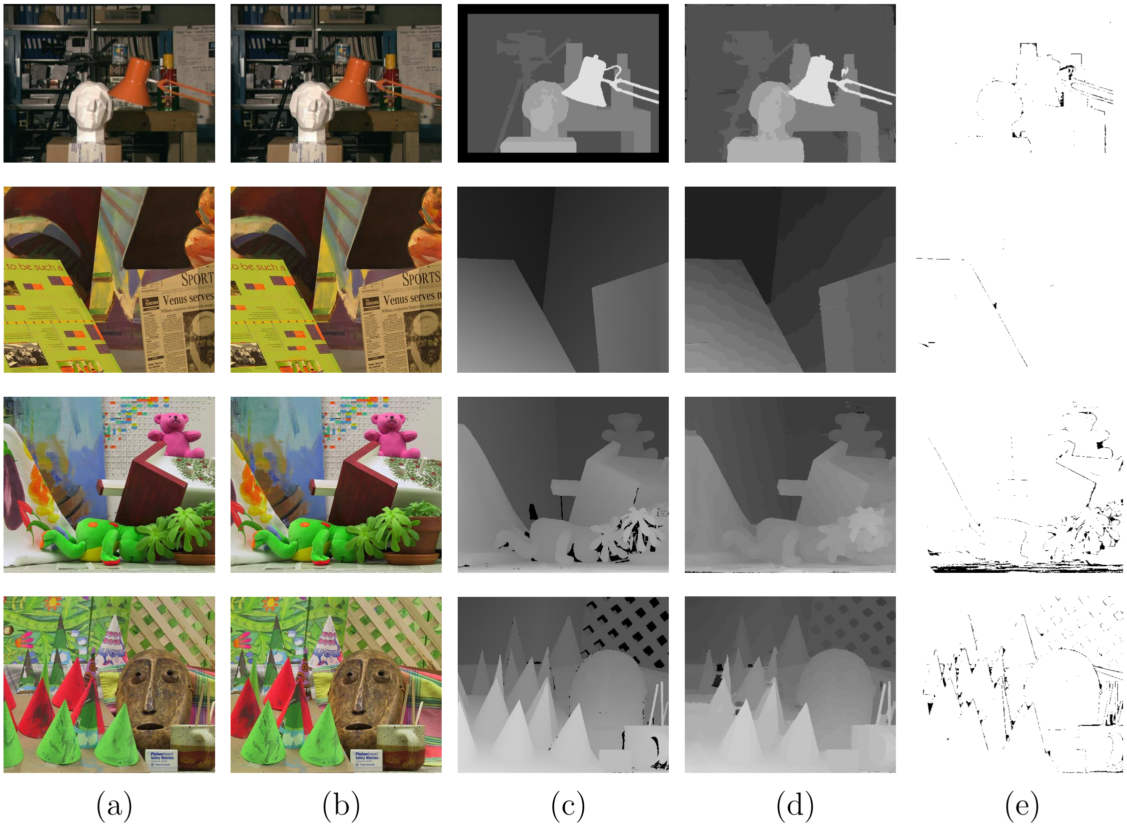


Fig. 4.10: Results for the Middlebury Benchmark Version 2 Stereo Image Pairs. From Top Row to Bottom Row: Tsukuba, Venus, Teddy and Cones. (a) Left View of Original Images; (b) Right View of Original Images; (c) Ground-truth Disparity Maps; (d) Disparity Maps of the Proposed Method; (e) Error Map with Bad Pixels in "nonocc" Area Shown in Black Color.

Table 4.2: Bad Pixel Error Results Based on the Middlebury Stereo Evaluation Version 2 Benchmark Dataset. APBP stands for Average Percent of Bad Pixels.

Algorithm	Tsukuba			Venus			Teddy			Cones			APBP
	Nonocc	All	Disc	Nonocc	All	Disc	Nonocc	All	Disc	Nonocc	All	Disc	(%)
Proposed	**1.15**	**1.62**	**5.94**	**0.17**	**0.39**	**1.85**	**4.62**	**10.62**	**12.90**	**2.47**	**8.38**	**7.27**	**4.78**
AdaptiveGF [42]	1.04	1.53	5.62	0.17	0.41	1.98	5.71	11.3	14.3	2.44	8.22	7.05	4.98
HistAggr2 [41]	1.93	2.30	6.39	0.16	0.46	2.22	5.88	11.3	14.7	2.41	8.18	7.21	5.20
CrossLMF [40]	2.46	2.78	6.26	0.27	0.38	2.15	5.50	10.6	14.2	2.34	7.82	6.80	5.13
CostFilter [70]	1.51	1.85	7.61	0.20	0.39	2.42	6.16	11.8	16.0	2.71	8.24	7.66	5.55
NonLocalFilter [111]	1.47	1.85	7.88	0.25	0.42	2.60	6.01	11.6	14.3	2.87	8.45	8.10	5.48
RecursiveBF [112]	1.85	2.51	7.45	0.35	0.88	3.01	6.28	12.1	14.3	2.80	8.91	7.79	5.68
AdaptWeight [65]	1.38	1.85	6.90	0.71	1.19	6.13	7.88	13.3	18.6	3.97	9.79	8.26	6.67

4.3.1 Accuracy Evaluation

A: Disparity Results For The Middlebury Stereo Database Version 2

The proposed local stereo matching system is tested on the Middlebury stereo benchmark version 2 data set [38] using four benchmark stereo pairs: Tsukuba, Venus, Teddy and Cones. The input left image, right image and ground-truth disparity map of these datasets are shown in Fig. 4.10 (column (a) to (c)). The ratio N_s of the disparity subset size with respect to the total disparity range is set to 0.4. The notion of “nonocc” corresponds to the non-occluded regions, “all” corresponds to all regions including half-occluded regions, and “disc” corresponds to regions near depth discontinuities. The masks of “nonocc”, “all” and “disc” for 4 image sets are shown in Fig. 4.11. In the Middlebury benchmark evaluation, the bad pixels are defined as the pixels whose disparity difference between the calculated disparity and the ground-truth disparity value is larger than a threshold of 1.0. The bad pixel error is the ratio of bad pixel number over the total image pixel number. The bad pixel error is evaluated in the white color region for each mask, respectively.

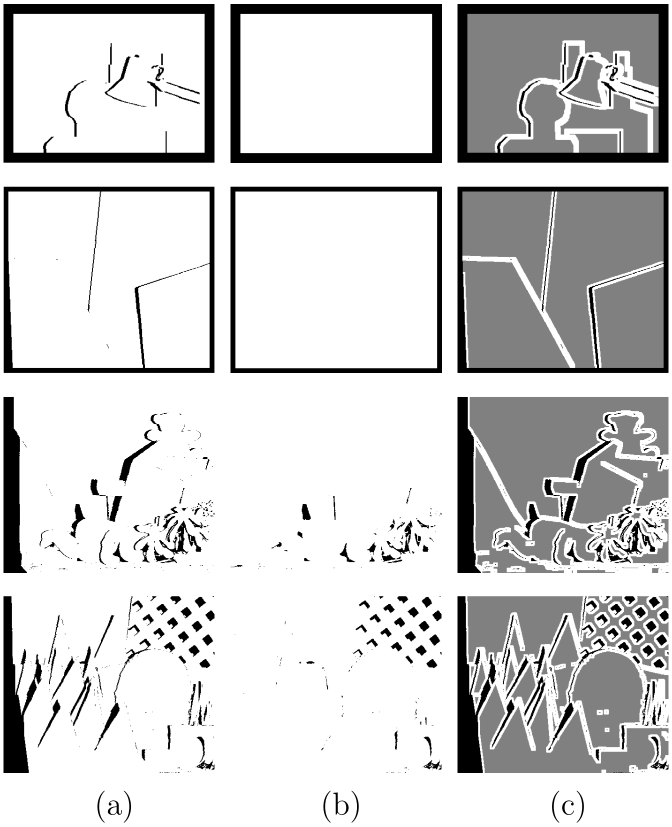


Fig. 4.11: The Masks for Bad Pixel Error Evaluation for the Middlebury Benchmark Version 2 Stereo Image Pairs. From Top Row to Bottom Row: Tsukuba, Venus, Teddy and Cones. (a) “nonocc”; (b) “all”; (c) “disc”. Bad Pixels Error is Evaluated in White Regions of the Masks.

The accuracy of our propose SLAC stereo matching system is evaluated based on the Middlebury benchmark version 2. For each dataset, the bad pixel error is calculated for all pixels (all), non-occluded regions (nonocc) and regions near depth discontinuities (disc). The calculated disparity maps and bad pixel images are shown in Fig. 4.10 (columns (d) and (e)). Instead of using the whole disparity range for cost aggregation and disparity calculation, the proposed stereo matching method using sparse locally adaptive cost aggregation is capable to generate accurate disparity maps with object boundaries and depth discontinuities preserved. The disparity results are shown in Table 4.2 for the proposed method along with several state-of-art local stereo matching algorithms, such as AdaptiveGF [42], HistAggr2 [41], CrossLMF [40] and CostFilter [70]. Our proposed method is ranked 17 out of 168 methods in the Middlebury version 2 stereo benchmark. The proposed SLAC stereo matching method has better performance than existing local stereo matching methods.

B. Disparity Results For The Middlebury Stereo Database Version 3

The accuracy of our proposed SLAC stereo matching system is evaluated based on the Middlebury benchmark dataset version 3 [38]. The Middlebury version 3 dataset contains 15 training image sets and 15 testing image sets. For each dataset, the stereo images have three resolutions: full, half and quarter resolution. The bad pixel error is calculated for non-occluded regions (nonocc). We test the proposed SLAC stereo system on all 15 pairs of stereo images in the training image set with quarter resolution (Midd3_Q) and half resolution (Midd3_H). The ratio N_s of the disparity subset size with respect to the total disparity range is set to 0.4. The disparity error threshold is set to 1.0 in quarter resolution image sets, and is set to 2.0 in half resolution image sets equivalently. The calculated disparity map and bad pixel images are shown in Fig. 4.12 and Fig. 4.13 (columns (d) and (e)).

The numerical results for the training dataset are shown in Table 4.3, and the results for the testing dataset are shown in Table 4.4. At the time of submission in December 2016, the ranks of the proposed method are as indicated below, and the error threshold is 1 pixel at quarter resolution (i.e., “bad 4.0”). The weighted average of bad pixel error on the half-resolution training dataset is 10.6, with a rank of 20. The weighted average of bad pixel error on the quarter-resolution training dataset is 11.71, with a rank of 23. The weighted average of bad pixel error on the half-resolution testing dataset is 11.7, with a rank of 22. The online Middlebury

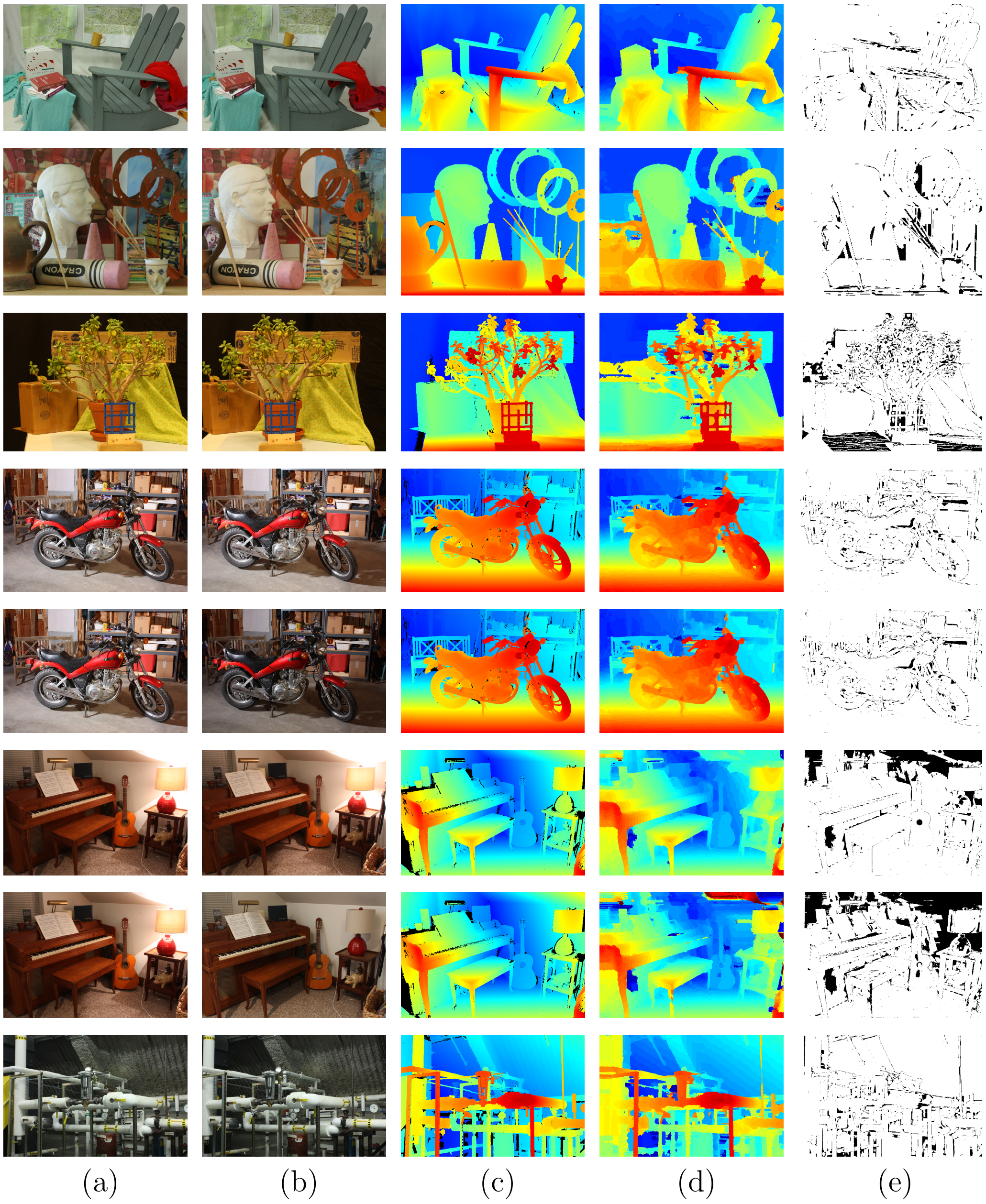


Fig. 4.12: Results of Middlebury Benchmark Version 3 Quarter Resolution Training Stereo Image Pairs, From Top Row to Bottom Row: Adirondack, ArtL, Jadeplant, Motorcycle, MotorcycleE, Piano, PianoL and Pipes. (a) Left View of Original Images; (b) Right View of Original Images; (c) Ground-truth Disparity Maps; (d) Disparity Maps of the Proposed Method. Disparity Values Ascend from Blue to Red Color; (e) Error Map with Bad Pixels in “nonocc” Area Shown in Black Color.

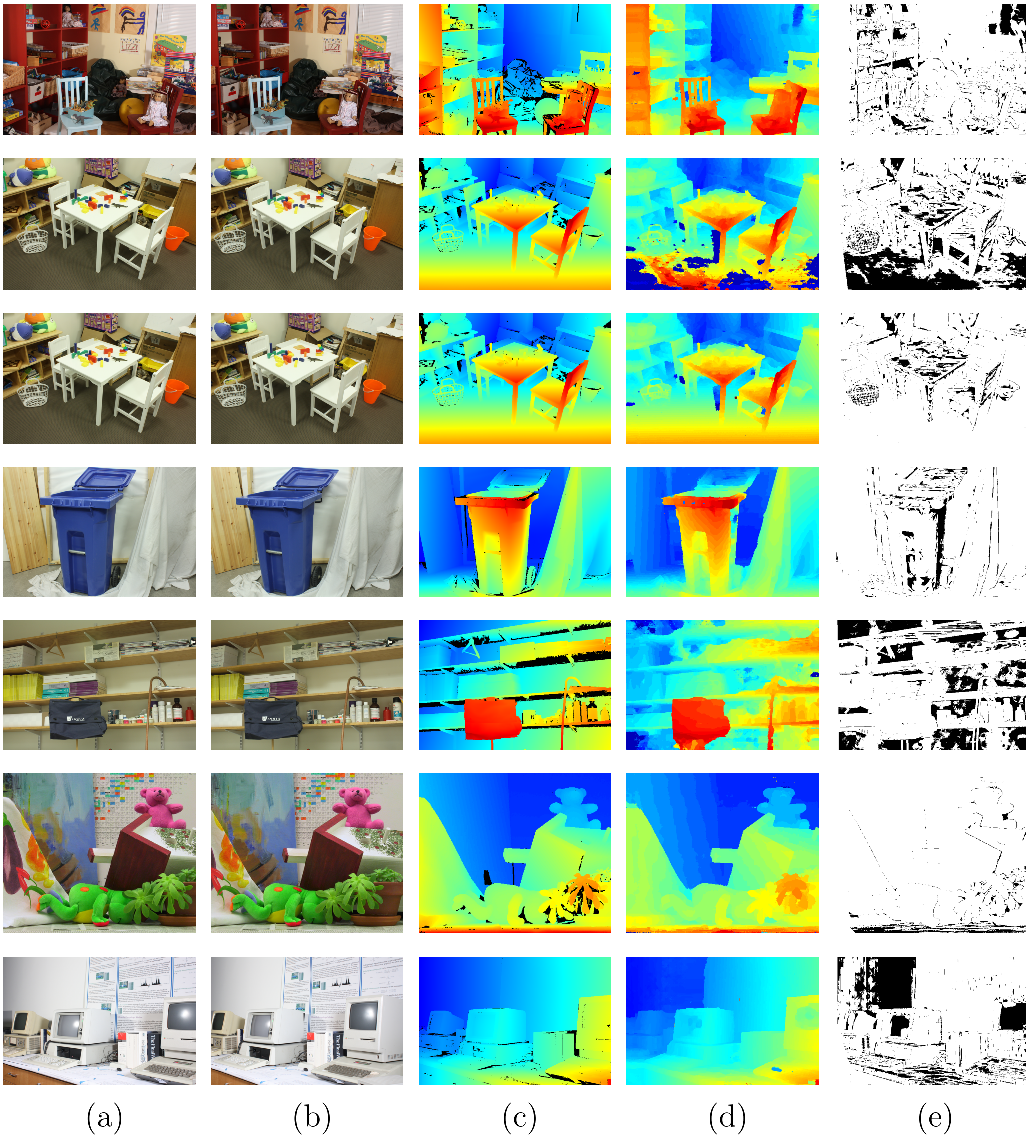


Fig. 4.13: Results of Middlebury Benchmark Version 3 Quarter Resolution Training Stereo Image Pairs, From Top Row to Bottom Row: Playroom, Playtable, PlaytableP, Recycle, Shelves, Teddy and Vintage. (a) Left View of Original Images; (b) Right View of Original Images; (c) Ground-truth Disparity Maps; (d) Disparity Maps of the Proposed Method. Disparity Values Ascend from Blue to Red Color; (e) Error Map with Bad Pixels in “nonocc” Area Shown in Black Color.

Table 4.3: Bad Pixel Error of the Proposed Stereo Matching Algorithm Based on the Middlebury Stereo Evaluation Benchmark Version 3 Quarter Resolution and Half Resolution Training Image Set. The Disparity Error Threshold for Quarter Resolution Image Sets is 1 Pixel, and the Disparity Error Threshold for Half Resolution Image Set is 2 Pixels Equivalently.

Algorithm	Adiron	ArtL	Jadepl	Motor	MotorE	Piano	PianoL	Pipes	Playrm	Playt	PlaytP	Recyc	Shelves	Teddy	Vintge	Weighted Avg
Proposed SLAC _H	**4.21**	**6.26**	**13.6**	**4.96**	**4.54**	**12.9**	**22.6**	**7.11**	**10.1**	**37.9**	**6.94**	**6.27**	**32.6**	**3.54**	**22.0**	**10.6**
Proposed SLAC _Q	**4.50**	**8.89**	**16.57**	**5.69**	**5.40**	**15.43**	**24.76**	**7.90**	**12.78**	**29.64**	**8.19**	**7.50**	**32.42**	**4.62**	**23.71**	**11.71**
SGM_H [43]	7.95	5.92	12.3	5.91	4.87	11.7	22.1	7.87	12.1	42.4	10.3	8.42	35.4	4.70	31.4	12.1
SGM_Q [43]	5.43	11.1	18.1	6.32	5.71	14.5	26.6	10.1	14.6	25.8	11.4	8.34	35.0	5.47	30.3	13.0
LAMC_DSM [105]	10.8	11.1	20.3	7.40	6.63	15.6	23.8	12.4	23.5	23.2	14.9	11.4	42.2	6.62	29.1	15.0
SNCC [113]	9.37	10.5	19.0	7.00	6.16	16.8	24.5	12.1	22.0	45.5	11.2	10.6	38.1	6.93	32.4	15.3
Cens5 [82]	15.0	11.3	18.3	8.70	7.46	18.9	28.7	11.7	23.3	46.5	13.1	12.3	41.2	7.82	42.4	17.3

Table 4.4: Bad Pixel Error of the Proposed Stereo Matching Algorithm Based on the Middlebury Stereo Evaluation Benchmark Version 3 Half Resolution Testing Image Set. The Disparity Error Threshold for Half Resolution Image Set is 2 Pixels.

Algorithm	Austr	AustrP	Bicyc2	Class	ClassE	Cpmpu	Crusa	CrusaP	Djemb	DjembL	Hoops	Livgrm	Nkuba	Plants	Stairs	Weighted Avg
Proposed SLAC _H	**20.3**	**3.61**	**6.49**	**17.4**	**31.8**	**6.21**	**11.6**	**5.94**	**2.89**	**32.8**	**14.5**	**13.5**	**14.8**	**8.59**	**11.2**	**11.7**
SGM_Q [43]	20.1	6.25	7.99	9.98	21.9	12.0	9.53	5.64	3.73	27.5	18.2	14.6	13.7	10.9	18.2	11.8
SGM_H [43]	26.7	3.56	5.02	20.0	34.4	6.61	9.90	3.27	2.70	19.8	17.2	17.8	15.0	8.47	21.2	12.2
SNCC [113]	34.7	6.19	7.65	22.4	41.1	8.34	20.2	7.12	4.74	17.4	25.1	23.0	16.2	11.6	21.1	15.8
Cens5 [82]	34.6	6.46	8.60	22.2	38.7	13.9	21.7	10.9	5.66	27.5	30.8	25.5	19.9	14.0	35.7	18.6
LAMC_DSM [105]	39.1	10.2	12.9	14.7	35.5	13.5	23.4	15.3	4.48	26.6	25.9	23.5	21.4	21.9	30.9	19.2

version 3 evaluation table consists of a total of 54 submitted results using various stereo matching methods on December 2016, but most of the methods are based on global stereo matching and optimization algorithms. Only 6 local stereo methods are evaluated based on the Middlebury version 3 dataset. Among the evaluation results of the training and testing dataset, there are no local stereo methods in the top 30 methods. The proposed local stereo matching system achieves top performance in the category of local stereo matching, and has an overall ranking of 20 on half-resulution training images and 22 on half-resolution testing images.

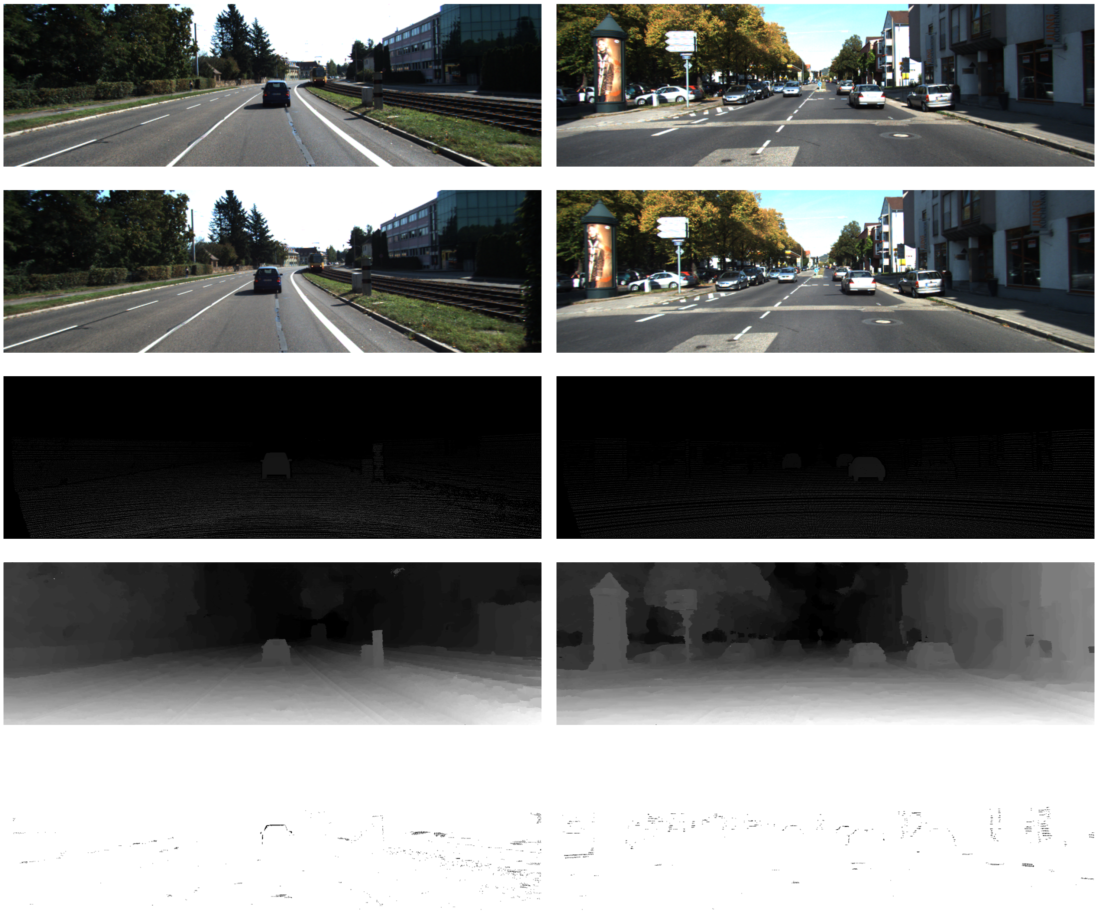


Fig. 4.14: Bad Pixel Error of the Proposed Stereo Matching Algorithm Based on the KITTI 2015 Stereo Evaluation Benchmark Training Image Set. The Disparity Error Threshold is 3.0 Pixel. The Results for Image Number 110 and 138 are Shown in Column 1 and Column 2, Respectively. From Top Row to Bottom Row: Left Image, Right Image, Ground-Truth Disparity Map, Disparity Map of the Proposed Method, Bad Pixel in Black Color.

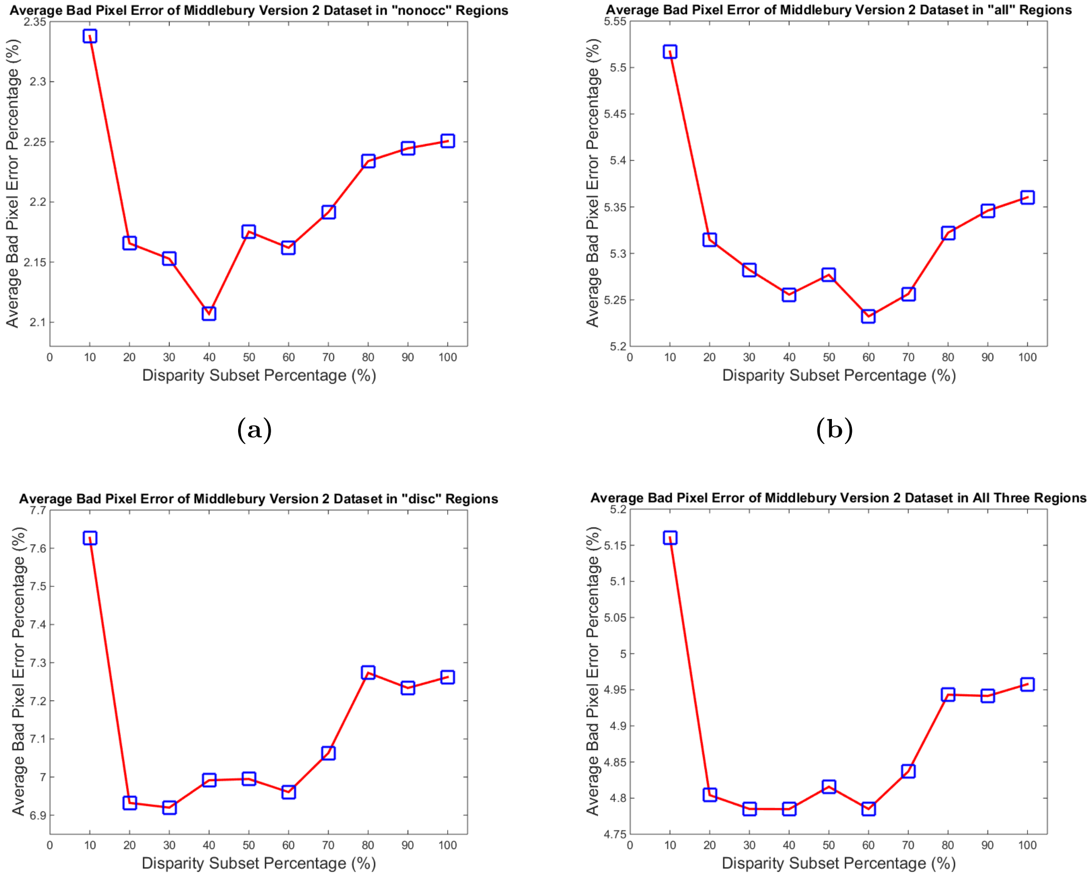
C. Disparity Results For The KITTI 2015 Stereo Dataset

The accuracy of our proposed SLAC stereo matching system is evaluated based on the KITTI 2015 Stereo Dataset [39]. The image scenes in the KITTI database consist of outdoor scenes with large homogeneous regions and background regions with far objects. We test our system on all 200 training image pairs. The disparity error threshold is 3.0 pixels according to [39]. The average bad pixel error for 200 training image pairs is 6.49%. Some examples of the calculated disparity maps and bad pixel images are shown in Fig. 4.14 (row 4 and row 5).

4.3.2 Evaluation of Disparity Subset Performance

We further analyze the performance of the proposed SLAC stereo matching system in terms of the l_0 norm of the disparity subset on the Middlebury stereo benchmark data set version 2 and version 3 training image pairs.

With a varyiing number of disparity subset candidates, the average bad pixel error of “nonocc”, “all” and “disc” area for four Middlebury version 2 datasets is shown in Figs. 4.15 (a) - (c). The average bad pixel error of all three regions are plotted for different disparity subset size ratio in Fig. 4.15 (d). The weighted average bad pixel error of “nonocc” area for the Middlebury version 3 quarter-resolution training dataset is shown in Fig. 4.16. The disparity subset size ratio N_s is varied from 10% to 100%. As shown in Fig. 4.15 and Fig. 4.16 for both Middlebury datasets, the error rate does not decrease and converge as the number of selected disparity subset candidates increases. The accuracy using the full disparity range does not produce the best disparity results. Instead, the error rate has a slightly increasing trend as the number of disparity subset candidates increases to the total disparity number for most datasets. This observation coincides with the conclusions in [41] that using all disparity values in cost aggregation does not guarantee a more accurate disparity map. By selecting a proper disparity subset, the estimated disparity map can have a higher accuracy compared to using the full disparity range.



(a)

(b)

(c)

(d)

Fig. 4.15: Performance Evaluation With Respect To Disparity Subset Size using the Middlebury Version 2 Benchmark Dataset. Average Bad Pixel Error of the Four Image Sets in Different Regions is Plotted for Different Size Ratio of Disparity Subset. (a) Average Bad Pixel Error in “nonocc” Region, (b) Average Bad Pixel Error in “all” Region, (c) Average Bad Pixel Error in “disc” Region, (d) Total Average Bad Pixel Error in All Three Regions.

4.3.3 Evaluation of Intermediate Cost Aggregation Steps

In order to illustrate the effectiveness of the proposed SLAC system, the performance of each cost aggregation and refinement step is evaluated based on the Middlebury version 3 quarter-resolution training dataset. The intermediate disparity map is computed after each step including fast coarse cost aggregation, sparse

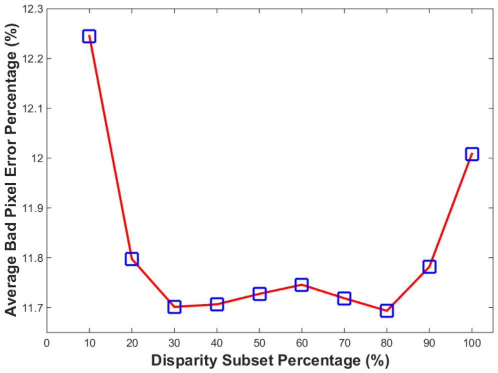


Fig. 4.16: Performance Evaluation With Respect To Disparity Subset Percentage. Weighted Average Bad Pixel Error of "nonocc" Area is Plotted for 15 Training Image Pairs in the Quarter-Resolution Middlebury Version 3 Benchmark Dataset.

guided-filter-based cost aggregation, localized support-region-based propagation, and disparity refinement. The weighted average bad pixel error rate is calculated for all 15 training image pairs. The plot of weighted average error rate after each step is shown in Fig. 4.17. The matching error rate after the fast coarse cost aggregation is the highest because we only aggregate the per-pixel cost uniformly in the support region, and there are more noise and mis-matches near the depth discontinuities. The matching error rate greatly drops after the sparse cost aggregation and the localized support-region-based propagation. In the disparity refinement steps, the occlusion detection, weighted cost propagation and filling further reduce the mis-matched and occluded areas to generate the final disparity result.

4.3.4 Complexity and Efficiency of The Proposed SLAC System

We briefly analyze the complexity of the proposed SLAC system in terms of Operations Order. Suppose the image size is $W \cdot H$, the full disparity range is D_{max} and the size ratio of the disparity subset is N_s. In the cost computation step, the total computation complexity is $O(N_p W H D_{max})$, where N_p represents the census

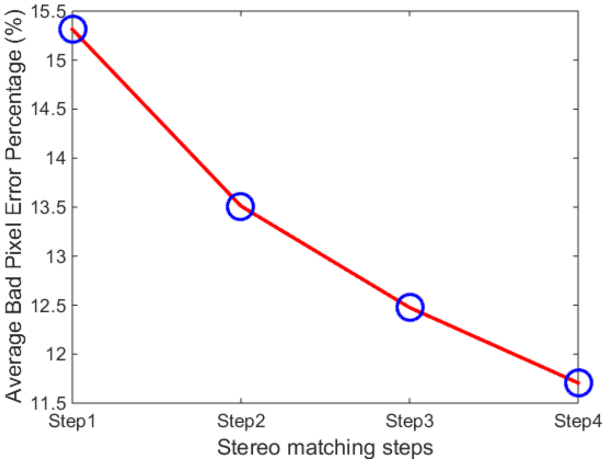


Fig. 4.17: Performance Evaluation After Intermediate Stereo Matching Steps. Step1: Fast Coarse Aggregation; Step2: Sparse Guided Filter; Step3: Localized Support-Region-Based Propagation; Step4: Disparity Refinement. Weighted Average Bad Pixel Error of "nonocc" Area is Plotted for 15 Training Dataset in the Middlebury Quarter Resolution Version 3 Stereo Database.

window size. Computing a cross-based arm requires $O(WHL)$ operations, where L is the arm length. The complexity of fast coarse cost aggregation and disparity subset computation is $O(WHD_{max})$ for each. The sparse guided-filter-based cost aggregation, multi-direction weighted propagation and localized support-region-based propagation have a complexity of $O(WHD_{max}N_s)$ since the computation is only done on the disparity subset. The complexity of the winner-take-all disparity computation is $O(WHD_{max}N_s)$ and of disparity refinement using adaptive interpolation is $O(WHL^2 + WHD_{max}N_s)$, where L is the arm length of the support region. From the above complexity analysis, we can see that the proposed SLAC system retains the linear complexity of local stereo matching methods.

Additionally, using the sparse disparity subset further helps in saving computations in the steps of sparse cost aggregation, sparse localized support-region-based

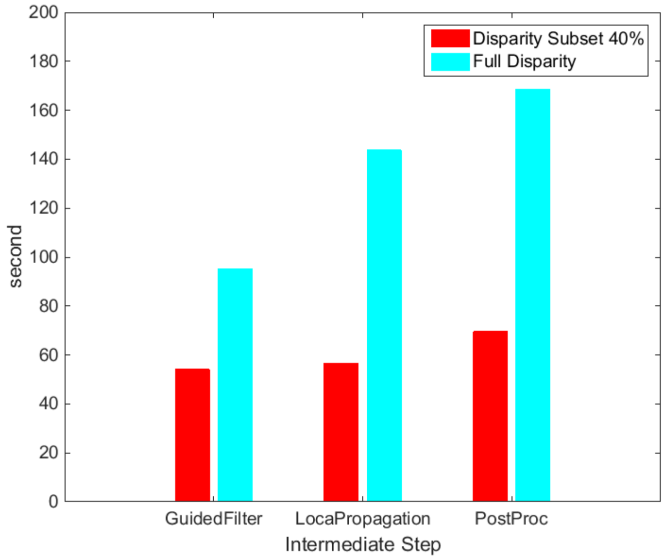


Fig. 4.18: Comparison of Processing Time Between Using Disparity Subset with $N_s = 40\%$ and Using Full Disparity Range, for Guided-Filter Cost Aggregation, Localized Support-Region-Based Propagation and Weighted Propagation in Disparity Refinement.

propagation and weighted propagation used for disparity refinement. In Fig. 4.18, a comparison of processing time between using the disparity subset with $N_s = 40\%$ and using a full disparity range for guided-filter cost aggregation, and weighted propagation used for disparity refinement is plotted as a bar chart. The total processing time of the above three steps using a full disparity range is about 2.26 times of the time using a disparity subset.

In order to analyze the efficiency of the proposed stereo system, the processing time of each step is profiled based on the C++ OpenCV code on a computer with i7 core @2.67GHz. The average processing time using the Middlebury version 3 quarter resolution training images is shown in Fig. 4.19, for 7 intermediate steps: cost computation, cross-based support region computation, fast coarse cost aggregation, disparity subset computation, sparse guided-filter-based cost aggregation, sparse localized support-region-based propagation, and disparity post processing using weighted prop-

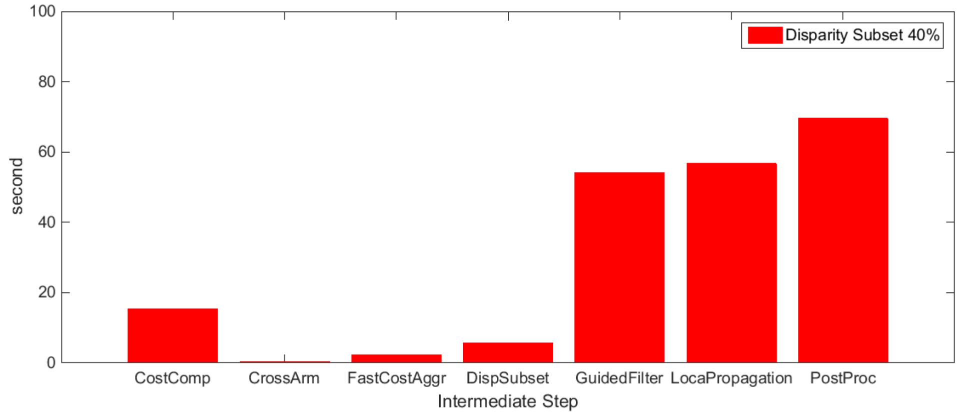


Fig. 4.19: Average Processing Time for the Middlebury Version 3 Quarter Resolution Training Images for 7 Intermediate Steps: Cost Computation, Cross-Based Support Region Computation, Fast Coarse Cost Aggregation, Disparity Subset Computation, Sparse Guided-Filter-Based Cost Aggregation, Sparse Localized Support-Region-Based Propagation, and Disparity Refinement using Weighted Propagation.

agation. From the bar chart, it can be observed that the sparse guided filter, localized support-region-based propagation and weighted propagation takes most of the processing time. But all the steps in the proposed stereo matching method can be easily implemented using a GPU platform to show real-time complexity and performance.

4.4 Conclusion

In this chapter, a novel local stereo matching algorithm using sparse locally adaptive cost aggregation (SLAC) is proposed. The proposed SLAC local stereo matching method consists of a fast initial cost aggregation stage followed by a refined cost aggregation that is only performed over a sparse subset of disparities. In the proposed method, the cost aggregation is performed in a locally adaptive manner by adapting the support region to the local image intensity and structure. In order to reduce outlier disparity values that correspond to mis-matches, a novel sparse disparity subset selection method is proposed by assigning a significance status to candidate disparity values, and selecting the significant disparity values adaptively. An adaptive guided filtering method using the disparity subset for refined cost aggregation and disparity calculation is demonstrated. The aggregated cost volume is optimized through semi-global optimization. The unreliable disparity pixels are further refined using a novel weight propagation with disparity and smoothness penalty terms. The proposed stereo matching system is tested on the Middlebury benchmark dataset (version 2 and version 3) and the KITTI 2015 stereo benchmark dataset. The analysis of system parameters demonstrates that the proposed SLAC stereo matching method is capable to generate accurate disparity results that outperform existing state-of-art local stereo matching methods.

Chapter 5

CONCLUSION

This work addresses locally adaptive stereo matching methods for different imaging setups and applications. Two novel methods are proposed for 3D reconstruction based on stereo images. The area of stereo 3D reconstruction is a promising area with more research problems to be investigated further. This chapter summarizes the main contributions of this dissertation and suggests possible research extensions.

5.1 Contributions

This dissertation presents novel locally adaptive stereo matching methods for different imaging setups and applications. The main contributions of this dissertation can be summarized as follows:

- An automatic, stereo vision based, in-line ball height and coplanarity inspection method is presented. The proposed method is computationally efficient compared to other image processing techniques for solder ball height and coplanarity detection and is shown to exhibit high accuracy, repeatability and reproducibility.

  1. A novel imaging setup is proposed based on an angled-stereo scheme with easy-to-setup and low-cost equipment. The proposed imaging setup is compatible with real-time and in-line solder ball height inspection. The proposed imaging setup is easy to calibrate using existing vision-based off-line calibration toolboxes.

    2. The proposed stereo matching and 3D reconstruction method is independent of prior depth references and active 3D sensor data. A novel iso-contour-based feature detection and matching algorithm is proposed for textureless objects and homogeneous object regions. The textureless regions of stereo images are represented with a novel iso-contour tree structure and the sparse stereo matching is represented as a tree structure matching problem for the first time. The proposed 3D reconstruction algorithm of ball height and coplanarity from the 2D features outperforms existing methods in terms of both the computational efficiency and the estimation accuracy.

- A novel local stereo matching algorithm using sparse locally adaptive cost aggregation (SLAC) is proposed for natural real-world image scenes. The performance of the proposed SLAC method is tested on the Middlebury and the KITTI benchmark datasets showing that the proposed SLAC method outperforms existing local methods in terms of accuracy.

    1. A novel adaptive support region computation method is proposed by adapting the support region to the local image intensity and structure. The local color intensity variance is adopted in the support region arm length calculation to generate a more robust support region for various scenes.

    2. A novel sparse disparity subset selection method is proposed in order to reduce outlier disparity values that correspond to mis-matches. An adaptive guided filtering method is demonstrated using the disparity subset for refined cost aggregation and disparity calculation. It is shown that using the robust disparity subset for cost aggregation helps in removing the matching ambiguity in disparity calculation, and outperforms methods

that make use of the full disparity range, in terms of computing time and disparity accuracy.

3. A novel and efficient disparity refinement algorithm is proposed to remove the disparity outliers using a multi-directional weighted cost propagation method with disparity change and smoothness penalty constraints. The proposed disparity refinement method helps to identify and correct the mis-matched disparity outliers in homogeneous and slanted regions.

5.2 Future Directions

The area of stereo 3D reconstruction is a promising area with more research problems to be investigated further. The following items summarize possible research directions to be further studied and implemented based on the work presented in this dissertation.

- Stereo Vision Based Automated Solder Ball Height and Substrate Coplanarity Inspection

  1. The proposed solder ball height and coplanarity detection system can be further enhanced and improved using higher resolution stereo images of BGA packages by more sophisticated area-scan CCD cameras. With more detailed features of the solder ball surface in 2D images, the accuracy of stereo matching can be improved.
  2. The stereo matching accuracy is sensitive to the camera angles in the stereo setup and relative position of corresponding feature points in stereo images. Therefore, there is a need to conduct a parametrized performance analysis using different camera angles and different rectification methods.

   3. The proposed iso-contour based feature detection and matching method could be applied to help in detecting the substrate feature points. This will generate a more accurate substrate ball boundary for each solder ball, and substrate feature points could be matched with higher sub-pixel accuracy by calculating the centroid of the substrate ball boundary.

   4. The proposed iso-contour based feature detection and height calculation method could be tested on different industrial packages including wafer bumps and components on integrated circuit boards.

- Sparse Locally Adaptive Image-Guided Cost Aggregation For Stereo Matching

   1. Recent advances in deep learning show that the learned features obtained from the multi-layer convolutional neural network (CNN) using a sufficient training dataset could be very useful in computer vision related tasks. Several researchers have shown that using the trained deep learning features for stereo cost computation is beneficial to improve the disparity estimation accuracy. Combining the deep CNN with the disparity subset computation could lead to a better disparity estimation.

   2. Since local stereo matching methods are easy to implement and parallelize, the computational speed of the proposed SLAC stereo matching method can be expedited by the GPU implementation with parallel computing architecture.

   3. The proposed idea of sparse disparity subset can be adopted to different stereo matching frameworks including local stereo cost aggregation, semi-global stereo matching and global-optimization-based stereo matching methods. This could lead to improvements in the stereo matching accuracy and efficiency.

4. In the localized-support-region-based optimization and weighted cost propagation procedure, the disparity change and smoothness penalty terms could be estimated adaptively according to the local scene and image characteristics. This could help in increasing the cost aggregation accuracy at depth discontinuities as well as in large homogeneous regions.

REFERENCES

[1] K. Zhang, J. Lu, and G. Lafruit, “Cross-based local stereo matching using orthogonal integral images,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 7, pp. 1073–1079, 2009.

[2] N. Cornelis, B. Leibe, K. Cornelis, and L. V. Gool, “3D urban scene modeling integrating recognition and reconstruction,” *International Journal of Computer Vision*, vol. 78, no. 2-3, pp. 121–141, 2007.

[3] E. Trucco, Y. Petillot, I. Ruiz, K. Plakas, and D. Lane, “Feature tracking in video and sonar subsea sequences with applications,” *Computer Vision and Image Understanding*, vol. 79, no. 1, pp. 92–122, 2000.

[4] M. G. Mostafa, E. E. Hemayed, and A. A. Farag, “Target recognition via 3D object reconstruction from image sequence and contour matching,” *Pattern Recognition Letters*, vol. 20, no. 11, pp. 1381–1387, 1999.

[5] W. E. Lorensen and H. E. Cline, “Marching cubes: A high resolution 3D surface construction algorithm,” *ACM Siggraph Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[6] T. McInerney and D. Terzopoulos, “Deformable models in medical image analysis: A survey,” *Medical Image Analysis*, vol. 1, no. 2, pp. 91–108, 1996.

[7] J. F. Guo, Y. L. Cai, and Y. P. Wang, “Morphology-based interpolation for 3D medical image reconstruction,” *Computerized Medical Imaging and Graphics*, vol. 19, no. 3, pp. 267–279, 1995.

[8] G. Vosselman and S. Dijkman, “3D building model reconstruction from point clouds and ground plans,” *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, vol. 34, no. 3/W4, pp. 37–44, 2001.

[9] C. Baillard and H. Matre, “3-D reconstruction of urban scenes from aerial stereo imagery: A focusing strategy,” *Computer Vision and Image Understanding*, vol. 76, no. 3, pp. 244–258, 1999.

[10] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, “An iterative regularization method for total variation-based image restoration,” *Multiscale Modeling And Simulation*, vol. 4, no. 2, pp. 460–489, 2005.

[11] H. Baker, D. Tanguay, I. Sobel, D. Gelb, M. E. Goss, W. B. Culbertson, and T. Malzbender, “The coliseum immersive teleconferencing system,” *Proceedings of International Workshop on Immersive Telepresence*, vol. 6, pp. 1–4, 2002.

[12] C. S. Kurashima, R. Yang, and A. Lastra, “Combining approximate geometry with view-dependent texture mapping - a hybrid approach to 3D video teleconferencing,” *XV Brazilian Symposium on Computer Graphics and Image Processing*, pp. 112–119, 2002.

[13] O. Morel, C. Stolz, F. Meriaudeau, and P. Gorria, "Active lighting applied to three-dimensional reconstruction of specular metallic surfaces by polarization imaging," *Applied Optics*, vol. 45, no. 17, pp. 4062–4068, 2006.

[14] J. Ghring, "Dense 3D surface acquisition by structured light using off-the-shelf components," *Photonics West 2001-Electronic Imaging*, pp. 220–231, 2000.

[15] N. A. Borghese, G. Ferrigno, G. Baroni, A. Pedotti, S. Ferrari, and R. Savare, "Autoscan: A flexible and portable 3D scanner," *IEEE Computer Graphics and Applications*, vol. 18, no. 3, pp. 38–41, 1998.

[16] C. D. Barry, C. P. Allott, N. W. John, P. M. Mellor, P. A. Arundel, D. S. Thomson, and J. C. Waterton, "Three-dimensional freehand ultrasound: Image reconstruction and volume analysis," *Ultrasound in medicine and biology*, vol. 23, no. 8, pp. 1209–1224, 1997.

[17] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," *IEEE International Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 1–1, 2003.

[18] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *ACM Siggraph Computer Graphics*, vol. 21, no. 4, pp. 163–169, 1987.

[19] C. Loop and Z. Zhang, "Computing rectifying homographies for stereo vision," *IEEE International Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 125–131, 1999.

[20] C. Harris and M. Stephens, "A combined corner and edge detector," *Proceedings of the 4th Alvey Vision Conference*, vol. 15, pp. 147–151, 1998.

[21] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–100, 2004.

[22] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "SURF: Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[23] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," *2011 IEEE International Conference on Computer Vision*, pp. 2548–2555, 2011.

[24] E. Rosten, R. Porter, and T. Drummond, "Faster and better: A machine learning approach to corner detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 1, pp. 105–119, 2010.

[25] C. Prakash, J. Li, F. Akhbari, and L. J. Karam, "Sparse depth calculation using real-time key-point detection and structure from motion for advanced driver assist systems," *Advances in Visual Computing*, pp. 740–751, 2014.

[26] A. R. Kalukin and V. Sankaran, "Three dimensional visualization of multilayered assemblies using x-ray laminography," *IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part A*, vol. 20, no. 3, pp. 361–366, 1997.

[27] P. Kim and S. Rhee, "Three dimensional inspection of ball grid array using laser vision system," *IEEE Transactions on Electronics Packaging Manufacturing*, vol. 22, no. 2, pp. 151–155, 1999.

[28] G. Udupa, M. Singaperumal, R. S. Sirohi, and M. P. Kothiyal, "Characterization of surface topography by confocal microscopy: I. principles and the measurement system," *Measurement Science and Technology*, vol. 11, no. 3, p. 315, 2000.

[29] M. Dong, R. Chung, Y. Zhao, and E. Lam, "Height inspection of wafer bumps without explicit 3D reconstruction," *Proceedings of SPIE the International Society for Optical Engineering*, vol. 6070, p. 607004, 2006.

[30] Y. J. Puah, H. W. Tang, and S. P. Teo, "Method and apparatus for 3-dimensinal vision and inspection of ball and like protrusions of electronic components," Patent US20 100 328 435 A1, 2010.

[31] O. Collet-Beillon, "Method of inspecting an array of solder ball connections of an integrated circuit module," Patent EP0 638 801 A1, 1995.

[32] T. Liu and Z. P. Fang, "Inspection system," Patent US20 090 123 060 A1, 2009.

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

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

[35] J. Sun, N. N. Zheng, and H. Y. Shum, "Stereo matching using belief propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 787–800, 2003.

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

[37] O. Veksler, "Stereo correspondence by dynamic programming on a tree," *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 384–390, 2005.

[38] (2014) Middlebury stereo dataset version 2 and version 3. http://vision.middlebury.edu/stereo/data/.

[39] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," *IEEE International Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3061–3070, 2015.

[40] J. Lu, K. Shi, D. Min, L. Lin, and M. N. Do, "Cross-based local multipoint filtering," *IEEE International Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 430–437, 2012.

[41] D. Min, J. Lu, and M. N. Do, "Joint histogram based cost aggregation for stereo matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 2539–2545, 2013.

[42] Q. Yang, P. Ji, D. Li, S. Yao, and M. Zhang, "Fast stereo matching using adaptive guided filtering," *Image and Vision Computing*, vol. 32, no. 3, pp. 202–211, 2014.

[43] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.

[44] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2002.

[45] R. C. Bolles, H. H. Baker, and D. H. Marimont, "Epipolar plane image analysis: An approach to determining structure from motion," *International Journal of Computer Vision*, vol. 1, no. 1, pp. 7–55, 1987.

[46] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 4, p. 353363, 1993.

[47] Y. Sumi, Y. Yoshimi, and F. Tomita, "3D object recognition in cluttered environments by segment-based stereo vision," *International Journal of Computer Vision*, vol. 46, no. 1, pp. 5–23, 2002.

[48] A. Broggi, C. Caraffi, R. I. Fedriga, and P. Grisleri, "Obstacle detection with stereo vision for off-road vehicle navigation," *IEEE Computer Vision and Pattern Recognition Workshops*, pp. 65–65, 2005.

[49] S. Lee and Y. Kay, "An accurate estimation of 3-D position and orientation of a moving object for robot stereo vision: Kalman filter approach," *IEEE International Conference on Robotics and Automation*, pp. 414–419, 1990.

[50] H. Ding, R. E. Powell, C. R. Hanna, and I. C. Ume, "Measurement comparison using shadow moiré and projection moiré methods," *IEEE Transactions on Components and Packaging Technologies*, vol. 25, no. 4, pp. 714–721, 2002.

[51] S. Wang, C. Quan, and C. J. Tay, "Optical micro-shadowgraph-based method for measuring micro-solderball height," *Optical Engineering*, vol. 44, no. 5, p. 050506, 2005.

[52] Z. Z. Wang, X. Y. Huang, R. G. Yang, and Y. M. Zhang, "Measurement of mirror surfaces using specular reflection and analytical computation," *Machine Vision and Applications*, vol. 24, no. 2, pp. 289–304, 2013.

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

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

[55] Z. Wang and Z. Zheng, "A region based stereo matching algorithm using cooperative optimization," *IEEE International Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008.

[56] Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," *The Proceedings of the 7th IEEE International Conference on Computer Vision*, vol. 1, pp. 666–673, 1999.

[57] (2008) Camera calibration toolbox for matlab website:. http://www.vision.caltech.edu/bouguetj/calib_doc/index.html.

[58] Z. Z. Wang and Y. M. Zhang, "Robust and automatic segmentation of a class of fuzzy edge images," *International Journal of Modelling, Identification and Control*, vol. 12, no. 1-2, pp. 88–95, 2011.

[59] A. F. Said, B. L. Bennett, L. J. Karam, and J. S. Pettinato, "Automated detection and classification of non-wet solder joints," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 1, pp. 67–80, 2011.

[60] B. Hong and M. Brady, "A topographic representation for mammogram segmentation," *Medical Image Computing and Computer-Assisted Intervention*, vol. 2879, pp. 730–737, 2003.

[61] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische Mathmatik*, vol. 14, no. 5, pp. 403–420, 1970.

[62] M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 993–1008, 2003.

[63] R. A. Hamzah and H. Ibrahim, "Literature survey on stereo vision disparity map algorithms," *Journal of Sensors*, p. 23, 2016.

[64] S. T. Barnard, "A stochastic approach to stereo vision," *Fifth Ntional Conference on Artificial Intelligence*, pp. 676–689, 1986.

[65] K. J. Yoon and I. S. Kweon, "Locally adaptive support-weight approach for visual correspondence search," *IEEE International Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, pp. 924–931, 2005.

[66] B. Weiss, “Fast median and bilateral filtering,” *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 519–526, 2006.

[67] S. Paris and F. Durand, “A fast approximation of the bilateral filter uisng a signal processing approach,” *European Conference on Computer Vision (ECCV)*, pp. 568–580, 2006.

[68] F. Porikli, “Constant time O(1) bilateral filtering,” *IEEE International Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–8, 2008.

[69] Q. Yang, “Recursive bilateral filtering,” *European Conference on Computer Vision (ECCV)*, pp. 1–8, 2012.

[70] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, “Fast cost-volume filtering for visual correspondence and beyond,” *IEEE International Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3017–3024, 2011.

[71] A. Hosni, M. Bleyer, C. Rhemann, M. Gelautz, and C. Rother, “Real-time local stereo matching using guided image filtering,” *2011 IEEE International Conference on Multimedia and Expo*, pp. 1–6, 2011.

[72] J. Jiao, R. Wang, W. Wang, S. Dong, Z. Wang, and W. Gao, “Local stereo matching with improved matching cost and disparity refinement,” *IEEE Multimedia*, vol. 21, no. 4, pp. 16–27, 2014.

[73] H. Han, X. Han, and F. Yang, “An improved gradient-based dense stereo correspondence algorithm using guided filter,” *Optik-International Journal for Light and Electron Optics*, vol. 125, no. 1, pp. 115–120, 2014.

[74] Q. Yang, D. Li, L. Wang, and M. Zhang, “Full-image guided filtering for fast stereo matching,” *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 237–240, 2013.

[75] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann, “Local stereo matching using geodesic support weights,” *IEEE International Conference on Image Processing*, pp. 2093–2096, 2009.

[76] C. Cigla and A. A. Alatan, “Information permeability for stereo matching,” *Signal Processing: Image Communications*, vol. 28, no. 9, pp. 1072–1088, 2013.

[77] K. J. He, J. Sun, and X. Tang, “Guided image filtering,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, 2013.

[78] Y. Zhan, Y. Gu, K. Huang, C. Zhang, and K. Hu, “Accurate image-guided stereo matching with efficient matching cost and disparity refinement,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1632–1645, 2016.

[79] G. A. Kordelas, D. S. Alexiadis, P. Daras, and E. Izquierdo, "Content-based guided image filtering, weighted semi-global optimization, and efficient disparity refinement for fast and accurate disparity estimation," *IEEE Transactions on Multimedia*, vol. 18, no. 2, pp. 155–170, 2016.

[80] A. Fusiello, V. Roberto, and E. Trucco, "Efficient stereo with multiple windowing," *IEEE International Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 858–863, 1997.

[81] S. B. Kang, R. Szeliski, and J. Chai, "Handling occlusions in dense multi-view stereo," *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 103–110, 2001.

[82] H. Hirschmuller, P. R. Innocent, and J. Garibaldi, "Real-time correlation-based stereo vision with reduced border errors," *International Journal of Computer Vision*, vol. 43, no. 1-3, pp. 229–246, 2002.

[83] A. Buades and G. Facciolo, "Reliable multiscale and multiwindow stereo matching," *SIAM Journal on Imaging Sciences*, vol. 8, no. 2, pp. 888–915, 2015.

[84] O. Veksler, "Stereo correspondence with compact windows via minimum ratio cycle," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1654–1660, 2002.

[85] O. Veksler, "Fast variable window for stereo correspondence using integral images," *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 556–561, 2003.

[86] J. Lu, G. Lafruit, and F. Catthoor, "Anisotropic local high-confidence voting for accurate stereo correspondence," *International Society for Optics and Photonics in Electronic Imaging*, pp. 68 120J–68 120J, 2008.

[87] X. Mei, X. Sun, M. Zhou, S. Jiao, H. Wang, and X. Zhang, "On building an accurate stereo matching system on graphics hardware," *IEEE ICCV Workshops*, pp. 467–474, 2011.

[88] B. Wang, X. Bai, Z. Tan, and A. Higashi, "Fast edge-aware cost aggregation for stereo correspondence," *International Conference on Machine Vision Applications*, pp. 339–342, 2013.

[89] Y. Peng, G. Li, R. Wang, and W. Wang, "Stereo matching with space-constrained cost aggregation and segmentation-based disparity refinement," *International Society for Optics and Photonics in Electronic Imaging*, pp. 939 309–939 309, 2015.

[90] D. Min and K. Sohn, "Cost aggregation and occlusion handling with WLS in stereo matching," *IEEE Transaction on Image Processing*, vol. 17, no. 8, pp. 1431–1442, 2008.

[91] M. A. Helala and F. Z. Qureshi, "Accelerating cost volume filtering using salient subvolumes and robust occlusion handling," *Asian Conference on Computer Vision (ACCV)*, pp. 316–331, 2014.

[92] F. C. Crow, "Summed-area tables for texture mapping," *ACM SIGGRAPH Computer Graphics*, vol. 18, no. 3, pp. 207–212, 1984.

[93] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 511–518, 2001.

[94] T. Kanade, H. Kano, S. Kimura, A. Yoshida, and K. Oda, "Development of a video-rate stereo machine," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 3, pp. 95–100, 1995.

[95] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," *International Journal of Computer Vision*, vol. 2, no. 3, pp. 283–310, 1989.

[96] L. Matthies, R. Seliski, and T. Kanade, "Kalman filter based algorithms for estimating depth from image sequences," *International Journal of Computer Vision*, vol. 3, no. 3, pp. 209–236, 1989.

[97] E. P. Simoncelli, E. H. Adelson, and D. J. Heeger, "Probability distributions of optic flow," *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 310–315, 1991.

[98] P. Seitz, "Using local orientation information as image primitive for robust object recognition," *Advances in Intelligent Robotics Systems Conference. International Society for Optics and Photonics*, pp. 1630–1639, 1989.

[99] D. Scharstein, "Matching images by comparing their gradient fields," *12th IAPR International Conference on Computer Vision & Amp; Image Processing*, vol. 1, pp. 572–575, 1994.

[100] S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 401–406, 1998.

[101] G. Egnal, "Mutual information as a stereo correspondence measure," *Technical Reports CIS*, p. 113, 2000.

[102] J. Kim, V. Kolmogorov, and R. Zabih, "Visual correspondence using energy minimization and mutual information," *International Conference on Computer Vision*, pp. 1033–1040, 2003.

[103] P. Paclik, J. Novovicova, and R. P. W. Duin, "Building road-sign classifiers using a trainable similarity measure," *IEEE Transaction on Intelligent Transportation Systems*, vol. 7, no. 3, pp. 309–321, 2006.

[104] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," *European Conference on Computer Vision (ECCV)*, pp. 151–158, 1994.

[105] C. Stentoumis, L. Grammatikopoulos, I. Kalisperakis, and G. Karras, "On accurate dense stereo-matching using a local adaptive multi-cost approach," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 91, pp. 29–49, 2014.

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

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

[108] G. A. Kordelas, D. S. Alexiadis, P. Daras, and E. Izquierdo, "Enhanced disparity estimation in stereo images," *Image and Vision Computing*, vol. 35, pp. 31–49, 2015.

[109] J. G. Robson and N. Graham, "Probability summation and regional variation in contrast sensitivity across the visual field," *Vision Research*, vol. 21, no. 3, pp. 409–418, 1981.

[110] R. Ferzli and L. J. Karam, "A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB)," *IEEE Transactions on Image Processing*, vol. 18, no. 4, pp. 717–728, 2009.

[111] Q. Yang, "A non-local cost aggregation method for stereo matching," *IEEE International Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1402–1409, 2012.

[112] Q. Yang, "Recursive bilateral filtering," *European Conference on Computer Vision (ECCV)*, pp. 399–413, 2012.

[113] N. Einecke and J. Eggert, "A two-stage correlation method for stereoscopic depth estimation," *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on*, pp. 227–234, 2010.