Chapter 3
Stereo Matching: Fundamentals, State-of-the-Art, and Existing Challenges

Chuang-Wei Liu, Hengli Wang, Sicen Guo, Mohammud Junaid Bocus, Qijun Chen, and Rui Fan

Abstract Stereo matching is the process of generating dense correspondences in stereo images in order to create a disparity map for depth perception. Stereo matching is different from flow estimation task due to stereo rectification, which ensures that correspondences are always co-linear in a pair of stereo images. Stereo vision has become increasingly popular in mobile devices, such as autonomous cars and unmanned aerial vehicles, thanks to recent advances in full-feature embedded microcomputers. However, due to limited computing resources, there is a growing need for stereo matching algorithms that strike a balance between disparity estimation accuracy and efficiency. Challenges in this field include the lack of disparity ground truth, domain adaptation, and intractable areas such as occlusions. This chapter covers the fundamentals of stereopsis, including the perspective camera model and epipolar geometry, and reviews the most advanced stereo matching algorithms. It also explores

C.-W. Liu · S. Guo · Q. Chen · R. Fan (✉)
The Machine Intelligence and Autonomous Systems (MIAS) Group, the Robotics and Artificial Intelligence Laboratory (RAIL), State Key Laboratory of Intelligent Autonomous Systems, The Department of Control Science and Engineering, and Frontiers Science Center for Intelligent Autonomous Systems, Tongji University, Shanghai 201804, People's Republic of China
e-mail: rfan@tongji.edu.cn

C.-W. Liu
e-mail: cwliu@tongji.edu.cn

S. Guo
e-mail: guosicen@tongji.edu.cn

Q. Chen
e-mail: qjchen@tongji.edu.cn

H. Wang
The Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong SAR, China
e-mail: hwangdf@connect.ust.hk

M. J. Bocus
University of Bristol, Bristol, United Kingdom
e-mail: junaid.bocus@bristol.ac.uk

R. Fan et al. (eds.), *Autonomous Driving Perception*, Advances in Computer Vision and Pattern Recognition, https://doi.org/10.1007/978-981-99-4287-9_3

disparity confidence measures, disparity estimation evaluation metrics, and publicly available datasets and benchmarks, before summarizing the outstanding challenges in this field.

3.1 Introduction

Stereo matching is one of the most important tasks in computer vision, drawing the attention of many researchers for its wide range of applications [1], including autonomous driving [2–4], unmanned aerial vehicles [5], road condition assessment [6–10], mobile robot navigation [11, 12], and remote sensing [13]. In addition to the rapid development of stereo matching algorithms, there has been a parallel effort to create public benchmarks and datasets [14–17] that can serve as baselines for researchers to evaluate the effectiveness of their own stereo matching algorithms. With the advancement of machine learning techniques, stereo matching algorithms have been able to achieve exceptional accuracy on public benchmarks [18]. However, many challenges remain unresolved, hindering more accurate disparity estimation and wider application [19]. These challenges include disparity estimation in occluded regions [20] and unsupervised training [21].

This chapter aims to offer researchers interested in stereopsis a thorough introduction to the fundamentals of stereo matching, which includes the perspective camera model and the epipolar geometry of a stereo system. Section 3.4 covers a detailed discussion of the latest stereo matching algorithms, which are classified into two categories: explicit programming-based and deep learning-based. Then, in Sect. 3.5, we introduce the disparity confidence measure, which is a commonly used auxiliary task in stereo matching due to its ability to provide prior knowledge of the degree of difficulty in disparity estimation. In Sect. 3.6, we discuss the metrics used for evaluating the accuracy and efficiency of disparity estimation. We also introduce several publicly available stereo matching datasets and commonly-used benchmarks in Sect. 3.7. Finally, in Sect. 3.8, we summarize the four main challenges that currently exist in the stereo matching task. This chapter provides a comprehensive overview of the field of stereo matching and serves as a valuable resource for researchers conducting related studies.

3.2 One Camera

3.2.1 Perspective Camera Model

The perspective camera model projects observed points in the world onto its imaging plane through the camera center. As illustrated in Fig. 3.1, $\boldsymbol{o}^{\mathrm{C}}$ denotes the camera center; $\boldsymbol{\Pi}$ and $\hat{\boldsymbol{\Pi}}$ represent the imaging plane and the normalized imag-

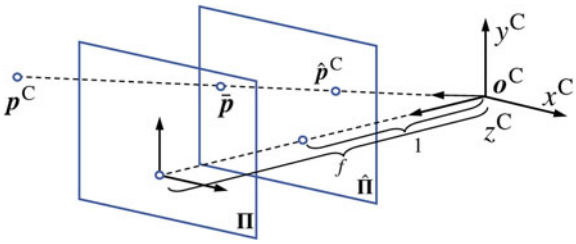

Fig. 3.1 Illustration of the perspective camera model, in which light travels in a straight line

ing plane, respectively; f represents the camera focal length. The optical axis is the ray originating at $\boldsymbol{o}^{\mathrm{C}}$ and passing perpendicularly through $\boldsymbol{\Pi}$. An observed 3D point $\boldsymbol{p}^{\mathrm{C}} = [x^{\mathrm{C}}, y^{\mathrm{C}}, z^{\mathrm{C}}]^{\top}$ in the camera coordinate system (CCS) is projected to $\bar{\boldsymbol{p}} = [x, y, f]^{\top}$ on $\boldsymbol{\Pi}$. The geometric relationship between $\boldsymbol{p}^{\mathrm{C}}$ and $\bar{\boldsymbol{p}}$ is as follows:

$$\frac{x^{\mathrm{C}}}{x} = \frac{y^{\mathrm{C}}}{y} = \frac{z^{\mathrm{C}}}{f}. \tag{3.1}$$

$\hat{\boldsymbol{p}}^{\mathrm{C}} = [\frac{x^{\mathrm{C}}}{z^{\mathrm{C}}}, \frac{y^{\mathrm{C}}}{z^{\mathrm{C}}}, 1]^{\top}$ is the normalized coordinates of $\boldsymbol{p}^{\mathrm{C}}$ on $\hat{\boldsymbol{\Pi}}$, and thus (3.1) can be rewritten as follows [22]:

$$\bar{\boldsymbol{p}} = f\hat{\boldsymbol{p}}^{\mathrm{C}} = \frac{f}{z^{\mathrm{C}}}\boldsymbol{p}^{\mathrm{C}}. \tag{3.2}$$

3.2.2 *Intrinsic Matrix*

In a perspective camera model, the relationship between point $\bar{\boldsymbol{p}} = [x, y, f]^{\top}$ on the image plane $\boldsymbol{\Pi}$ and its corresponding image pixel $\boldsymbol{p} = [u, v]^{\top}$ is given by [23]:

$$u - u_o = \alpha_x x, \quad v - v_o = \alpha_y y, \tag{3.3}$$

where the optical axis intersects the image plane at the image coordinates $[u_o, v_o]^{\top}$; α_x and α_y denote the effective size measured horizontally and vertically from the imaging plane $\boldsymbol{\Pi}$ to the image (in pixels per millimeter), respectively [22]. Plugging (3.2) into (3.3) yields:

$$u - u_o = \alpha_x f \frac{x^{\mathrm{C}}}{z^{\mathrm{C}}}, \quad v - v_o = \alpha_y f \frac{y^{\mathrm{C}}}{z^{\mathrm{C}}}. \tag{3.4}$$

Equation (3.4) can be rewritten as a matrix expression as follows:

$$\tilde{\boldsymbol{p}} = \frac{1}{z^{\mathrm{C}}}\boldsymbol{K}\boldsymbol{p}^{\mathrm{C}} = \frac{1}{z^{\mathrm{C}}}\begin{bmatrix} f_x & 0 & u_o \\ 0 & f_y & v_o \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x^{\mathrm{C}} \\ y^{\mathrm{C}} \\ z^{\mathrm{C}} \end{bmatrix}, \tag{3.5}$$

where $\boldsymbol{K}$ denotes the camera intrinsic matrix; $\tilde{\boldsymbol{p}} = [\boldsymbol{p}^\top, 1]^\top = [u, v, 1]^\top$ denotes the homogeneous coordinates of $\boldsymbol{p}$; $f_x = \alpha_x f$ and $f_y = \alpha_y f$. Plugging (3.5) into (3.2) yields:

$$\hat{\boldsymbol{p}}^{\mathrm{C}} = \boldsymbol{K}^{-1}\tilde{\boldsymbol{p}} = \frac{\bar{\boldsymbol{p}}}{f} = \frac{\boldsymbol{p}^{\mathrm{C}}}{z^{\mathrm{C}}}. \tag{3.6}$$

Therefore, all the 3D points on the ray originating at $\boldsymbol{o}^{\mathrm{C}}$ and passing through $\bar{\boldsymbol{p}}$ are projected on the image plane at $\bar{\boldsymbol{p}}$.

3.3 Two Cameras

3.3.1 Geometry of Multiple Images

3.3.1.1 Epipolar Geometry

The geometric constraint relationship between two perspective camera models is known as *epipolar geometry*, as illustrated in Fig. 3.2. Given two perspective camera models, $\boldsymbol{\Pi}_{\mathrm{L}}$ and $\boldsymbol{\Pi}_{\mathrm{R}}$ are the left and right image planes, respectively; $\boldsymbol{o}_{\mathrm{L}}^{\mathrm{C}}$ and $\boldsymbol{o}_{\mathrm{R}}^{\mathrm{C}}$ are the origins of the left camera coordinate system (LCCS) and the right camera coordinate system (RCCS) and the optic centers of the two camera models, respectively; $\boldsymbol{e}_{\mathrm{L}}$ and $\boldsymbol{e}_{\mathrm{R}}$ denote the left and right *epipolar line*, respectively. For a 3D point $\boldsymbol{p}^{\mathrm{W}}$ in the world coordinate system (WCS), its representations in the LCCS and RCCS are denoted by $\boldsymbol{p}_{\mathrm{L}}^{\mathrm{C}} = [x_{\mathrm{L}}^{\mathrm{C}}, y_{\mathrm{L}}^{\mathrm{C}}, z_{\mathrm{L}}^{\mathrm{C}}]^\top$ and $\boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}} = [x_{\mathrm{R}}^{\mathrm{C}}, y_{\mathrm{R}}^{\mathrm{C}}, z_{\mathrm{R}}^{\mathrm{C}}]^\top$, respectively. On the basis of 3D coordinate transformation in Appendix A, $\boldsymbol{p}_{\mathrm{L}}^{\mathrm{C}}$ can be transformed into $\boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}}$ using the rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ and the translation vector $\boldsymbol{t} \in \mathbb{R}^{3\times1}$ between the camera models:

$$\boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}} = \boldsymbol{R}\boldsymbol{p}_{\mathrm{L}}^{\mathrm{C}} + \boldsymbol{t}. \tag{3.7}$$

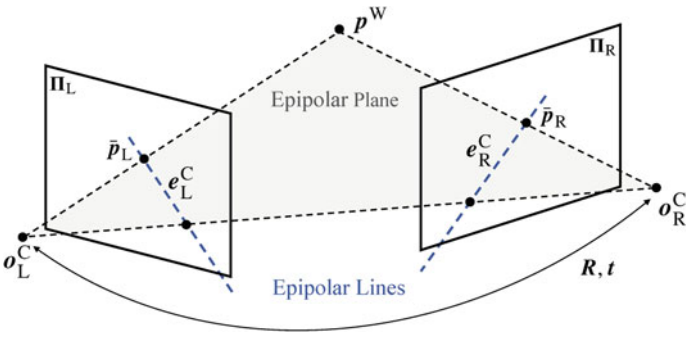

Fig. 3.2 Illustration of the epipolar geometry. The epipolar plane is uniquely defined by $\boldsymbol{o}_{\mathrm{L}}^{\mathrm{C}}$, $\boldsymbol{o}_{\mathrm{R}}^{\mathrm{C}}$, and $\boldsymbol{p}^{\mathrm{W}}$

The projection point of $\boldsymbol{p}^{\mathrm{W}}$ on Π_{L} and Π_{R} are $\bar{\boldsymbol{p}}_{\mathrm{L}} = [x_{\mathrm{L}}, y_{\mathrm{L}}, z_{\mathrm{L}}] = \frac{f_{\mathrm{L}}}{z_{\mathrm{L}}^{\mathrm{C}}} \boldsymbol{p}_{\mathrm{L}}^{\mathrm{C}}$ and $\bar{\boldsymbol{p}}_{\mathrm{R}} = [x_{\mathrm{R}}, y_{\mathrm{R}}, z_{\mathrm{R}}] = \frac{f_{\mathrm{R}}}{z_{\mathrm{R}}^{\mathrm{C}}} \boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}}$, respectively; f_{L} and f_{R} represent the focal lengths of the left and right cameras, respectively; The *epipolar plane* is defined by the camera centers $\boldsymbol{o}_{\mathrm{L}}^{\mathrm{C}}$, $\boldsymbol{o}_{\mathrm{R}}^{\mathrm{C}}$ and the 3D point $\boldsymbol{p}^{\mathrm{W}}$; The two epipolar lines represent the intersection of the epipolar plane and the two image planes Π_{L} and Π_{R}, respectively. Using (3.6), $\boldsymbol{p}_{\mathrm{L}}^{\mathrm{C}}$ and $\boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}}$ can be normalized as follows:

$$\hat{\boldsymbol{p}}_{\mathrm{L}}^{\mathrm{C}} = \frac{\boldsymbol{p}_{\mathrm{L}}^{\mathrm{C}}}{z_{\mathrm{L}}^{\mathrm{C}}} = \boldsymbol{K}_{\mathrm{L}}^{-1} \tilde{\boldsymbol{p}}_{\mathrm{L}}, \quad \hat{\boldsymbol{p}}_{\mathrm{R}}^{\mathrm{C}} = \frac{\boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}}}{z_{\mathrm{R}}^{\mathrm{C}}} = \boldsymbol{K}_{\mathrm{R}}^{-1} \tilde{\boldsymbol{p}}_{\mathrm{R}}, \tag{3.8}$$

where $\boldsymbol{K}_{\mathrm{L}}$ and $\boldsymbol{K}_{\mathrm{R}}$ are the left and right intrinsic matrices, respectively; $\tilde{\boldsymbol{p}}_{\mathrm{L}} = [\boldsymbol{p}_{\mathrm{L}}^{\top}, 1]^{\top} = [u_{\mathrm{L}}, v_{\mathrm{L}}, 1]^{\top}$ and $\tilde{\boldsymbol{p}}_{\mathrm{R}} = [\boldsymbol{p}_{\mathrm{R}}^{\top}, 1] = [u_{\mathrm{R}}, v_{\mathrm{R}}, 1]^{\top}$ are the homogeneous coordinates of the image pixels $\boldsymbol{p}_{\mathrm{L}}$ and $\boldsymbol{p}_{\mathrm{R}}$, respectively.

Considering an arbitrary 3D point $\boldsymbol{p}_0^{\mathrm{W}}$ on the ray that emanates from $\bar{\boldsymbol{p}}_{\mathrm{L}}$ and passes through $\boldsymbol{p}^{\mathrm{W}}$, it is always projected on Π_{L} at $\boldsymbol{p}_{\mathrm{L}}$ as discussed in Sect. 3.2.2. Additionally, its projection on Π_{R} is on the right epipolar line $\boldsymbol{e}_{\mathrm{R}}$, denoted by right image pixel $\boldsymbol{p}_{\mathrm{R}_0} = [u_{\mathrm{R}_0}, v_{\mathrm{R}_0}]^{\top}$. In the right image, the right epipolar line $\boldsymbol{e}_{\mathrm{R}}$ can be described by a vector $\boldsymbol{e}_{\mathrm{R}} = [a_{\mathrm{R}}, b_{\mathrm{R}}, c_{\mathrm{R}}]^{\top}$, which satisfies the following property:

$$\tilde{\boldsymbol{p}}_{\mathrm{R}_0}^{\top} \boldsymbol{e}_{\mathrm{R}} = [\boldsymbol{p}_{\mathrm{R}_0}^{\top}, 1] \boldsymbol{e}_{\mathrm{R}} = a_{\mathrm{R}} u_{\mathrm{R}_0} + b_{\mathrm{R}} v_{\mathrm{R}_0} + c_{\mathrm{R}} = 0, \tag{3.9}$$

where $\tilde{\boldsymbol{p}}_{\mathrm{R}_0} = [\boldsymbol{p}_{\mathrm{R}_0}^{\top}, 1]^{\top}$ represents the homogeneous coordinate of the image pixel $\boldsymbol{p}_{\mathrm{R}_0}$.

3.3.1.2 Fundamental Matrix

Fundamental matrix $\boldsymbol{F}$ is a 3×3 matrix that describes the epipolar geometry between stereo images. It corresponds an arbitrary image pixel $\boldsymbol{p}_{\mathrm{L}}$ in one view to a straight line in the other view. Multiplying both sides of (3.7) by $\boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}\top}[\boldsymbol{t}]_{\times}$ yields:

$$\boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}\top}[\boldsymbol{t}]_{\times} \boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}} = \boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}\top}[\boldsymbol{t}]_{\times} (\boldsymbol{R} \boldsymbol{p}_{\mathrm{L}}^{\mathrm{C}} + \boldsymbol{t}), \tag{3.10}$$

where $[\boldsymbol{t}]_{\times}$ denotes the *skew-symmetric* (or *antisymmetric*) matrix of $\boldsymbol{t}$ (see Appendix B for the definition and properties of a skew-symmetric matrix). According to (B.4), (3.10) can be rewritten as follows:

$$-\boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}\top}[\boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}}]_{\times} \boldsymbol{t} = \boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}\top}[\boldsymbol{t}]_{\times} \boldsymbol{R} \boldsymbol{p}_{\mathrm{L}}^{\mathrm{C}} + \boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}\top}[\boldsymbol{t}]_{\times} \boldsymbol{t}. \tag{3.11}$$

Applying (B.3) to (3.11) results in:

$$\boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}\top}[\boldsymbol{t}]_{\times} \boldsymbol{R} \boldsymbol{p}_{\mathrm{L}}^{\mathrm{C}} = 0. \tag{3.12}$$

Plugging (3.8) into (3.12) yields:

$$\tilde{\boldsymbol{p}}_{\mathrm{R}}^{\top} {\boldsymbol{K}_{\mathrm{R}}}^{-\top} [\boldsymbol{t}]_{\times} \boldsymbol{R} {\boldsymbol{K}_{\mathrm{L}}}^{-1} \tilde{\boldsymbol{p}}_{\mathrm{L}} = \tilde{\boldsymbol{p}}_{\mathrm{R}}^{\top} \boldsymbol{F} \tilde{\boldsymbol{p}}_{\mathrm{L}} = 0, \tag{3.13}$$

where the fundamental matrix $\boldsymbol{F}$ is defined as:

$$\boldsymbol{F} = {\boldsymbol{K}_{\mathrm{R}}}^{-\top} [\boldsymbol{t}]_{\times} \boldsymbol{R} {\boldsymbol{K}_{\mathrm{L}}}^{-1}. \tag{3.14}$$

According to (3.9), (3.13) can be rewritten as follows:

$$\tilde{\boldsymbol{p}}_{\mathrm{R}_0}^{\top} \boldsymbol{F} \tilde{\boldsymbol{p}}_{\mathrm{L}} = \tilde{\boldsymbol{p}}_{\mathrm{R}_0}^{\top} \boldsymbol{e}_{\mathrm{R}} = 0, \tag{3.15}$$

which indicates that for all points on the ray originating at $\boldsymbol{o}_{\mathrm{L}}^{\mathrm{C}}$ and passing through $\tilde{\boldsymbol{p}}_{\mathrm{L}}$, their corresponding pixels in the right image plane form the right epipolar line $\boldsymbol{e}_{\mathrm{R}}$. $\boldsymbol{e}_{\mathrm{R}}$ is uniquely defined by $\boldsymbol{p}_{\mathrm{L}}$ as follows:

$$\boldsymbol{e}_{\mathrm{R}} = \boldsymbol{F} \tilde{\boldsymbol{p}}_{\mathrm{L}}. \tag{3.16}$$

3.3.1.3 Essential Matrix

Similar to the fundamental matrix, the *essential matrix* $\boldsymbol{E} \in \mathbb{R}^{3\times3}$ depicts the point-line correspondence in the left and right normalized planes. An essential matrix can be considered as a special case of a fundamental matrix, that is, the intrinsic matrices $\boldsymbol{K}_{\mathrm{L}}$ and $\boldsymbol{K}_{\mathrm{R}}$ of two calibrated left and right cameras are already known. Thus (3.12) can be rewritten as follows:

$$\boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}\top} [\boldsymbol{t}]_{\times} \boldsymbol{R} \boldsymbol{p}_{\mathrm{L}}^{\mathrm{C}} = \boldsymbol{p}_{\mathrm{R}}^{\mathrm{C}\top} \boldsymbol{E} \boldsymbol{p}_{\mathrm{L}}^{\mathrm{C}} = 0, \tag{3.17}$$

where the essential matrix $\boldsymbol{E}$ is defined as:

$$\boldsymbol{E} = [\boldsymbol{t}]_{\times} \boldsymbol{R}. \tag{3.18}$$

Applying (3.8) to (3.17) yields:

$$\hat{\boldsymbol{p}}_{\mathrm{R}}^{\mathrm{C}\top} \boldsymbol{E} \hat{\boldsymbol{p}}_{\mathrm{L}}^{\mathrm{C}} = 0. \tag{3.19}$$

Similar to (3.16), $\boldsymbol{E} \hat{\boldsymbol{p}}_{\mathrm{L}}^{\mathrm{C}}$ denotes the parameters of the corresponding straight line in the normalized right image plane of the normalized left point $\hat{\boldsymbol{p}}_{\mathrm{L}}^{\mathrm{C}}$.

3.3.2 Stereopsis

3.3.2.1 Stereo Rectification

A pair of synchronized cameras can be used for 3D scene reconstruction, which involves determining the corresponding pixels between the left and right images. However, finding the corresponding pixel pairs on a 2D image space (optical flow estimation) is extremely time consuming. As discussed in Sect. 3.3.1.1, for a properly calibrated stereo vision system, $\boldsymbol{R}$, $\boldsymbol{t}$, $\boldsymbol{K}_{\mathrm{L}}$, and $\boldsymbol{K}_{\mathrm{R}}$ are already known, allowing the search range to be reduced to 1D along the epipolar line. In general, *stereo rectification* is applied as an image transformation process to align the two epipolar lines horizontally when the following relationship is satisfied:

$$\boldsymbol{R} = \boldsymbol{I},\ \boldsymbol{t} = [T, 0, 0]. \tag{3.20}$$

Plugging (3.20) into (3.18) yields:

$$\boldsymbol{E} = [\boldsymbol{t}]_{\times}\boldsymbol{R} = \boldsymbol{t} \times \boldsymbol{R} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -T \\ 0 & T & 0 \end{bmatrix}. \tag{3.21}$$

Considering a correspondence of normalized points $\hat{\boldsymbol{p}}_{\mathrm{L}}^{\mathrm{C}} = [x_{\mathrm{L}}, y_{\mathrm{L}}, 1]$ in the LCCS and $\hat{\boldsymbol{p}}_{\mathrm{R}}^{\mathrm{C}} = [y_{\mathrm{R}}, y_{\mathrm{R}}, 1]$ in the RCCS, applying (3.21) to (3.19) results in:

$$\hat{\boldsymbol{p}}_{\mathrm{R}}^{\mathrm{C}\top}\boldsymbol{E}\hat{\boldsymbol{p}}_{\mathrm{L}}^{\mathrm{C}} = [x_{\mathrm{L}}, y_{\mathrm{L}}, 1][0, -T, T \times y_{\mathrm{R}}]^{\top} = T \times (y_{\mathrm{R}} - y_{\mathrm{L}}) = 0, \tag{3.22}$$

which indicates that $\hat{\boldsymbol{p}}_{\mathrm{L}}^{\mathrm{C}}$ and $\hat{\boldsymbol{p}}_{\mathrm{R}}^{\mathrm{C}}$ have the same y coordinate, that is, the search range is further simplified to a horizontal row. The stereo rectification mainly involves the following steps [24]:

1. Orient the right camera with respect to the left camera using $\boldsymbol{R}$ to ensure the right image plane is parallel to the left image plane;
2. Rotate the right camera by $\boldsymbol{R}_{\mathrm{rect}}$ so that the left epipole is at infinity;
3. Reapply the same rotation to the right camera to restore the initial epipolar geometry;
4. Scale the left and right images to ensure an identical equivalent intrinsic matrix of the stereo cameras.

After the stereo rectification, the two image planes become coplanar, and conjugate epipolar lines become collinear and parallel to the horizontal image axis [22], as illustrated in Fig. 3.3, where Π'_{L} and Π'_{R} denote the rectified image planes. Therefore, the process of finding corresponding pairs is further simplified to a 1D horizontal search problem.

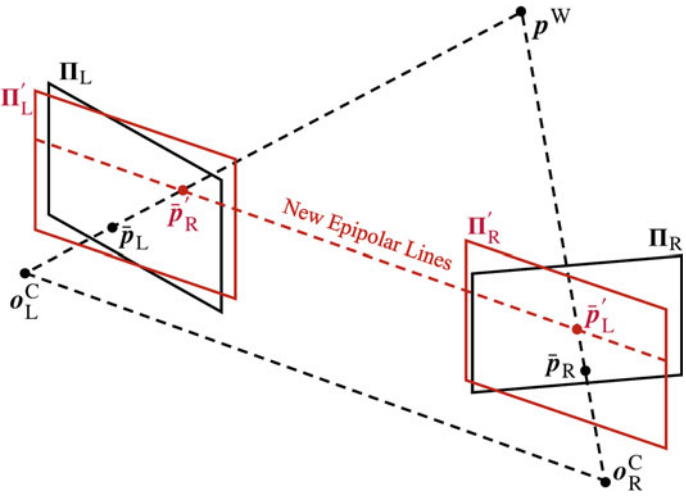

Fig. 3.3 Illustration of the stereo rectification. The left and right epipolar lines in a rectified stereo vision system are co-linear

3.3.2.2 Stereo Vision System

A properly rectified stereo vision system is depicted in Fig. 3.4, where the cameras are aligned in a parallel configuration. As a result of this alignment, the left and right cameras are considered identical, and their intrinsic matrices $\boldsymbol{K}_{\mathrm{L}}$ and $\boldsymbol{K}_{\mathrm{R}}$ are given by:

$$\boldsymbol{K}_{\mathrm{L}} = \boldsymbol{K}_{\mathrm{R}} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{3.23}$$

The length of the baseline of the stereo vision system is T_c, and the origin o^{W} of the WCS is positioned at the center of the baseline. The x, y and z-axis of the WCS are parallel to which of the LCCS and RCCS, and their x-axis are collinear. Hence, the rotation matrices between the three coordinate systems are both a third-order identity matrix $\boldsymbol{I}$, and the translation vectors from the WCS to the LCCS and the RCCS are $\boldsymbol{t}_{\mathrm{L}} = [-T_c/2, 0, 0]^{\top}$ and $\boldsymbol{t}_{\mathrm{R}} = [T_c/2, 0, 0]^{\top}$, respectively. Plugging (3.7) and (3.23) into (3.8) yields the following expressions:

$$\begin{aligned} x_{\mathrm{L}} &= f_x \frac{x^{\mathrm{W}} + T_c/2}{z^{\mathrm{W}}}, \quad y_{\mathrm{L}} = f_y \frac{y^{\mathrm{W}}}{z^{\mathrm{W}}}, \\ x_{\mathrm{R}} &= f_x \frac{x^{\mathrm{W}} - T_c/2}{z^{\mathrm{W}}}, \quad y_{\mathrm{R}} = f_y \frac{y^{\mathrm{W}}}{z^{\mathrm{W}}}. \end{aligned} \tag{3.24}$$

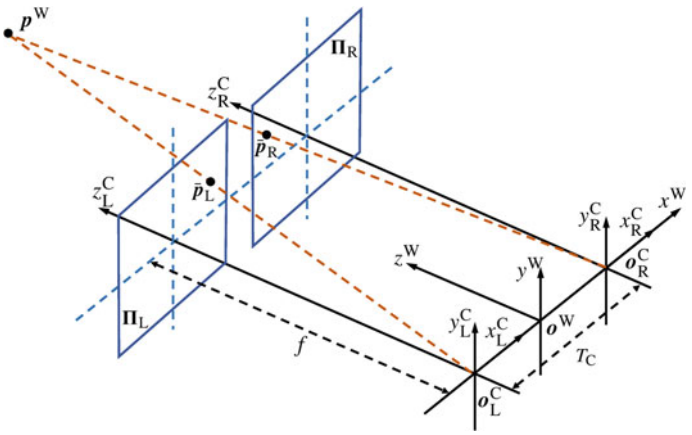

Fig. 3.4 Basic stereo vision system

For a 3D world point $\boldsymbol{p}^{\mathrm{W}} = [x^{\mathrm{W}}, y^{\mathrm{W}}, z^{\mathrm{W}}]^{\top}$, its corresponding pixel in the left and right images are denoted by $\boldsymbol{p}_{\mathrm{L}}$ and $\boldsymbol{p}_{\mathrm{R}}$, respectively. Applying (3.24) into (3.3) results in the coordinates of $\boldsymbol{p}_{\mathrm{L}}$ and $\boldsymbol{p}_{\mathrm{R}}$:

$$\boldsymbol{p}_{\mathrm{L}} = \begin{bmatrix} u_{\mathrm{L}} \\ v_{\mathrm{L}} \end{bmatrix} = \begin{bmatrix} f_x \frac{x^{\mathrm{W}}}{z^{\mathrm{W}}} + u_o + f_x \frac{T_c}{2z^{\mathrm{W}}} \\ f_y \frac{y^{\mathrm{W}}}{z^{\mathrm{W}}} + v_o \end{bmatrix}, \quad \boldsymbol{p}_{\mathrm{R}} = \begin{bmatrix} u_{\mathrm{R}} \\ v_{\mathrm{R}} \end{bmatrix} = \begin{bmatrix} f_x \frac{x^{\mathrm{W}}}{z^{\mathrm{W}}} + u_o - f_x \frac{T_c}{2z^{\mathrm{W}}} \\ f_y \frac{y^{\mathrm{W}}}{z^{\mathrm{W}}} + v_o \end{bmatrix}, \tag{3.25}$$

which further illustrates that $\hat{\boldsymbol{p}}_{\mathrm{L}}^{\mathrm{C}}$ and $\hat{\boldsymbol{p}}_{\mathrm{R}}^{\mathrm{C}}$ have the same y coordinate, as discussed in Sect. 3.3.2.1. Therefore, the disparity d (distance between co-row corresponding pixels) and depth z^{W} can be linked using:

$$d = u_{\mathrm{L}} - u_{\mathrm{R}} = f_x \frac{T_c}{z^{\mathrm{W}}}, \tag{3.26}$$

which indicates that d is inversely proportional to z^{W}. That is, when the 3D point $\boldsymbol{p}^{\mathrm{W}}$ lies away from the stereo vision system along the z-axis, its corresponding disparity d, the distance between u_{L} and u_{R}, is small.

3.4 Stereo Matching

With a properly rectified stereo vision system, the stereo matching task consists of finding the corresponding pairs of points ($\boldsymbol{p}_{\mathrm{L}}$ and $\boldsymbol{p}_{\mathrm{R}}$), and generating the dense disparity maps, as shown in Fig. 3.5. The performance of stereo matching algorithms is typically evaluated based on two main criterias: speed and accuracy. In general, there is a trade-off between speed and accuracy in stereo matching algo-

Fig. 3.5 **a** Left stereo image; **b** right stereo image; **c** left disparity map; **d** right disparity map

rithms. Well-designed algorithms tend to require more computations, resulting in a slower processing speed. The emphasis on speed or accuracy depends on the specific application of the stereo vision system [25]. For example, real-time performance is crucial for stereo vision systems used in autonomous driving [23], while accuracy is more important for indoor environment modeling [26]. Although advances in hardware are likely to reduce processing time in the future, improvements in algorithms and software are still significant.

Stereo matching algorithms can be divided into two classes: explicit programming-based and machine learning-based. The former considers disparity estimation as a local block matching problem or a global energy minimization problem [27], while the latter considers disparity estimation as a regression problem [28].

3.4.1 Explicit Programming-Based Stereo Matching Algorithms

Explicit programming-based stereo matching algorithms can be categorized into three types: local, global, and semi-global [29]. Local algorithms determine the disparity by finding the lowest cost or highest correlation, using a strategy called winner-takes-all (WTA). Global stereo matching algorithms formulate the problem of disparity estimation as a global energy minimization problem. This can be solved using Markov random fields (MRF)-based optimization algorithms [30] such as graph-cut (GC) [31] and dynamic programming (DP) [32]. Semi-global matching (SGM) [33, 34] approximates the MRF inference by performing cost aggregation along all directions in the image, which greatly improves the trade-off between the accuracy and efficiency of stereo matching [35]. Generally, the explicit programming-based stereo matching algorithms consist of four main steps: (1) cost computation, (2) cost aggregation, (3) disparity optimization and (4) disparity refinement [36].

3.4.1.1 Cost Computation

For all the pixels in the stereo images, each disparity candidate is associated with a matching cost. Generally, cost computation is used for evaluating the similarity of correspondence pairs, and a lower cost indicates a higher matching probability. Distance formulas including Manhattan distance and Euclidean distance are widely applied in cost computation, such as the absolute difference cost c_{AD} and the squared difference cost c_{SD}, which are defined as follows [37]:

$$\begin{aligned} c_{\mathrm{AD}}(u, v, d) &= \left| I_{\mathrm{L}}(u, v) - I_{\mathrm{R}}(u - d, v) \right|, \\ c_{\mathrm{SD}}(u, v, d) &= \left(I_{\mathrm{L}}(u, v) - I_{\mathrm{R}}(u - d, v) \right)^2, \end{aligned} \tag{3.27}$$

where d represents disparity; $I_{\mathrm{L}}(u, v)$, $I_{\mathrm{R}}(u - d, v)$ denote the pixel intensities of (u, v) in the left stereo image and $(u - d, v)$ in the right stereo image, respectively.

3.4.1.2 Cost Aggregation

In regions with weak texture or repetitive patterns, the effectiveness of the WTA strategy can be significantly impaired as there may be several correspondence pairs with similar costs. In order to avoid incorrect matches, c_{agg} aggregates matching cost by weighing the matching cost of a support region centered around the pixel $\boldsymbol{p} = [u, v]^\top$ [38]:

$$c_{\mathrm{agg}}(\boldsymbol{p}, d) = w(\boldsymbol{p}, d) * C(\boldsymbol{p}, d), \tag{3.28}$$

where w denotes the support region kernel; C refers to the matching costs or correlations of all the pixels within w. A commonly adopted method for computing aggregation cost c_{agg} is by performing a convolution between w and C.

Primal methods [27] use fixed square windows centered at each pixel as support regions, and all the elements in w are 1. In this way, the aggregation process becomes a uniform box filtering, and (3.27) can be reformulated as follows:

$$\begin{aligned} c_{\mathrm{SAD}}(u, v, d) &= \sum_{\boldsymbol{q} \in \mathcal{N}_{\mathbf{p}}} \left| I_{\mathrm{L}}(u, v) - I_{\mathrm{R}}(u - d, v) \right|, \\ c_{\mathrm{SSD}}(u, v, d) &= \sum_{\boldsymbol{q} \in \mathcal{N}_{\mathbf{p}}} \left(I_{\mathrm{L}}(u, v) - I_{\mathrm{R}}(u - d, v) \right)^2, \end{aligned} \tag{3.29}$$

where $\mathcal{N}_{\mathbf{p}}$ denotes the square window centered at $\boldsymbol{p} = [u, v]^\top$. Manhattan distance is used in SAD and SSD, which makes these methods computationally efficient but vulnerable to image intensity noise. In contrast, other more robust approaches such

as normalized cross-correlation (NCC) have become prevalent in this process. NCC can be formulated as follows [27]:

$$c_{\mathrm{NCC}}(\boldsymbol{p}, d) = \frac{1}{n\sigma_{\mathrm{L}}\sigma_{\mathrm{R}}} \sum_{\boldsymbol{q}\in\mathcal{N}_{\mathbf{p}}} \Big(I_{\mathrm{L}}(\boldsymbol{q}) - \mu_{\mathrm{L}}\Big)\Big(I_{\mathrm{R}}(\boldsymbol{q}-\boldsymbol{d}) - \mu_{\mathrm{R}}\Big), \tag{3.30}$$

where

$$\sigma_{\mathrm{L}} = \sqrt{\sum_{\boldsymbol{q}\in\mathcal{N}_{\mathbf{p}}} \Big(I_{\mathrm{L}}(\boldsymbol{q}) - \mu_{\mathrm{L}}\Big)^2/n}, \quad \sigma_{\mathrm{R}} = \sqrt{\sum_{\boldsymbol{q}\in\mathcal{N}_{\mathbf{p}}} \Big(I_{\mathrm{R}}(\boldsymbol{q}-\boldsymbol{d}) - \mu_{\mathrm{R}}\Big)^2/n}; \tag{3.31}$$

$\boldsymbol{d} = [d, 0]^{\top}$; $\mu_{\mathrm{L,R}}$ and $\sigma_{\mathrm{L,R}}$ represent the average and standard deviation of the square windows of the stereo images, respectively; n is the number of pixels within the square window. The resulting $c_{\mathrm{NCC}} \in [-1, 1]$ signifies the similarity between the given pair of square windows. A higher c_{NCC} indicates a higher matching similarity.

In general, using a larger square window can help reduce uncertainty in disparity estimation, but it also significantly increases computation. The primal methods assume that pixels within fixed square windows share similar disparity, which is not always the case, particularly in regions with disparity discontinuities. Therefore, several adaptive cost aggregation strategies have been proposed to optimize support region generation and improve performance in these challenging regions. Consequently, numerous adaptive cost aggregation strategies have been proposed to optimize the support region generation. One such strategy [39] assigns weights to the pixels in the support region according to the color correlation. Fast bilateral stereo (FBS) [40] leverages a bilateral filter to aggregate the matching costs adaptively. FBS formulates the cost aggregation as follows:

$$c_{\mathrm{agg}}(\boldsymbol{p}, d) = \frac{\sum_{\boldsymbol{q}\in\mathcal{N}_{\mathbf{q}}} \omega_d(\boldsymbol{q})\omega_r(\boldsymbol{q})c(\boldsymbol{q}, d)}{\sum_{\boldsymbol{q}\in\mathcal{N}_{\mathbf{q}}} \omega_d(\boldsymbol{q})\omega_r(\boldsymbol{q})}, \tag{3.32}$$

where function ω_d is calculated from the spatial distance. Function ω_r reflects the color similarity.

3.4.1.3 Disparity Optimization

Unlike the WTA strategy in local algorithms, global algorithms determine the disparity by optimizing a global energy function, which typically has the following form [41]:

$$E(d) = E_{\mathrm{data}}(d) + E_{\mathrm{smooth}}(d), \tag{3.33}$$

where $E_{\mathrm{data}}(d)$ and $E_{\mathrm{smooth}}(d)$ represent a data term and a smooth term, respectively. The former denotes the global matching cost minimization, and the latter imposes

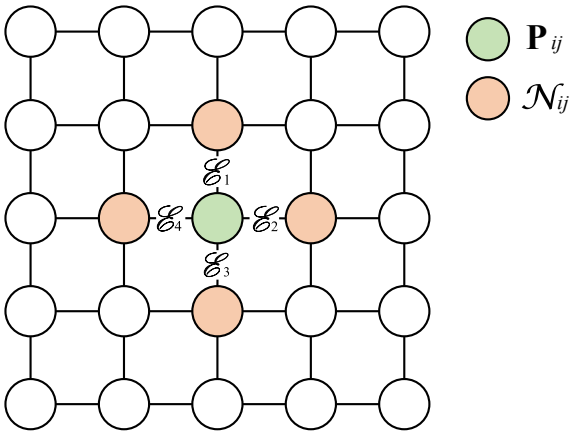

Fig. 3.6 An example of the MRF model in disparity optimization

constraints to reduce noise and errors in the cost volume, such as enforcing disparity smoothness and preserving image discontinuities [23].

Most global algorithms model the disparity optimization process with a MRF model. Figure 3.6 depicts a MRF model $\mathscr{G} = (\mathscr{P}, \mathscr{E})$, where $\mathscr{P} = \{\boldsymbol{p}_{11}, \boldsymbol{p}_{12}, \ldots, \boldsymbol{p}_{mn}\}$ denotes the vertices; $\mathscr{E} = \{(\boldsymbol{p}_{ij}, \boldsymbol{p}_{st}) \mid \boldsymbol{p}_{ij}, \boldsymbol{p}_{st} \in \mathscr{P}\}$ signifies the edges; and $\mathscr{N}_{ij} = \{\boldsymbol{q}_{1_{\boldsymbol{p}_{ij}}}, \boldsymbol{q}_{2_{\boldsymbol{p}_{ij}}}, \ldots, \boldsymbol{q}_{k_{\boldsymbol{p}_{ij}}} \mid \boldsymbol{q}_{\boldsymbol{p}_{ij}} \in \mathscr{P}\}$ represents a neighborhood system of $\boldsymbol{p}_{ij}$. Referring to the stereo matching task, all the vertices in $\mathscr{P}$ make up the $m \times n$ pixel disparity map. More specifically, $\boldsymbol{p}_{ij}$ denotes the vertex at (i, j) with a node value d_{ij} representing the disparity. In general, disparity nodes in the MRF model tend to have a stronger correlation with their neighboring nodes than other randomly chosen vertices. Therefore, (3.33) can be reformulated as follows:

$$E(\boldsymbol{p}) = \sum_{\boldsymbol{p}_{ij} \in \mathscr{P}} D(\boldsymbol{p}_{ij}, q_{\boldsymbol{p}_{ij}}) + \sum_{\boldsymbol{q}_{\boldsymbol{p}_{ij}} \in \mathscr{N}_{ij}} V(\boldsymbol{p}_{ij}, \boldsymbol{q}_{\boldsymbol{p}_{ij}}), \tag{3.34}$$

where $q_{\boldsymbol{p}_{ij}}$ represents image intensity differences; $D(\cdot)$ represents the matching cost or correlation and $V(\cdot)$ denotes the aggregation process inside a neighboring system, respectively.

Although global algorithms can achieve more accurate disparity results in textureless areas compared to local algorithms, the minimization of a global energy function which significantly increases computational requirements. Therefore, SGM [33] breaks down (3.34) into:

$$E(\boldsymbol{D}) = \sum_{\boldsymbol{p}} \left(\boldsymbol{c}(\boldsymbol{p}, d_{\boldsymbol{p}}) + \sum_{\boldsymbol{q} \in \mathscr{N}_{\mathbf{p}}} \lambda_1 \delta(|d_{\boldsymbol{p}} - d_{\boldsymbol{q}}| = 1) + \sum_{\boldsymbol{q} \in \mathscr{N}_{\mathbf{p}}} \lambda_2 \delta(|d_{\boldsymbol{p}} - d_{\boldsymbol{q}}| > 1) \right), \tag{3.35}$$

where $\boldsymbol{D}$ is the disparity map; $\boldsymbol{c}$ represents the cost volume; $\mathscr{N}_{\mathbf{p}}$ is the neighborhood system of pixel $\boldsymbol{p}$; λ_1 and λ_2 penalize its vicinities with different scales of disparity

differences, i.e., one pixel and larger than one pixel, respectively; and $\delta(\cdot)$ takes on a value of 1 with a valid argument and 0 otherwise.

3.4.1.4 Disparity Refinement

At this point, the obtained disparity map usually has some intensity noise and holes. Therefore, post-processing steps are required for refinement. The left-right disparity consistency check (LRDCC) [27] generates both left and right disparity maps $\boldsymbol{D}_{\mathrm{L}}$ and $\boldsymbol{D}_{\mathrm{R}}$, and removes pixels with inconsistent disparities, which is defined as follows:

$$|\boldsymbol{D}_{\mathrm{L}}(u, v) - \boldsymbol{D}_{\mathrm{R}}(u - \boldsymbol{D}_{\mathrm{L}}(u, v), v)| > \sigma, \tag{3.36}$$

where σ is a manually set threshold and one pixel is commonly adopted. Subpixel enhancement increases the disparity image resolution by inserting a matching cost near the initial disparity [27]. Afterwards, a median filter [42] can be applied to fill the pixels that failed to match and the holes generated by LRDCC. Additionally, other methods such as robust plane fitting, intensity consistent, and locally consistent are also commonly adopted to improve the disparity accuracy. Nevertheless, the successive application of these methods is generally contingent upon the specific application needs and the stereo matching algorithm employed.

3.4.2 Machine Learning-Based Stereo Matching Algorithms

Convolutional neural networks (CNNs) have gained significant attention in the field of stereo matching for their ability in performing effective feature extraction and information aggregation. Consequently, they have become a popular choice for disparity estimation. These algorithms are primarily data-driven, requiring huge amounts of data for the CNNs to learn to perform stereo matching. CNN-based algorithms can be classified into two types: supervised and unsupervised. Supervised methods utilize disparity ground truth to guide the training and optimize CNN parameters [34, 43, 44], while unsupervised methods use geometric constraints between stereo images as supervision during training [45–47].

3.4.2.1 Supervised Approaches

Supervised stereo matching approaches can be broadly categorized into three groups: (1) learning better feature correspondences [43], (2) learning better regularization [34], and (3) learning dense disparity estimation in an end-to-end paradigm [44].

Specifically, the first group adopts the learned representations to calculate matching costs, which are subsequently subjected to cost aggregation and regularization methods for disparity estimation. For instance, Žbontar and LeCun [43] designed a

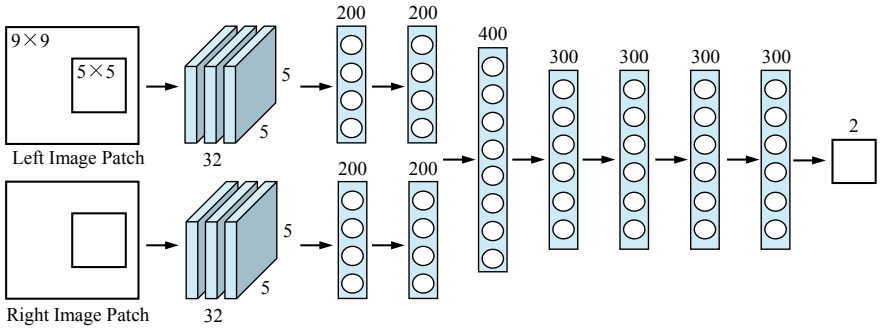

Fig. 3.7 The architecture of the CNN proposed in [43]

CNN to calculate patch-wise similarity scores, as shown in Fig. 3.7. It consists of a convolutional layer L_1 with 32 convolution kernels of size $5\times5\times1$ and seven fully connected (FC) layers L_2–L_8 with 200, 300 or 400 neurons. Two 9×9 pixel gray-scale image patches are firstly input into a sequence of L_1, L_2 and L_3, respectively. Then, the generated 200-dimensional feature maps are concatenated into a 400-dimensional feature map, before passing through L_4–L_8. The output of layer L_8 is two real numbers, which is then fed into a softmax function to generate a distribution over two classes: (1) good match and (2) bad match. Finally, explicit programming-based cost aggregation and disparity optimization methods are deployed to generate the final disparity maps. Although these approaches achieved the state-of-the-art (SoTA) accuracy at that time, the employed explicit programming-based cost aggregation methods can produce wrong predictions in occluded or texture-less/reflective regions and therefore limit their stereo matching performance [48].

Therefore, some researchers have focused on the second category of approaches, which uses CNN to improve the cost aggregation step. Ăkihito Seki and Marc Pollefeys [34] proposed SGM-Nets, which uses a 5×5-pixel gray-scale image patch as input and predicts SGM penalty parameters λ_1 and λ_2 with a CNN, as shown in Fig. 3.8. SGM-Nets consists of (1) two convolution layers, each followed by a rectified linear unit (ReLU) layer; (2) a concatenate layer to merge the two types of input; (3) two FC layers, followed by a ReLU layer and an exponential linear unit (ELU) layer, respectively; and (4) a constant layer for keeping SGM penalty values positive. Similar to SGM [33], the costs are then accumulated along four directions. In general, the outputs of SGM-Nets denote standard parameterization.

Recently, the third category of methods, i.e., end-to-end deep CNNs, has gained popularity because of their impressive performance in stereo matching tasks. For instance, GCNet [49] incorporated feature extraction (cost computation), cost aggregation and disparity optimization/refinement into a single end-to-end CNN model, as shown in Fig. 3.9. The cost volume construction method used in GCNet is different from previous methods. GCNet concatenates each feature vector with its corresponding vector extracted from the opposite stereo image across each disparity level, and packs these into a 4D volume. This approach allows GCNet to preserve more knowl-

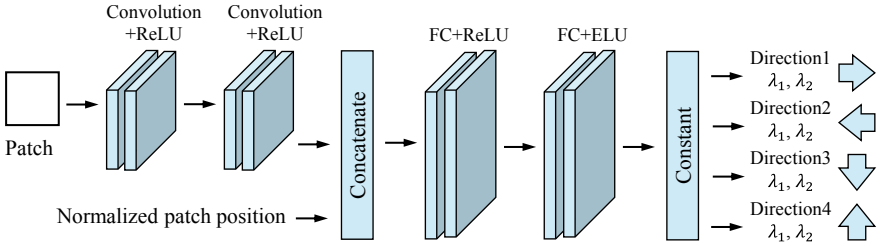

Fig. 3.8 The architecture of SGM-Nets [34]

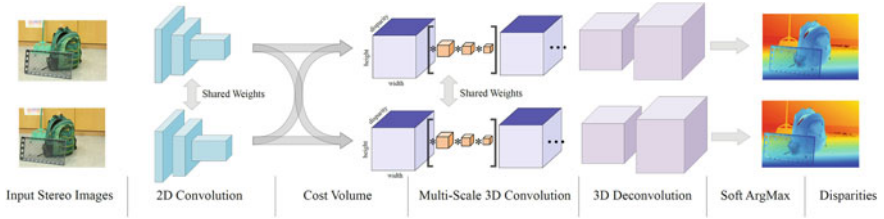

Fig. 3.9 The architecture of GCNet [49]

edge of the geometry of stereo vision, unlike previous methods. As a result, GCNet uses 3D convolution instead of 2D convolution for the disparity refinement process. Later on, Chang et al. [44] proposed Pyramid Stereo Matching Network (PSMNet), improving the feature extraction process with a spatial pyramid pooling module, and the disparity refinement process with a stacked hourglass 3D CNN, as shown in Fig. 3.10. The former enlarges the receptive fields and thus can aggregate more context information, while the latter regularizes the cost volume and further extends the regional support of context information in cost volume. In addition, Guo et al. [50] proposed group-wise correlation, which first divides the left and right features into groups and then computes correlation maps among each group. After that, the computed correlation maps are packed together for the following disparity estimation. Compared with previous cost volume construction methods by concatenation, group-wise correlation additionally introduces feature correlation into the cost volume, which avoids using more parameters to learn feature similarity in the disparity aggregation process.

In 2020, Cheng et al. [51] adopted neural architecture search techniques to obtain an effective network architecture for stereo matching. The stereo images are first processed by the Feature Net to generate visual features, which are then processed by the Matching Net to generate a 3D cost volume. The disparity map can finally be computed from the cost volume via the soft-argmin operation. Since the learnable parameters are only located in the Feature Net and the Matching Net. Cheng et al. [51] used neural architecture search techniques to select the optimal structures for them.

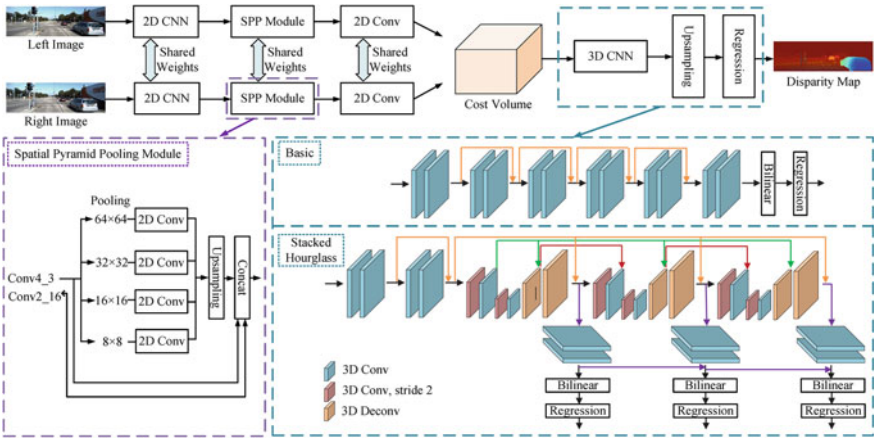

Fig. 3.10 The architecture of PSMNet [44]

Although the aforementioned end-to-end disparity estimation methods have achieved impressive results, they usually have a huge number of learnable parameters, resulting in a long inference time. To address this issue, some researchers have turned their focuses towards improving the inference speed of end-to-end methods for stereo matching. Specifically, Xu et al. [52] proposed a sparse point-based inter-scale cost aggregation module and a cross-scale cost aggregation module for efficient stereo matching. These two kinds of modules are lightweight and complementary, leading to an efficient and effective architecture for stereo matching. Inspired by RAFT [53], Wang et al. [45] presented OptStereo, which first builds multi-scale cost volumes and then iteratively updates disparity estimations at high resolution. This novel architecture can achieve a great trade-off between the accuracy and efficiency for stereo matching. The architecture of OptStereo is shown in Fig. 3.11. Later on, Lahav et al. proposed RAFT-Stereo [54], which inherits the gate recurrent unit (GRU) iteration architecture of [53], and includes a multilevel GRU to aggregate

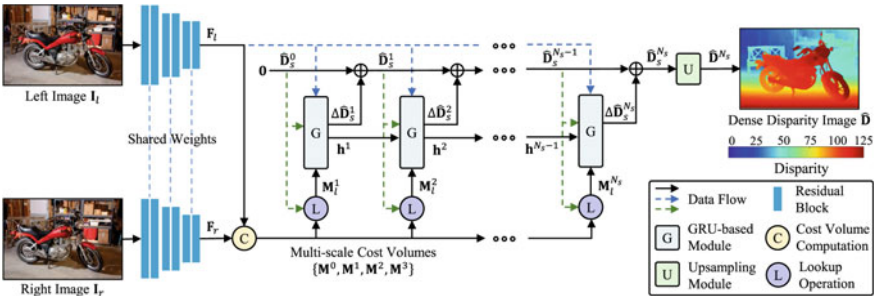

Fig. 3.11 The architecture of OptStereo [45]

feature maps at different resolutions. In 2022, Li et al. presented another recurrent structural network, CREStereo [55]. In CREStereo, the fixed-shape local search window used in previous works [45, 53, 54] is replaced with a deformable convolution operator [56] for constructing the cost volumes, which considerably improves the disparity estimation accuracy in occluded or texture-less areas. In addition, some researchers adopted the popular coarse-to-fine framework to balance the trade-off between the accuracy and efficiency for stereo matching [57–60].

3.4.2.2 Unsupervised Approaches

Although supervised approaches have achieved impressive results for disparity estimation, the collection of large amounts of disparity ground truth is time-consuming and requires significant labor. To address this issue, many researchers have focused on developing unsupervised stereo matching approaches to eliminate the need for disparity ground truth. The CNN architectures of unsupervised approaches are similar to those of supervised approaches, while the main difference lies in the network training process.

Specifically, Zhong et al. [46] proposed SsSMnet, which employs image warping error to drive the learning process, as shown in Fig. 3.12. Zhong et al. [46] leverages the following loss functions:

$$\mathcal{L} = \omega_p \left(\mathcal{L}_u^l + \mathcal{L}_u^r\right) + \omega_s \left(\mathcal{L}_s^l + \mathcal{L}_s^r\right) + \omega_c \left(\mathcal{L}_c^l + \mathcal{L}_c^r\right) + \omega_m \left(\mathcal{L}_m^l + \mathcal{L}_m^r\right), \quad (3.37)$$

where $\mathcal{L}_u^l$ and $\mathcal{L}_u^r$ denote the unary term; $\mathcal{L}_s^l$ and $\mathcal{L}_s^r$ denote the regularization term; $\mathcal{L}_c^l$ and $\mathcal{L}_c^r$ denote the consistency constraint term; $\mathcal{L}_m^l$ and $\mathcal{L}_m^r$ denote the maximization depth heuristic. The unary term is designed to minimize the discrepancy between the stereo images based on the disparity estimation, and has the following formulation:

$$\mathcal{L}_u^l \left(I_\mathrm{L}, I_\mathrm{L}'\right) = \frac{1}{N} \sum \lambda_1 \frac{1 - \mathrm{SSIM}\left(I_\mathrm{L}, I_\mathrm{L}'\right)}{2} + \lambda_2 \left|I_\mathrm{L} - I_\mathrm{L}'\right| + \lambda_3 \left|\nabla I_\mathrm{L} - \nabla I_\mathrm{L}'\right|, \quad (3.38)$$

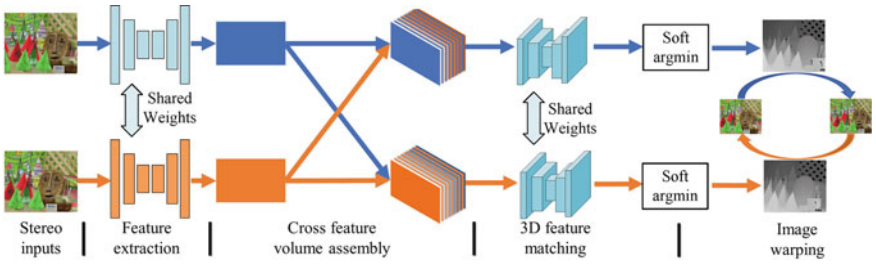

Fig. 3.12 The architecture of SsSMnet [46]

where N denotes the total number of pixels and I'_{L} denotes the reconstructed left image; SSIM$(\cdot)$ [61] measures the similarity between two images; λ_1, λ_2 and λ_3 are three weighting parameters for balancing the structural similarity, image appearance difference and image gradient difference. The regularization term is designed to make the disparity estimation locally smooth and is defined as follows:

$$\mathcal{L}_s^l = \frac{1}{N} \sum \left| \nabla_u^2 d_L \right| e^{-\left| \nabla_u^2 I_{\mathrm{L}} \right|} + \left| \nabla_v^2 d_L \right| e^{-\left| \nabla_v^2 I_{\mathrm{L}} \right|}. \tag{3.39}$$

The consistency constraint term is designed based on the concept of left-right consistency check, and has the following formulation:

$$\mathcal{L}_c^L = \left| I_{\mathrm{L}} - I''_{\mathrm{L}} \right|, \tag{3.40}$$

where I''_{L} is generated by warping the left image to the right view and warping back to the left image coordinate. Finally, the maximum depth heuristic minimizes the sum of all the disparities or maximizes the sum of all the depths, and has the following formulation:

$$\mathcal{L}_m^L = \frac{1}{N} \sum |d_L| . \tag{3.41}$$

These four terms work together to supervise the network training without the use of disparity ground truth, and are commonly used in subsequent proposed unsupervised stereo matching approaches.

Due to occlusion, certain regions in the left and right images are not commonly visible, leading to unreliable constraints for the networks' disparity estimation. This occurs as the unary term and the consistency constraint term cannot provide sufficient guidance in such areas. The unsupervised stereo matching approach presented by Zhao et al. [62] involves an random initialization of the network, followed by iterative updates of network parameters using left-right check. More specifically, in each iteration, a disparity map is predicted, which is then used to generate a confidence map based on left-right check. Pixels with confident disparity will be ignored in the next iteration. Li et al. [63] incorporated occlusion reasoning to improve the unsupervised stereo matching performance. In [63], an occlusion mask $\boldsymbol{O}$ is predicted before the disparity regression and is then used to ignore the $\mathcal{L}_u$ and $\mathcal{L}_c$ in the occluded area. The network architecture of [63] is illustrated in Fig. 3.13. Additionally, to avoid predicting an all-one occlusion map, an occlusion regularization loss is introduced, which is defined as follows:

$$\mathcal{L}_o = \frac{1}{N} \sum_{\boldsymbol{p}} \left| \boldsymbol{O}(\boldsymbol{p}) - \left(1 - e^{-\left| d_L(\boldsymbol{p}) - d_R(\boldsymbol{q}) \right|}\right) \right|, \tag{3.42}$$

where $\boldsymbol{p}$ and $\boldsymbol{q}$ represent a pixel in the left disparity map and its corresponding pixel in the right disparity map, respectively. Joung et al. [64] utilized correspondence consistency between stereo images as a basis constraint for their approach, and

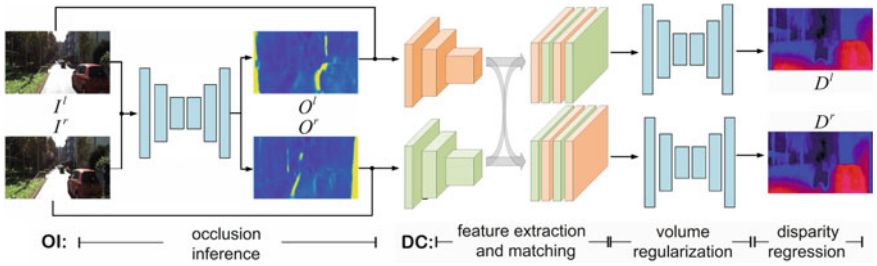

Fig. 3.13 The architecture of the occlusion-aware stereo matching network [63]

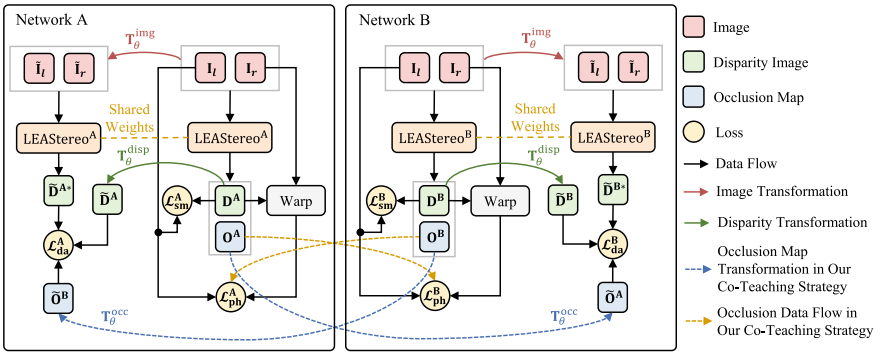

Fig. 3.14 The architecture of CoT-Stereo [47]

additionally developed a positive sample propagation scheme to enhance its stereo matching performance. In 2020, Liu et al. proposed Flow2Stereo [65], which leverages the geometric constraints presented in stereo videos to estimate disparity and optical flow simultaneously in an unsupervised manner.

However, the aforementioned unsupervised approaches still exhibit unstable performance in challenging areas, especially in regions with occlusions. This can be attributed to the sensitivity of a single network to outliers. To address this issue, Wang et al. [47] presented a co-teaching framework, where two networks teach each other about challenging regions in an unsupervised manner, as shown in Fig. 3.14. In CoT-Stereo [47], the adopted photometric loss $\mathcal{L}_{\text{ph}}$ is similar to the unary loss in (3.38), and the adopted smoothness loss is similar to the regularization loss in (3.39). To further improve the stereo matching accuracy, Wang et al. [47] adopted a data-augmentation loss, which has the following formulation [66]:

$$\begin{aligned} \mathcal{L}_{\text{da}} &= \frac{\sum l\left(\left|\mathcal{S}(\widetilde{\boldsymbol{D}}) - \widetilde{\boldsymbol{D}}^{*}\right|\right) \cdot \mathcal{S}(\widetilde{\boldsymbol{O}})}{\sum \mathcal{S}(\widetilde{\boldsymbol{O}})}, \\ l(x) &= \begin{cases} x - 0.5, & x \geq 1 \\ x^2/2, & x < 1 \end{cases}, \end{aligned} \tag{3.43}$$

where $\mathcal{S}(\cdot)$ denotes the stop-gradient; $\widetilde{\boldsymbol{D}}$, $\widetilde{\boldsymbol{D}}^*$ and $\widetilde{\boldsymbol{O}}$ denote the augmented samples. This data-augmentation paradigm can effectively enable the network to better handle occlusions. In addition, CoT-Stereo consists of a dynamic threshold selection scheme and an occlusion estimation swapping operation. The former ensures that the networks do not memorize possible outliers, while the latter enables the two networks to adaptively correct the inaccurate occlusion estimation. Inspired by [67], Fan et al. [68] proposed an occlusion-aware stereo matching algorithm based on confidence guided raw disparity fusion (CRD-Fusion). In CRD-Fusion, a correlation cost volume is firstly processed by five 3D convolutions to generate a raw disparity, which is then used to generate a raw occlusion map with consistency-based confidence measures (see Sect. 3.5.3). Afterwards, the raw disparity map and raw occlusion map are refined jointly using a hierarchical refinement module.

Another popular paradigm for unsupervised stereo matching is to generate reliable pseudo disparity ground truth for supervision. For example, Wang et al. [45] proposed OptStereo, where a pyramid voting module can generate reliable semi-dense disparity maps to supervise the CNN training. Despite recent advances in unsupervised stereo matching, there remains a significant performance gap between existing supervised and unsupervised approaches. As a result, we firmly believe that exploring more effective unsupervised training strategies is a promising direction for future research in this field.

3.5 Disparity Confidence Measures

In parallel with the rapid evolution of stereo matching algorithms, deciding the confidence of estimated disparity has also grown in popularity [69]. It is important to recognize that estimating the disparity confidence is not intended to predict potential disparity error margins. Instead, it serves as a metric for the probability that a stereo matching algorithm may fail in challenging situations such as occlusions and reflections. In other words, the disparity confidence map identifies areas that require extra attention for stereo matching.

Typically, the cost volume serves as the primary source of information for confidence measures. In this chapter, the cost volume is generated with NCC (3.30). For a left pixel $\boldsymbol{p} = [x, y]^\top$, its cost curve $c(\boldsymbol{p})$ is shown in Fig. 3.15, where $c_i(\boldsymbol{p})$ denotes the matching cost for disparity candidate i; $d_1(\boldsymbol{p})$, $d_2(\boldsymbol{p})$ and $d_{2m}(\boldsymbol{p})$ represent the disparity candidate possessing the minimum, the second minimum and the second smallest local minimum matching cost, respectively, and their corresponding matching costs are $c_{d_1}(\boldsymbol{p})$, $c_{d_2}(\boldsymbol{p})$ and $c_{d_{2m}}(\boldsymbol{p})$, respectively; $\boldsymbol{p}^r = [x - d_1(\boldsymbol{p}), y]^\top$ represents the corresponding right pixel of $\boldsymbol{p}$. However, most confidence measures only require a portion of the information in Fig. 3.15, and thus can be mainly grouped into cost-based, disparity-based, consistency-based and image-based. Example images are visualized in Fig. 3.16, where the first two columns show the different sources of information, and the last three columns present the different confidence measures.

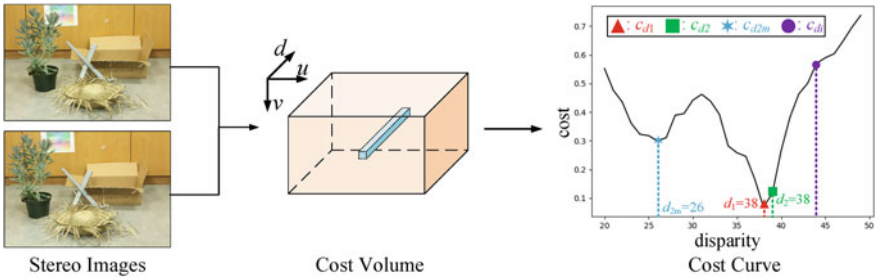

Fig. 3.15 Example of cost curve and illustrations of terms defined on the cost curve

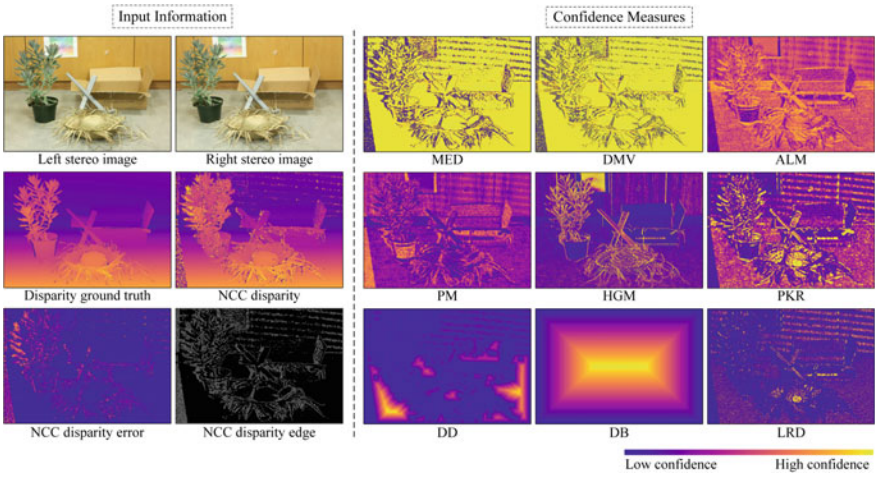

Fig. 3.16 Example images of confidence measures. The first two columns provide the sources of information, and the last three columns present the confidence measures

3.5.1 Cost-Based Confidence Measures

Cost-based confidence measures are obtained from the cost volume, mostly encoded by $c_{d_1}(\boldsymbol{p})$, $c_{d_2}(\boldsymbol{p})$ and $c_{d_{2m}}(\boldsymbol{p})$. In general, these confidence measures score a pixel with high confidence if it has a significantly lower $c_{d_1}(\boldsymbol{p})$, which indicates a strong match to its corresponding pixel in the other image.

1. The matching score metric (MSM) [70] directly uses the minimum cost $c_{d_1}(\boldsymbol{p})$ as a confidence measure, which is defined as follows:

$$\mathrm{MSM}(\boldsymbol{p}) = -c_{d_1}(\boldsymbol{p}). \tag{3.44}$$

2. Peak ratio (PKR) [70] is expressed by the ratio of $c_{d_{2m}}$ to c_{d_1}:

$$\mathrm{PKR}(\boldsymbol{p}) = \frac{c_{d_{2m}}(\boldsymbol{p})}{c_{d_1}(\boldsymbol{p})}. \tag{3.45}$$

3. The perturbation measure (PM) [71] considers the linearity of the cost curve, and is formulated as follows:

$$\mathrm{PM}(\boldsymbol{p}) = \sum_{i \neq d_1} e^{-\frac{\left[c_{d_1}(\boldsymbol{p}) - c_i(\boldsymbol{p})\right]^2}{s^2}}, \tag{3.46}$$

where s is a scaling parameter.

4. The attainable likelihood measure (ALM) [72] models the cost curve using a Gaussian distribution centered at $c_{d_1}(\boldsymbol{p})$:

$$\begin{aligned} \mathrm{ALM}(\boldsymbol{p}) &= \frac{1}{\sum_{i \in D} e^{-\frac{c_i(\boldsymbol{p})}{2\sigma(\boldsymbol{p})}}}, \\ \sigma(\boldsymbol{p}) &= \frac{\sum_{i \in D} (c_i(\boldsymbol{p}) - c_{d1}(\boldsymbol{p}))^2}{m}, \end{aligned} \tag{3.47}$$

where D represents the set of disparity candidates, and m denotes the number of disparity candidates.

3.5.2 Disparity-Based Confidence Measures

Disparity-based confidence measures use the disparity map generated with the WTA strategy from the cost volume as the input domain, without any additional cues from the cost volume.

1. The distance from discontinuities (DD) [73] metric derives confidence from the distance to a disparity discontinuity:

$$\mathrm{DD}(\boldsymbol{p}) = \min_{\boldsymbol{q} \in \hat{d}} |\boldsymbol{p} - \boldsymbol{q}|, \tag{3.48}$$

where $\hat{d}$ is obtained by applying a Canny edge detector [74] to the disparity map, as shown in row 3, column 2 of Fig. 3.16.

2. Disparity map variance (DMV) [71] measures the gradient over the disparity map:

$$\mathrm{DMV}(\boldsymbol{p}) = \|\nabla d_1(\boldsymbol{p})\| . \tag{3.49}$$

3. Difference with median disparity (MED) [73] calculates the difference between $d_1(\boldsymbol{p})$ and the median disparity in the 5×5 windows centered at $\boldsymbol{p}$ in the disparity map.

3.5.3 Consistency-Based Confidence Measures

Consistency-based confidence measures evaluate the consistency between corresponding pixels in the stereo images based on epipolar geometry.

1. Left-right checking (LRC) [75] is a method that assumes that the disparity values at corresponding pixels in the left and right views should be consistent, except in the occluded regions. Therefore, the presence of inconsistent disparities signifies the difficulty of stereo matching, particularly in regions with occlusions. LRC is defined as follows:

$$\mathrm{LRC}(\boldsymbol{p}) = -|d_1(\boldsymbol{p}) - d_1^r(\boldsymbol{p}^r)|, \tag{3.50}$$

   where d_1^r represents the right disparity generated with the WTA strategy.
2. Left-right difference (LRD) [76] further measures the consistency of the minimum costs across the stereo images:

$$\mathrm{LRD}(\boldsymbol{p}) = \frac{c_{d2}(\boldsymbol{p}) - c_{d1}(\boldsymbol{p})}{|c_{d1}(\boldsymbol{p}) - c_{d1}^r(\boldsymbol{p}^r)|}. \tag{3.51}$$

3.5.4 Image-Based Confidence Measures

Image-based confidence measures consider only the stereo images as input.

1. Distance from border (DB) [73] calculates the distance between a pixel and its nearest image border, assuming that pixels close to the image border are more likely to be occluded or unseen in the other reference view. DB is defined as follows:

$$\mathrm{DB}(\boldsymbol{p}) = \min(x, y, W - x, H - y), \tag{3.52}$$

   where W and H represent the image width and height, respectively.

2. Horizontal gradient magnitude (HGM) [71] is based on the assumption that a higher image gradient indicates a richer texture, and thus a higher matching confidence. Similar to DMV, HGM can be formulated as follows:

$$\mathrm{HGM}(\boldsymbol{p}) = \|\nabla_x I_{\mathrm{L}}(\boldsymbol{p})\|\,, \tag{3.53}$$

where $I_{\mathrm{L}}(\boldsymbol{p})$ represents the image intensity at pixel $\boldsymbol{p}$ in the left image.

3.6 Evaluation Metrics

The performance evaluation of a given stereo matching algorithm usually involves measuring both the accuracy and efficiency of the disparity estimation [25]. Commonly used metrics to evaluate the accuracy of an estimated disparity map are listed below [77]:

1. Root-mean-squared error (RMS), e_{RMS} is defined as follows [78]:

$$e_{\mathrm{RMS}} = \sqrt{\frac{1}{N}\sum_{\boldsymbol{p}\in\mathscr{P}} |\boldsymbol{D}_{\mathrm{E}}(\boldsymbol{p}) - \boldsymbol{D}_{\mathrm{G}}(\boldsymbol{p})|^2}, \tag{3.54}$$

where $\boldsymbol{D}_{\mathrm{E}}$ and $\boldsymbol{D}_{\mathrm{G}}$ denote the estimated and ground truth disparity maps, respectively; $\mathscr{P}$ represents the set of disparities used for evaluation, and N denotes the size of $\mathscr{P}$.

2. End-point error (EPE), e_{EPE}, is expressed as the average disparity estimation error across all pixels [77]:

$$e_{\mathrm{EPE}} = \frac{1}{N}\sum_{\boldsymbol{p}\in\mathscr{P}} |\boldsymbol{D}_{\mathrm{E}}(\boldsymbol{p}) - \boldsymbol{D}_{\mathrm{G}}(\boldsymbol{p})|. \tag{3.55}$$

3. Percentage of error pixels (PEP), e_{PEP}, is given by [42]:

$$e_{\mathrm{PEP}} = \frac{1}{N}\sum_{\boldsymbol{p}\in\mathscr{P}} \delta\Big(|\boldsymbol{D}_{\mathrm{E}}(\boldsymbol{p}) - \boldsymbol{D}_{\mathrm{G}}(\boldsymbol{p})| > \delta_d\Big) \times 100\%, \tag{3.56}$$

where δ_d represents the disparity evaluation tolerance.

4. D1-all, $e_{\text{D1-all}}$ takes into consideration the magnitude of disparity ground truth, and is expressed as [79]:

$$e_{\text{D1-all}} = \frac{1}{N}\sum_{\boldsymbol{p}\in\mathscr{P}} \delta\Big(|\boldsymbol{D}_{\mathrm{E}}(\boldsymbol{p}) - \boldsymbol{D}_{\mathrm{G}}(\boldsymbol{p})| > \max\{0.05 \times \boldsymbol{D}_{\mathrm{G}}(\boldsymbol{p}), 3\}\Big) \times 100\%. \tag{3.57}$$

5. The η error quantile represents the highest disparity error among the given percentage η of best matched pixels. For example, for $\eta = 50\%$, this is the median error.

In terms of evaluating the efficiency of disparity estimation, the most common approach is to measure the runtime processing per stereo image of the algorithm being evaluated. However, the runtime can vary depending on factors such as image resolution and disparity search range. Therefore, a more robust metric for evaluating stereo matching efficiency is Mde/s, which is given in millions of disparity evaluations per second [25]:

$$\text{Mde}/s = \frac{u_{\max} v_{\max} d_{\max}}{t} 10^{-6}. \tag{3.58}$$

Similarly, time/MP and time/GD measure the runtime in megapixels and gigadisparities [80], respectively.

3.7 Public Datasets and Benchamarks

Open-access datasets and benchmarks have been developed in conjunction with the measurable progress in stereo matching algorithms. These datasets cover different application scenarios, such as driving scenes [14, 81], indoor and outdoor scenes [16, 38]. Table 3.1 summarizes the existing datasets, and we provide a detailed explanation of some of the most commonly used benchmarks to assist researchers in making their choices.

3.7.1 Middlebury Benchmark

The Middlebury dataset [15, 38, 42, 88] offers high-resolution stereo images of static indoor scenes and ground truth disparity maps with high accuracy, as depicted in Fig. 3.17. The disparity ground truth is generated using a structured light acquisition system, and Middlebury 2014 [15] attains a disparity accuracy of 0.2 pixels on most observed surfaces, including in half-occluded regions. Although only dozens of training samples are provided, its highly accurate ground truth disparity and online evaluation interface make it one of the most widely used benchmarks in stereo matching. Table 3.2 displays the benchmark results of the above-mentioned stereo matching algorithms on the Middlebury 2014 dataset [15].

Table 3.1 Critical attributes of commonly adopted publicly available datasets

Dataset	Size		Resolution	Source	Scene
	Training set	Testing set			
Middlebury 2014 [a] [15]	23	10	2964 × 2000	Real-world	Indoor
Sintel [b] [82]	1064	564	1024 × 436	Synthetic	Hybrid
NYU V2 [c] [83]	1449	–	640 × 480	Real-world	Indoor
KITTI Stereo 2012 [d] [81]	194	195	1226 × 379	Real-world	Driving
KITTI Stereo 2015 [e] [14]	200	200	1242 × 375	Real-world	Driving
Virtual KITTI [f] [84]	21260	–	1242 × 375	Synthetic	Driving
Virtual KITTI 2 [g] [85]	21260	–	1242 × 375	Synthetic	Driving
Scene Flow [h] [17]	34801	4248	960 × 540	Synthetic	Hybrid
ETH3D [i] [16]	27	20	940 × 490	Real-world	Synthetic
Falling Things [j] [86]	61500	–	960 × 540	Synthetic	Indoor
Driving Stereo [k] [87]	174437	7751	1762 × 800	Real-world	Indoor

[a] https://vision.middlebury.edu/stereo/data/scenes2014/
[b] http://sintel.is.tue.mpg.de/stereo
[c] https://cs.nyu.edu/ silberman/datasets/nyu_depth_v2.html
[d] https://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=stereo
[e] https://www.cvlibs.net/datasets/kitti/eval_scene_flow.php?benchmark=stereo
[f] https://europe.naverlabs.com/research/computer-vision/proxy-virtual-worlds-vkitti-1/
[g] https://europe.naverlabs.com/research/computer-vision/proxy-virtual-worlds-vkitti-2/
[h] https://lmb.informatik.uni-freiburg.de/resources/datasets/SceneFlowDatasets.en.html
[i] https://www.eth3d.net/low_res_two_view
[j] https://research.nvidia.com/publication/2018-06_falling-things-synthetic-dataset-3d-object-detection-and-pose-estimation
[k] https://drivingstereo-dataset.github.io/

Fig. 3.17 Example images of middlebury 2014 dataset [15]. **a** Left stereo image; **b** right stereo image; **c** ground truth disparity map

Table 3.2 Online evaluation results of noc areas on Middlebury benchmark

Method	PEP (%) ↓		EPE (px) ↓	RMS (px) ↓	A50[a] (px) ↓	Time/MP (s) ↓	Time/GD (s) ↓
	$\delta_d = 1$	$\delta_d = 4$					
PSMNet [44]	63.9	23.5	6.68	19.4	1.94	2.62	32.2
GANet [48]	37.8	11.2	12.2	35.4	0.83	6.33	16.4
AANet [52]	25.5	10.8	6.37	23.5	0.56	2.48	7.07
LEAStereo [51]	20.8	2.75	1.43	8.11	0.53	2.53	7.27
AdaStereo [89]	29.5	6.35	2.22	10.2	0.65	0.13	0.38
HITNet [59]	13.3	3.81	1.71	9.97	0.40	0.11	0.29
RAFT-Stereo [54]	9.37	2.75	1.27	8.41	0.26	2.19	5.76
CREStereo [55]	8.25	2.04	1.15	7.70	0.28	0.77	2.22

[a] A50 represents 50% error quantile

3.7.2 KITTI Benchmark

The KITTI dataset [14, 81] was captured in rural and highway areas of Karlsruhe and has been commonly used in a variety of visul tasks [90], as illustrated in Fig. 3.18a, b. It comprises a rectified stereo vision system with two high-resolution color and grayscale video cameras mounted on a wagon, providing stereo images, and a Velodyne laser scanner to offer sparse ground truth disparity, as displayed in Fig. 3.18. The KITTI Stereo 2012 dataset [81] and KITTI Stereo 2015 dataset [14] both contain about 400 image pairs with accurate disparity ground truth. The KITTI dataset has been of great interest to a wide range of researchers in the field of autonomous driving due to sufficient samples for training CNNs.

Disparities for pixels that are visible in only one view due to occlusion cannot be directly obtained from stereo images, but they can be obtained from depth information (3.26). Therefore, the KITTI dataset provides both occluded (occ) and

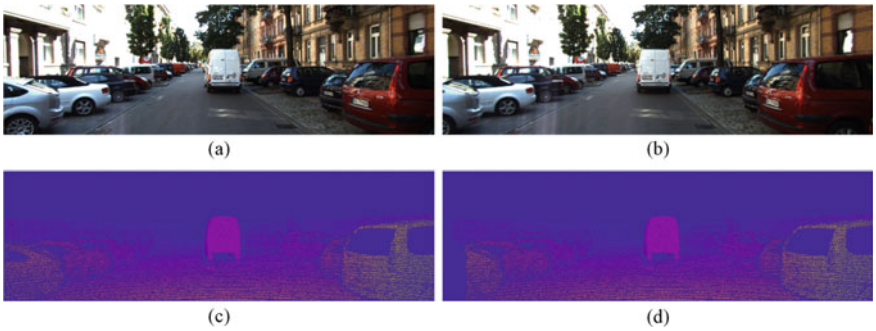

Fig. 3.18 Example images of KITTI Stereo 2015 dataset [14]. **a** Left stereo image; **b** right stereo image; **c** ground truth occluded (occ) disparity map; **d** ground truth non-occluded (noc) disparity map

Table 3.3 Online evaluation results of noc areas on KITTI Stereo 2012 dataset

Method	PEP (%) ↓				EPE (px) ↓	Runtime (s) ↓	Environment
	$\delta_d = 2$	$\delta_d = 3$	$\delta_d = 4$	$\delta_d = 5$			
GCnet [49]	2.71	1.77	–	1.12	0.6	0.90	Nvidia Titan Xp
PSMNet [44]	2.44	1.49	1.12	0.90	0.5	0.41	Nvidia Titan Xp
GANet [48]	1.89	1.19	0.91	0.76	0.4	1.80	GPU@2.5Ghz
GWCNet [50]	2.12	1.32	0.99	0.80	0.5	0.32	GPU@2.0Ghz
AANet [52]	2.30	1.55	1.20	0.98	0.4	0.06	NVIDIA V100
HITNet [59]	2.00	1.41	1.14	0.96	0.5	0.02	GPU@2.5Ghz
LEAStereo [51]	1.90	1.13	0.83	0.67	0.5	0.30	GPU@2.5Ghz
OptStereo [45]	1.91	1.20	0.92	0.77	0.4	0.10	GPU@2.5Ghz
SCV-Stereo [91]	2.01	1.27	0.97	0.81	0.5	0.08	GPU@2.5Ghz
CRD-Fusion [68]	5.07	3.35	2.66	2.26	0.9	0.02	GPU@2.5Ghz
CREStereo [55]	1.72	1.14	0.90	0.76	0.4	0.40	GPU@3.5Ghz

non-occluded (noc) ground truth disparity maps. The disparities of occluded areas are ignored in the latter, as shown in Fig. 3.18c, d. It is important to note that disparities in reflective areas such as glasses are not included. Generally, machine learning-based algorithms, which have greater generalization ability compared to explicit programming-based algorithms, are more likely to achieve better performance when using the occ ground truth disparity maps. The benchmark results for the

Table 3.4 Online evaluation results on KITTI Stereo 2015 dataset

Method	D1-all (%) ↓		Runtime (s) ↓	Environment
	occ areas	noc areas		
GCnet [49]	2.87	2.61	0.90	Nvidia Titan Xp
PSMNet [44]	2.32	2.14	0.41	Nvidia Titan Xp
GANet [48]	1.81	1.63	1.80	GPU@2.5Ghz
GWCNet [50]	2.11	1.92	0,32	GPU@2.0Ghz
AANet [52]	2.03	1.85	0.06	NVIDIA V100
HITNet [59]	1.98	1.74	0.02	GPU@2.5Ghz
LEAStereo [51]	1.65	1.51	0.30	GPU@2.5Ghz
OptStereo [45]	1.82	1.64	0.10	GPU@2.5Ghz
SCV-Stereo [91]	2.02	1.84	0.08	GPU@2.5Ghz
CRD-Fusion [68]	6.11	5.69	0.02	GPU@2.5Ghz
CREStereo [55]	1.69	1.54	0.41	GPU@3.5Ghz

Table 3.5 Online evaluation results of noc areas on ETH3D benchmark

Method	PEP (%) ↓		EPE (px) ↓	RMS (px) ↓	A50 (px) ↓	Runtime (s) ↓
	$\delta_d = 1$	$\delta_d = 4$				
PSMNet [44]	5.02	0.41	0.33	0.66	0.21	0.54
GANet [48]	6.22	0.55	0.36	0.75	0.21	1.00
GWCNet [50]	6.42	0.50	0.35	0.69	0.20	0.12
AANet [52]	5.01	0.75	0.31	0.68	0.16	1.26
AdaStereo [89]	3.09	0.20	0.24	0.44	0.15	0.40
HITNet [59]	2.79	0.19	0.20	0.46	0.10	0.02
RAFT-Stereo [54]	2.44	0.15	0.18	0.36	0.10	0.81
CREStereo [55]	0.98	0.10	0.13	0.28	0.09	–

KITTI Stereo 2012 dataset and KITTI Stereo 2015 dataset are presented in Tables 3.3 and 3.4, respectively.

Fig. 3.19 Example images of eth3d benchmark [16]. **a** Left stereo image; **b** right stereo image; **c** ground truth disparity map; **d** noc mask

3.7.3 ETH3D Benchmark

The ETH3D benchmark [16] comprises grayscale stereo images of both indoor and outdoor static scenes, including 27 training image pairs and 20 test image pairs as depicted in Fig. 3.19. The high-precision laser scanner is utilized to generate the ground truth disparity maps. Furthermore, an occlusion mask is provided for its training image pairs, encompassing non-commonly visible areas and intractable areas such as glasses in the stereo matching task, as demonstrated in Fig. 3.19d. The benchmark results on the ETH3D dataset are presented in Table 3.5.

3.8 Existing Challenges

3.8.1 Unsupervised Training

Earlier research has indicated that CNN-based stereo matching algorithms can effectively solve disparity estimation [92]. However, such data-driven algorithms require a significant amount of disparity ground truth, which is challenging to obtain in real-world scenarios. Due to the absence of large-scale stereo datasets with disparity ground truth, it is challenging to generalize supervised stereo matching algorithms to novel settings [93]. Consequently, many researchers are now focusing on unsupervised stereo matching algorithms that predict depth maps directly from the intensity images, as discussed in Sect. 3.4.2.2. Although there have been significant advances in unsupervised stereo matching algorithms, there is still a considerable gap in disparity estimation accuracy compared to supervised stereo matching algorithms [18].

3.8.2 Domain Adaptation

Another approach to reduce the reliance on disparity ground truth is domain adaptation. These algorithms can generalize stereo matching knowledge learned from synthetic datasets [17, 94, 95] to unseen scenes. DSMNet [94] proposed the domain-invariant normalization (DN) method, which normalizes feature maps along both the

spatial dimension and the channel axis to enhance the feature maps' invariance to domain differences. DSMNet also proposed a structure-preserving graph-based filter (SGF) to reduce sensitivity to local textures. Song et al. [89] further improved domain adaptation by performing image-level alignment, feature-level alignment, and output-space alignment using a color transfer method, a cost volume norm layer, and an auxiliary occlusion-aware task, respectively. In 2022, Zhang et al. [96] introduced the stereo selective whitening loss to better preserve feature consistency between domains.

3.8.3 Trade-Off Between Speed and Accuracy

Recent advances in parallel computing architectures and microcomputers have made stereo matching algorithms widely deployable on intelligent devices, such as autonomous cars. However, resource-limited hardware requires a balanced trade-off between speed and accuracy for stereo matching algorithms, which are often in opposition to each other [25]. Therefore, researchers are actively working on improving the real-time performance of stereo matching algorithms. For instance, AANet [52] replaces 3D convolutions commonly used in deep CNN-based algorithms and achieves a drastic reduction in the amount of calculation. Furthermore, in the widely adopted recurrent networks [45, 54, 55], the trade-off between accuracy and efficiency can be easily achieved with early stopping.

3.8.4 Intractable Areas

Middlebury 2002 dataset [42] has put extra emphasis on the texture-less areas and occluded areas for making the process of disparity estimation more difficult. In texture-less areas, a lack of visual features can result in pixel mismatches, which is typically addressed using the disparity smoothness constraint (3.39) [97, 98]. On the other hand, occluded areas pose a challenge due to the absence of corresponding information in the stereo images. To address this, CNN-based approaches and confidence measures discussed in Sect. 3.5 can be used to solve an auxiliary occlusion-aware task. Looser constraints are given within occluded areas during the training stage of stereo matching networks [21, 68, 89, 99]. Kim et al. [100] trained a cost aggregation network and a confidence estimation network in an adversarial structure to jointly learn disparity and confidence estimation. In 2022, [101] associated occlusions with disparity discontinuity and generated an occlusion mask from the disparity map using Sobel filter [102]. Additionally, [103] introduced an auxiliary semantic segmentation task to assist in guiding the network training for handling occlusion boundaries.

3.9 Summary

This chapter provided an overview of stereo vision and the stereo matching task. We discussed both explicit programming-based and machine learning-based stereo matching algorithms, as well as auxiliary disparity confidence measures. We also covered the metrics for evaluating disparity estimation and public stereo matching benchmarks. Finally, we summarized existing stereo matching challenges, providing researchers with valuable starting points for further studies.

Acknowledgements This work was supported by the National Key R&D Program of China under Grant 2020AAA0108100, the National Natural Science Foundation of China under Grant 62233013, and the Science and Technology Commission of Shanghai Municipal under Grant 22511104500.

Appendix

A Lie Group $SO(3)$

In the three-dimensional space, the coordinates of a 3D point in two coordinate systems are $\boldsymbol{x}_1 = [x_1, y_1, z_1]^\top$ and $\boldsymbol{x}_2 = [x_2, y_2, z_2]^\top \in \mathbb{R}^{3\times 1}$, respectively. $\boldsymbol{x}_1$ can be transformed into $\boldsymbol{x}_2$ using a rotation matrix $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$ and a translation vector $\boldsymbol{t} \in \mathbb{R}^{3\times 1}$:

$$\boldsymbol{x_2} = \boldsymbol{R}\boldsymbol{x_1} + \boldsymbol{t}, \tag{A.1}$$

where $\boldsymbol{R}$ satisfies orthogonality:

$$\boldsymbol{R}^\top = \boldsymbol{R}^{-1} \text{ and } |\det(\boldsymbol{R})| = 1, \tag{A.2}$$

where $\det(\boldsymbol{R})$ represents the determinant of $\boldsymbol{R}$. The subgroup of orthogonal matrices with $\det(\boldsymbol{R}) = +1$ is referred to as a *special orthogonal group* and is denoted as $SO(3)$.

B Skew-Symmetric Matrix

In linear algebra, a skew-symmetric matrix $\boldsymbol{A}$ satisfies the following property:

$$\boldsymbol{A} \text{ is a skew-symmetrix matrix} \Leftrightarrow \boldsymbol{A}^\top = -\boldsymbol{A}. \tag{B.1}$$

In 3D computer vision, the skew-symmetric matrix $[\boldsymbol{a}]_\times$ of a vector $\boldsymbol{a} = [a_1, a_2, a_3]$ is defined as [104]:

$$[\boldsymbol{a}]_\times = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{B.2}$$

The two properties of a skew-symmetric matrix are as follows:

$$\boldsymbol{a}^\top [\boldsymbol{a}]_\times = \mathbf{0}^\top, \ [\boldsymbol{a}]_\times \boldsymbol{a} = \mathbf{0}, \tag{B.3}$$

where $\mathbf{0} = [0, 0, 0]^\top$ is a zero vector. Furthermore, a skew-symmetric matrix can also represent cross product as matrix multiplication. Specifically, for two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$, their cross product can be expressed as [104]:

$$\boldsymbol{a} \times \boldsymbol{b} = [\boldsymbol{a}]_\times \boldsymbol{b} = -[\boldsymbol{b}]_\times \boldsymbol{a}. \tag{B.4}$$

References

1. Zhou K et al (2020) Review of stereo matching algorithms based on deep learning. Comput Intell Neurosci 2020
2. Fan R et al (2022) Learning collision-free space detection from stereo images: homography matrix brings better data augmentation. IEEE/ASME Trans Mechatron 27(1):225–233
3. Wang H et al (2022) UnDAF: a general unsupervised domain adaptation framework for disparity or optical flow estimation. In: 2022 international conference on robotics and automation (ICRA). IEEE, pp 01–07
4. Ozgunalp U et al (2017) Multiple lane detection algorithm based on novel dense vanishing point estimation. IEEE Trans Intell Transp Syst 18(3):621–632
5. Duan R et al (2022) Stereo orientation prior for uav robust and accurate visual odometry. IEEE/ASME Trans Mechatron 27(5):3440–3450
6. Fan R et al (2020) Pothole detection based on disparity transformation and road surface modeling. IEEE Trans Image Process 29:897–908
7. Ma N et al (2022) Computer vision for road imaging and pothole detection: a state-of-the-art review of systems and algorithms. Transp Safety Environ 4(4):tdac026
8. Fan R et al (2021) Graph attention layer evolves semantic segmentation for road pothole detection: a benchmark and algorithms. IEEE Trans Image Process 30:8144–8154
9. Fan R, Liu M (2020) Road damage detection based on unsupervised disparity map segmentation. IEEE Trans Intell Transp Syst 21(11):4906–4911
10. Sicen G et al (2023) Road environment perception for safe and comfortable driving. Springer submitted for publication
11. Wang H et al (2022) Dynamic fusion module evolves drivable area and road anomaly detection: a benchmark and algorithms. IEEE Trans Cybern 52(10):10 750–10 760
12. Fan R et al (2020) SNE-RoadSeg: incorporating surface normal information into semantic segmentation for accurate freespace detection. In: Computer vision-ECCV (2020) 16th European conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16. Springer, pp 340–356
13. Shean DE et al (2016) An automated, open-source pipeline for mass production of digital elevation models (DEMs) from very-high-resolution commercial stereo satellite imagery. ISPRS J Photogramm Remote Sens 116:101–117
14. Menze M et al (2015) Object scene flow for autonomous vehicles. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3061–3070

15. Scharstein D et al (2014) High-resolution stereo datasets with subpixel-accurate ground truth. In: German conference on pattern recognition (GVPR). Springer, pp 31–42
16. Schops T et al (2017) A multi-view stereo benchmark with high-resolution images and multi-camera videos. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 3260–3269
17. Mayer N et al (2016) A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 4040–4048
18. Hamid MS et al (2022) Stereo matching algorithm based on deep learning: a survey. J King Saud Univ-Comput Inf Sci 34(5):1663–1673
19. Bendig K et al (2022) Self-superflow: self-supervised scene flow prediction in stereo sequences. In: Proceedings of the IEEE international conference on image processing (ICIP). IEEE, pp 481–485
20. Lee M-J et al (2022) Refinement of inverse depth plane in textureless and occluded regions in a multiview stereo matching scheme. J Sens 2022
21. Gidaris S et al (2018) Unsupervised representation learning by predicting image rotations. arXiv:1803.07728
22. Trucco E et al (1998) Introductory techniques for 3-D computer vision, vol 201 . Prentice Hall Englewood Cliffs
23. Fan R et al (2023) Computer stereo vision for autonomous driving: theory and algorithms. In: Recent advances in computer vision applications using parallel processing. Springer, pp 41–70
24. Loop C et al (1999) Computing rectifying homographies for stereo vision. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition (CVPR), vol 1. IEEE, pp 125–131
25. Tippetts B et al (2016) Review of stereo vision algorithms and their suitability for resource-limited systems. J Real-Time Image Proc 11(1):5–25
26. Ding J et al (2021) High-accuracy recognition and localization of moving targets in an indoor environment using binocular stereo vision. ISPRS Int J Geo Inf 10(4):234
27. Fan R et al (2018) Road surface 3D reconstruction based on dense subpixel disparity map estimation. IEEE Trans Image Process 27(6):3025–3035
28. Luo W et al (2016) Efficient deep learning for stereo matching. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 5695–5703
29. Hamzah RA et al (2016) Literature survey on stereo vision disparity map algorithms. J Sens 2016
30. Yamaguchi K et al (2012) Continuous markov random fields for robust stereo estimation. In: European conference on computer vision (ECCV). Springer, pp 45–58
31. Boykov Y et al (2001) Fast approximate energy minimization via graph cuts. IEEE Trans Pattern Anal Mach Intell 23(11):1222–1239
32. Brown MZ et al (2003) Advances in computational stereo. IEEE Trans Pattern Anal Mach Intell 25(8):993–1008
33. Hirschmuller H (2007) Stereo processing by semiglobal matching and mutual information. IEEE Trans Pattern Anal Mach Intell 30(2):328–341
34. Seki A et al (2017) SGM-Nets: semi-global matching with neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 231–240
35. Spangenberg R et al (2013) Weighted semi-global matching and center-symmetric census transform for robust driver assistance. In: International conference on computer analysis of images and patterns (CAIP). Springer, pp 34–41
36. Žbontar J et al (2016) Stereo matching by training a convolutional neural network to compare image patches. J Mach Learn Res 17(1):2287–2318
37. Hirschmuller H et al (2008) Evaluation of stereo matching costs on images with radiometric differences. IEEE Trans Pattern Anal Mach Intell 31(9):1582–1599
38. Scharstein D et al (2003) High-accuracy stereo depth maps using structured light. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition (CVPR), vol 1. IEEE, pp 195–202

39. Yang Q et al (2008) Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling. IEEE Trans Pattern Anal Mach Intell 31(3):492–504
40. Fan R et al (2019) Real-time dense stereo embedded in a UAV for road inspection. In: IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW), pp 535–543
41. Mattoccia S (2011) Stereo vision: algorithms and applications, vol 22. University of Bologna
42. Scharstein D et al (2002) A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int J Comput Vision 47(1):7–42
43. Zbontar J et al (2015) Computing the stereo matching cost with a convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1592–1599
44. Chang J-R et al (2018) Pyramid stereo matching network. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 5410–5418
45. Wang H et al (2021) PVStereo: pyramid voting module for end-to-end self-supervised stereo matching. IEEE Robot Autom Lett 6(3):4353–4360
46. Zhong Y et al (2017) Self-supervised learning for stereo matching with self-improving ability. arXiv:1709.00930
47. Wang H et al (2021) Co-teaching: an ark to unsupervised stereo matching. In: Proceedings of the IEEE international conference on image processing (ICIP). IEEE, pp 3328–3332
48. Zhang F et al (2019) GA-Net: guided aggregation net for end-to-end stereo matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 185–194
49. Kendall A et al (2017) End-to-end learning of geometry and context for deep stereo regression. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 66–75
50. Guo X et al (2019) Group-wise correlation stereo network. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 3273–3282
51. Cheng X et al (2020) Hierarchical neural architecture search for deep stereo matching. Adv Neural Inf Process Syst 33:22 158–22 169
52. Xu H et al (2020) AANet: adaptive aggregation network for efficient stereo matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 1959–1968
53. Teed Z et al (2020) RAFT: recurrent all-pairs field transforms for optical flow. In: European conference on computer vision (ECCV). Springer, pp 402–419
54. Lipson L et al (2021) RAFT-stereo: multilevel recurrent field transforms for stereo matching. In: Proceedings of the IEEE international conference on 3D vision (3DV). IEEE, pp 218–227
55. Li J et al (2022) Practical stereo matching via cascaded recurrent network with adaptive correlation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 16 263–16 272
56. Zhu X et al (2019) Deformable ConvNets V2: more deformable, better results. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 9308–9316
57. Wang Y et al (2019) Anytime stereo image depth estimation on mobile devices. In: Proceedings of the IEEE international conference on robotics and automation (ICRA). IEEE, pp 5893–5900
58. Yee K et al (2020) Fast deep stereo with 2D convolutional processing of cost signatures. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV), pp 183–191
59. Tankovich V et al (2021) HitNet: hierarchical iterative tile refinement network for real-time stereo matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 14 362–14 372
60. Chen S et al (2023) Feature enhancement network for stereo matching. Image Vis Comput 131:104614
61. Wang Z et al (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612
62. Zhou C et al (2017) Unsupervised learning of stereo matching. In: Proceedings of the IEEE international conference on computer vision (ICCV), pp 1567–1575

63. Li A et al (2018) Occlusion aware stereo matching via cooperative unsupervised learning. In: Asian conference on computer vision (ACCV). Springer, pp 197–213
64. Joung S et al (2019) Unsupervised stereo matching using confidential correspondence consistency. IEEE Trans Intell Transp Syst 21(5):2190–2203
65. Liu P et al (2020) Flow2Stereo: effective self-supervised learning of optical flow and stereo matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 6648–6657
66. Liu L et al (2020) Learning by analogy: reliable supervision from transformations for unsupervised optical flow estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 6489–6498
67. Khamis S et al (2018) StereoNet: guided hierarchical refinement for real-time edge-aware depth prediction. In: European conference on computer vision (ECCV). Springer, pp 573–590
68. Fan X et al (2022) Occlusion-aware self-supervised stereo matching with confidence guided raw disparity fusion. In: Proceedings of the conference on robots and vision (CRV). IEEE, pp 132–139
69. Poggi M et al (2021) On the confidence of stereo matching in a deep-learning era: a quantitative evaluation. arXiv:2101.00431
70. Egnal G et al (2004) A stereo confidence metric using single view imagery with comparison to five alternative approaches. Image Vis Comput 22(12):943–957
71. Haeusler R et al (2013) Ensemble learning for confidence measures in stereo vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 305–312
72. Matthies L (1992) Stereo vision for planetary rovers: stochastic modeling to near real-time implementation. Int J Comput Vision 8(1):71–91
73. Spyropoulos A et al (2014) Learning to detect ground control points for improving the accuracy of stereo matching. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 1621–1628
74. Ding L et al (2001) On the canny edge detector. Pattern Recogn 34(3):721–725
75. Egnal G et al (2002) Detecting binocular half-occlusions: empirical comparisons of five approaches. IEEE Trans Pattern Anal Mach Intell 24(8):1127–1133
76. Hu X et al (2012) A quantitative evaluation of confidence measures for stereo vision. IEEE Trans Pattern Anal Mach Intell 34(11):2121–2133
77. Barron JL et al (1994) Performance of optical flow techniques. Int J Comput Vision 12:43–77
78. Szeliski R (1999) Prediction error as a quality metric for motion and stereo. In: Proceedings of the IEEE international conference on computer vision (ICCV), vol 2. IEEE, pp 781–788
79. Baker S et al (2011) A database and evaluation methodology for optical flow. Int J Comput Vis 92:1–31
80. Kalarot R et al (2011) Analysis of real-time stereo vision algorithms on gpu. In: International conference on image and vision computing New Zealand (IVCNZ), p 1
81. Geiger A et al (2012) Are we ready for autonomous driving? The kitti vision benchmark suite. In: Conference on computer vision and pattern recognition (CVPR)
82. Butler DJ et al (2012) A naturalistic open source movie for optical flow evaluation. In: European conference on computer vision (ECCV). Springer, pp 611–625
83. Silberman N et al (2012) Indoor segmentation and support inference from rgbd images. In: European conference on computer vision (ECCV). Springer, pp 746–760
84. Gaidon A et al (2016) Virtual worlds as proxy for multi-object tracking analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), pp 4340–4349
85. Cabon Y et al (2020) Virtual KITTI 2. arXiv:2001.10773
86. Tremblay J et al (2018) Falling things: a synthetic dataset for 3d object detection and pose estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops (CVPRW), pp 2038–2041
87. Yang G et al (2019) DrivingStereo: a large-scale dataset for stereo matching in autonomous driving scenarios. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 899–908

88. Hirschmuller H et al (2007) Evaluation of cost functions for stereo matching. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 1–8
89. Song X et al (2021) AdaStereo: a simple and efficient approach for adaptive stereo matching. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 10 328–10 337
90. Jingwei Y et al (2023) Semantic segmentation for autonomous driving. Springer submitted for publication
91. Wang H et al (2021) SCV-Stereo: learning stereo matching from a sparse cost volume. In: Proceedings of the IEEE international conference on image processing (ICIP). IEEE, pp 3203–3207
92. Dosovitskiy A et al (2015) FlowNet: learning optical flow with convolutional networks. In: Proceedings of the IEEE international conference on computer vision (CVPR), pp 2758–2766
93. Kuznietsov Y et al (2017) Semi-supervised deep learning for monocular depth map prediction. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6647–6655
94. Zhang F et al (2020) Domain-invariant stereo matching networks. In: European conference on computer vision (ECCV) 2020. Springer, pp 420–439
95. Yang G et al (2019) Hierarchical deep stereo matching on high-resolution images. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 5515–5524
96. Zhang J et al (2022) Revisiting domain generalized stereo matching networks from a feature consistency perspective. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 13 001–13 011
97. Watson J et al (2021) The temporal opportunist: self-supervised multi-frame monocular depth. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 1164–1174
98. Shu C et al (2020) Feature-metric loss for self-supervised learning of depth and egomotion. In: European conference on computer vision (ECCV). Springer, pp 572–588
99. Godard C et al (2019) Digging into self-supervised monocular depth estimation. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV), pp 3828–3838
100. Kim S et al (2020) Adversarial confidence estimation networks for robust stereo matching. IEEE Trans Intell Transp Syst 22(11):6875–6889
101. Wu C-Y et al (2022) Toward practical monocular indoor depth estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 3814–3824
102. Kanopoulos N et al (1988) Design of an image edge detection filter using the sobel operator. IEEE J Solid-State Circuits 23(2):358–367
103. Liu H et al (2021) Pseudo supervised monocular depth estimation with teacher-student network. arXiv:2110.11545
104. Hartley R et al (2003) Multiple view geometry in computer vision. Cambridge University Press