

# Scene Flow Estimation: A Survey

Zike Yan<sup>1,a</sup>, Xuezhi Xiang<sup>2,a,\*</sup>

<sup>a</sup>College of Information and Communication Engineering,  
Harbin Engineering University, Harbin 150001, China

May 23, 2017

## Abstract

This paper is the first to review the scene flow estimation, which analyzes and compares methods, technical challenges, evaluation methodologies and performance of scene flow estimation. Existing algorithms are categorized in terms of scene representation, data source, and calculation scheme, and the pros and cons in each category are compared briefly. The datasets and evaluation protocols are enumerated, and the performance of the most representative methods is presented. A future vision is illustrated with few questions arisen for discussion. This survey presents a general introduction and analysis of scene flow estimation.

## 1 Introduction

Scene flow is a three-dimensional motion field of the surface in world space, or in other words, it shows the three-dimensional displacement vector of each surface point between two frames. As most computer vision issues are, scene flow estimation is essentially an ill-posed energy minimization problem with three unknowns. Prior knowledge in multiple aspects is required to make the energy function solvable with just a few pairs of images. Hence, it's essential to fully make use of information from the data source and to weigh different prior knowledge for a better performance.

---

\* Corresponding author

Email addresses: yanlike@hrbeu.edu.cn (Zike Yan<sup>1</sup>), xiangxuezhi@hrbeu.edu.cn (Xuezhi Xiang<sup>2</sup>)

The paper attempts to reveal clues by providing a comprehensive literature survey in this field. Scene flow is first introduced by Vedula in 1999 [1] and has made constant progress over the years. Diverse data sources has emerged thus scene flow estimation don't need to set up the complicated array of cameras. The conventional framework derived from optical flow field [2, 3] has extended to this three-dimensional motion field estimation task, while diverse ideas and optimization manners has improved the performance noticeably. The widely concerned learning based method has been utilized for scene flow estimation [4], which brings fresh blood to this integrated field. Moreover, a few methods have achieved real-time estimation with GPU implementation at the QVGA( $320 \times 240$ ) resolution [5, 6, 7, 8], which insure a promising efficiency. The emergence of these methods stands for the fact that scene flow estimation will be widely utilized and applied to practice soon in the near future.

The paper is organized as follows. Section 2 illustrates the relevant issues, challenges and applications of scene flow as a background . Section 3 provides classification of scene flow in terms of three major components. Emerging datasets that are publicly available and the diverse evaluation protocols are presented and analyzed in Section 4. Section 5 arises few questions to briefly discuss the content mentioned above, and the future vision is provided. Finally, a conclusion is presented in Section 6.

## 2 Background

We provide relevant issues, major challenges and applications as the background information for better understanding this field.

### 2.1 Relevant issues

Scene flow estimation is an integrated task, which is relevant to multiple issues. Firstly, optical flow is the projection of scene flow onto an image plane, which is the basis of scene flow and has made steady progress over the years. The basic framework and innovations of scene flow estimation mainly derives from optical flow estimation field. Secondly, in a binocular setting, scene flow can be simply acquired by coupling stereo and optical flow, which makes the stereo matching an essential

part for scene flow estimation. Most scene flow estimation methods with promising performance are initialized with a robust optical flow method or a stereo matching method. And the innovation in scene flow mostly derived from these two fields. Hence, we provides the changes and trend in the relevant issues as heuristic information.

### 2.1.1 Optical flow

Optical flow is a two-dimensional motion field. The global variational Horn-Schunck(H-S) method and the local total-least-square(TLS) Lucas-Kanade(L-K) method have led the optical flow field and scene flow field over the years [2, 3]. Early works was studied and categorized by Barron and Otte with quantitative evaluation models [9, 10]. Afterwards, Brox implemented the coarse-to-fine strategy to deal with large displacement [11], while Sun studied the statistics of optical flow methods to find the best way for modeling [12]. Baker proposed a thorough taxonomy of current optical flow methods and introduced the Middlebury dataset for evaluation [13], and comparisons between error evaluation methodologies, statistics and datasets are presented as well.

Currently, optical flow estimation has reached to a promising status. A segmentation-based method with the approximate nearest neighbor field to handle large displacement ranks the top of Middlebury dataset in terms of both endpoint error(EPE) and average angular error(AAE) currently [14], where EPE varies from  $0.07px$  to  $0.41px$  in different data and AAE varies from  $0.99^\circ$  to  $2.39^\circ$ . A similar method reached promising results as well [15]. Moreover, there are a variety of methods which achieve top-tier performance and solve different problems respectively. Rushwan utilized a tensor voting method to preserve discontinuity [16]. Xu introduced a novel extended coarse-to-fine optimization framework for large displacement [17], while Stoll combines the feature matching method with variational estimation to keep small displacement area from being compromised [18]. He also introduced a multi-frame method utilizing trilateral filter [19]. To handle non-rigid optical flow, Li proposed a Laplacian mesh energy formula which combines both Laplacian deformation and mesh deformation [20].

### **2.1.2 Stereo matching**

Stereo matching is essential to scene flow estimation under binocular setting. A stereo algorithm generally consists of four parts:(1) matching cost computation, (2) cost aggregation, (3) estimation and optimization and (4) refinement. It is categorized into local methods and global methods depending on how the cost aggregation and computation are performed. Local methods suffer from the textureless region, while global methods are computationally expensive. A semi-global-matching(SGM) method combines local smoothness and global pixel-wise estimation and leads to a dense matching result at low runtime [21], which is commonly utilized as the modification. A comprehensive review is presented by Scharstein in 2001 [22].

The upper rank algorithms of Middlebury stereo dataset and KITTI stereo dataset [22, 23] are mainly occupied by unpublished papers, indicating the rapid development in this field. Learning methods are utilized with promising efficiency and accuracy [24, 25]. Besides, Zhang proposed a mesh-based approach considering the high speed of rendering and ranks the top among the published papers [26], while segmentation-based methods are proven to tackle the textureless problem [17, 27].

## **2.2 Challenges**

The complex scene and the limited image capture approach post challenges in diverse ways, which are discussed as follows.

### **2.2.1 Accuracy and efficiency**

To achieve better accuracy, sufficient and complicated prior knowledge is obliged, while in terms of efficiency, the data need to be listed down to a reasonable scale and the calculation scheme should be as simple as possible. We cannot only consider the enhancement of efficiency and accuracy, but also value the trade-off between these two. The trade-off is discussed in Section 5, and the performance is illustrated in Figure 11.

### 2.2.2 Occlusion

Occlusion is common in a complex scene with multiple moving objects. It occurs between views and frames as Figure 1 illustrates. It violates the data consistency assumption and may lead to mismatching on account of missing information of the occluded object. Besides, occlusion may perturb the consistency between frames and affect the multi-frames tracking method.

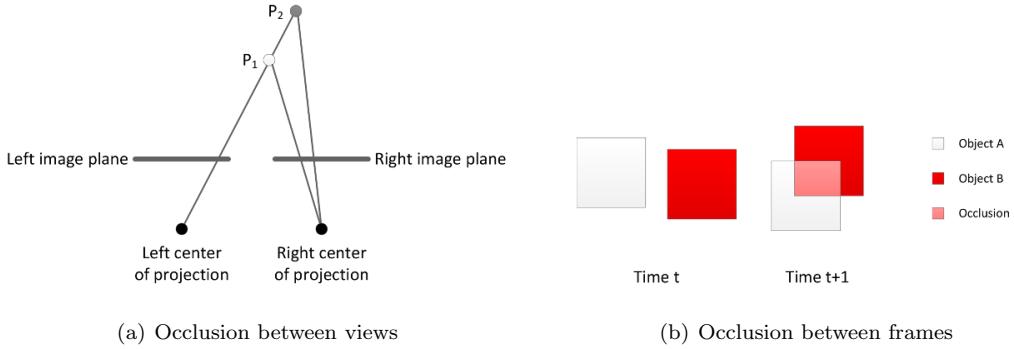


Figure 1: Occlusion conditions

The occlusion between views can be handled well under the multi-view stereopsis with abundant prior knowledge, while temporal constraint may provide robust temporal coherence and prediction to alleviate occlusion between frames.

### 2.2.3 Large displacement

Large displacement occurs frequently when an object is moving at a high speed or under a limited frame-rate. Moreover, articulated motion may lead to large displacement as well. This kind of problem is hard to tackle on account that the scene flow algorithms normally assume the constancy and smoothness within a small region, large displacement may make the solution to energy function trapped into a local minimum which leads to enormous errors propagated by iteration procedure.

Brox implemented the coarse-to-fine method along with a gradient constancy assumption to alleviate the impact caused by large displacement in the optical flow field [11]. Currently, several matching algorithms have been introduced to handle this issue specifically and achieved promising results [28, 29].

#### 2.2.4 Varying illumination

Brightness constancy doesn't obey the illumination-varying environment. However, this issue is common in an outdoor scene, e.g., drifting clouds that block the sunlight, sudden reflection from a window, and lens flares. The night scenario may make it worse when lights start to flash. In the optical flow field, additional assumptions such as gradient constancy and some more complicated constraints have been added to make it more robust to the illumination changes [13]. Schuchert specifically studied range flow estimation under varying illumination [30]. In his paper, pre-filtering and changes of brightness model improve the accuracy. Afterwards, Gotardo introduced an albedo consistency assumption for further study [31]. A relighting procedure was proposed as a key element to handle the multiplexed situation in his paper as well.

#### 2.2.5 Insufficient texture

The lack of texture may make the scene flow estimation still an ill-posed problem, which is a challenge for discovering consistency. It is also a challenge to stereo matching, which may lead to enormous errors in the binocular-based scene flow estimation. The textureless region is still a major contribution to the estimation error.

To overcome this problem, different scene representations have been utilized. For example, Popham introduced a patch-based method [32]. The motion of each patch doesn't only rely on the texture information, but utilizes the motion from neighbor patches. This makes it more robust for a textureless region. As a solution to the occlusion issue, segmentation-based method is valid because it assumes uniform motion among the small regions to deal with the ambiguousness [33, 34].

### 2.3 Applications

Scene flow estimation is a comprehensive problem. Motion information reveals the temporal coherence between two moments. In a long sequence, scene flow can be utilized to get the initial value for the next frame and serve as a constraint in its relevant issue fields. Scene flow can not only profit from its relevant issues, but also facilitate them mutually. Gotardo captured three-dimensional scene flow to provide delicate geometric details [31], while Liu utilized scene flow as a soft constraint for

stereo matching and a prediction for next frame disparity estimation [34]. Ghuffar combined local estimation and global regularization in a TLS framework and utilized scene flow for segmentation and trajectory generation [35].

Beyond that, scene flow can be a valuable input or mobile robotics and autonomous driving field, which consist of multiple task such as obstacle avoidance and scene understanding. Frank first fused optical flow and stereo by means of Kalman filter for obstacle avoidance [36]. Alcantarilla combined scene flow estimation with the visual SLAM to enhance the robustness and accuracy [37]. Herbst got object segmentation with the RGB-D scene flow estimation result [38], aiming to achieve autonomous exploration of indoor scenes. Menze utilized scene flow to reason objects by regarding the scene as a set of rigid objects [39]. Autonomous driving could make use of both the geometric information that represents distance and the scene flow information that represents motion for multiple tasks.

In addition, scene flow can be utilized to serve as a feature as the histogram of optical flow(HOF) [40] or the motion boundary histogram(MBH) [41] feature descriptors for object detection and recognition, e.g., facial expression, gesture, and body motion recognition. It may enrich the information in the descriptor with additional depth dimension and can be applied for motion like rotation or dolly moves that optical flow can't handle. For instance, in 2009, Furukawa recorded the motion model of the facial expression using scene flow estimation[42].

### 3 A taxonomy of scene flow estimation methods

Scene flow estimation is viewed as an ill-posed problem, which includes three main steps: data acquisition, energy function modeling, energy minimization and optimization. The general taxonomy is depicted in Figure 2.

The energy function consists of data term and regularization as Equation 1 illustrates.

$$E(\mathbf{V}) = E_D(\mathbf{V}) + \alpha E_R(\mathbf{V}) \quad (1)$$

Data terms derive from different data sources assuming brightness constancy(BC) or gradient constancy(GC) as local constraint. While there are three unknown parameters, regularization terms need to be added to regularize the ill-posed problem and provide spatial coherence. Multiple regularization terms may make the However, miscellaneous regularization terms may lead to redundancy,

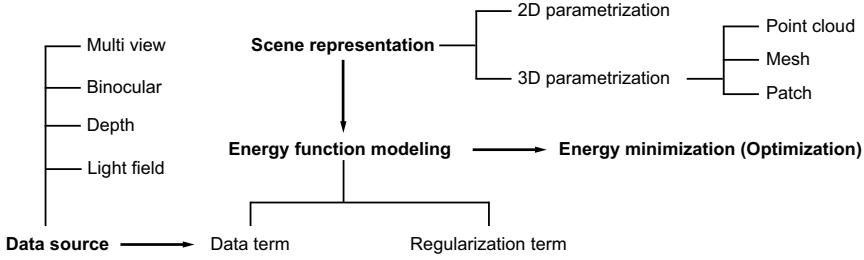


Figure 2: A general taxonomy of scene flow estimation

intractability and over-fitting, and that's why the design and solution of regularization term are key to a method.

Hence, in this section, existing methods are categorized in terms of three fundamental properties that distinguish the major algorithms: *Scene representation* presented the diverse representations for both scene and scene flow. *Data source* describes the major data acquisition manner and the corresponding data term choices. *Calculation scheme* mainly discuss the idea for estimation and optimization manner, including diverse choices of regularization terms and implement .

### 3.1 Scene representation

Over the years, diverse ways to represent the scene have emerged with different emphasis, which can be broadly categorized into *depth/disparity*, *point cloud*, *mesh* and *patch*.

#### 3.1.1 Depth/disparity

The convenient way to represent the scene is to couple color image and depth map as color-D information, where D stands for depth information with RGB-D data or disparity information under a binocular setting. Scene flow under this sort of representation is known as 2.5D scene flow or 2D parameterization of scene flow. It consists of optical flow component which is measured in pixels, and disparity or depth change component which is measured in pixels or  $m$ . The binocular-based scene flow [43, 44, 45, 5, 6, 46, 47, 48] can be presented as  $\mathbf{v} = (u, v, \delta d)$ , where  $(u, v)$  is the 2D optical flow, and the  $\delta d$  stands for disparity change. Likewise, in terms of RGB-D scene flow [38, 49, 8], the motion field consists can be presented  $\mathbf{v} = (u, v, \Delta Z) = (u, v, W)$ , where  $W$  stands for the depth

change.

Particularly, disparity value can be converted into depth value as Equation 2 illustrates.

$$Z = \frac{f * b}{d} \quad (2)$$

where  $f$  is the focal length of the camera, and  $b$  is the camera baseline value.

### 3.1.2 Point cloud

To truly present the three-dimensional scene, the image pixel need to be projected into the scene space. The projection is illustrated in Figure 3(a) and presented in Equation 3.

$$\pi^{-1}(\mathbf{x}, Z) = (Z \frac{x - c_x}{f_x}, Z \frac{y - c_y}{f_y}, Z)^T \quad (3)$$

where  $\mathbf{x}$  is the image pixel  $\mathbf{x} = (x, y)$ ,  $\pi^{-1}(\mathbf{x}, Z)$  is the projection  $\mathbb{R}^2 \times \mathbb{R} \rightarrow \mathbb{R}^3$  from a pixel value  $\mathbf{x}$  and a depth value  $z$  to a 3D point  $\mathbf{X} = (X, Y, Z)$ .  $f_x$  and  $f_y$  stands for the camera focal length, and  $c_x$  and  $c_y$  are the principal points.

Formulation 3 can also be presented as:

$$Z \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = A\mathbf{X} \quad (4)$$

where matrix  $A$  is known as the camera projection matrix.

Hence, scene flow under point cloud representation [1, 50, 51, 33, 52, 53, 54, 55, 56, 57, 58, 35, 59, 60] can be presented as  $\mathbf{V} = (\Delta X, \Delta Y, \Delta Z) = (U, V, W)$ , which truly reveal the three-dimensional displacement.

### 3.1.3 Mesh

Meshes represent a surface as a set of planar polygons, e.g., triangle, which connected to each other as is shown in Figure 3(b). This representation is a efficient way for rendering, and it occupies less memory. Mesh is essentially sort of point cloud representation as the vertex can be viewed as a point in the three-dimensional point cloud. Scene flow estimation methods with a mesh representation [61, 62, 63, 42, 64, 65, 66, 67, 32] are only under a multi-view setting and the geometry estimation is given simultaneously. The motion of vertice is solved with a point cloud methods, while the rest part are solved by interpolating along the meshes.

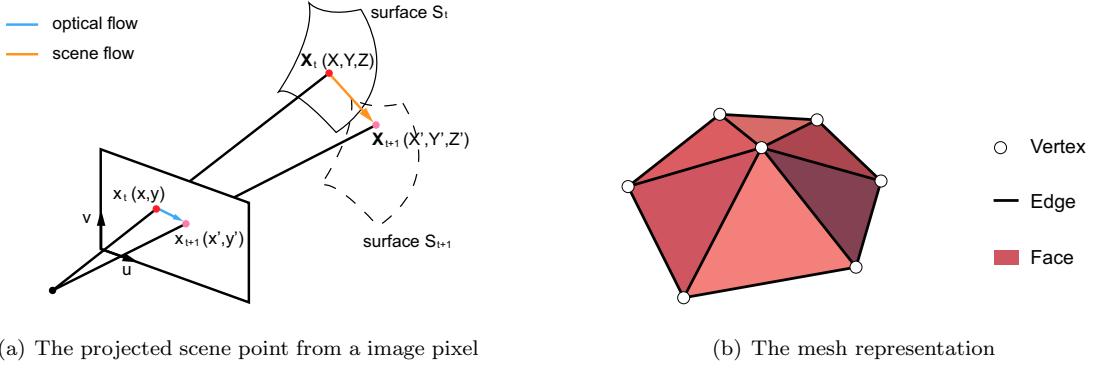


Figure 3: Brief sketches about point cloud representation and mesh representation

### 3.1.4 Patch

In terms of patch representation, the surface is represented by collections of small planar or sphere patches. Each patch is six-dimensional in terms of three-dimensional patch center position and three-dimensional patch direction. Patch representation is similar to mesh representation with different emphasis, where mesh representation focuses on the precision and deformable property of each vertex, and patch representation values the local consistency in terms of rigidity and motion within a small neighborhood region.

Early paper viewed patch as a surface element(surfel) under a multi-view setting[68, 69]. A few binocular-based scene flow methods utilized patches to fit the surface of the scene [43, 70, 71, 32, 72, 73, 31] on account that this kind of patch-based methods are common in stereo matching field. In addition, Hornacek uniquely exploited a pair of RGB-D data to seek patch correspondences in the 3D world space and leads to dense body motion field including both translation and rotation [74].

## 3.2 Data source

Over the years, scene flow has been estimated under three main kinds of data source: a calibrated multi-camera system, the binocular stereo camera or the RGB-D camera which consists of both RGB color information and depth information. Moreover, the emerging light field has been applied into the scene flow estimation with promising performance [48]. Data terms under different data sources differs from each other. Hence, in this section, scene flow estimation methods are categorized into

these four kinds: *multi-view stereopsis*, *binocular setting*, *RGB-D data* and *light field data*.

### 3.2.1 Multi view stereopsis

Most of the algorithms in the early 2000s assume a multi-view system, with multiple cameras set in a complex calibrated scene. Multi-view scene flow estimation is usually along with 3D geometry reconstruction simultaneously. Ample data sources and diverse prior knowledge ensure the robustness of the estimation, and occlusion issue can be handled well. However, it is commonly at a high computational cost with an intricate full-view scene to deal with.

Vedula proposed two choices for regularization and distinguished three scenarios in 1999 [1], which guides the multi-view scene flow estimation till now. In his paper, a multi-view scene flow can be estimated from one optical flow and the known surface geometry. The equation is formulated in Equation 5.

$$\mathbf{V}(\mathbf{X}) = \frac{\partial \mathbf{X}}{\partial \mathbf{x}} \mathbf{v}(\mathbf{x}) \quad (5)$$

where  $\mathbf{X} = (X, Y, Z)$  is the three-dimensional scene point,  $\mathbf{x} = (x, y)$  is the two-dimensional image pixel,  $\mathbf{V}(\mathbf{X})$  is the scene flow,  $\mathbf{v}(\mathbf{x})$  is the optical flow, and  $\frac{\partial \mathbf{X}}{\partial \mathbf{x}}$  is the inverse Jacobian which can be estimated from the surface gradient  $\nabla S(\mathbf{X})$ .

Afterwards, Zhang proposed two systems for estimation [50, 51, 33], where IMS assumed each small patch undergoes 3D affine motion, and EGS used segmentation to keep the boundary. These papers modeled energy function with multiple constraint, which provided a basic estimation process. Similarly, Pons presented a common variational framework with local similarity criteria constraint [75, 76]. Henceforth, different scene representations were introduced to describe the surface [68, 61, 62, 63, 42, 70, 65, 32]. Diverse multi-frame tracking methods mentioned for sparse estimation are utilized as well to build the temporal coherence [36, 69, 64, 67]. Moreover, Letouzey added an RGB-D camera into the multi-view system with a mesh representation [66], aiming to enrich the geometry information with the depth data constraint.

### 3.2.2 Binocular setting

Binocular setting is regarded as a basic and simplified version of multi-view system, while the difference between the two is that binocular scene flow estimation is usually along with disparity estimation

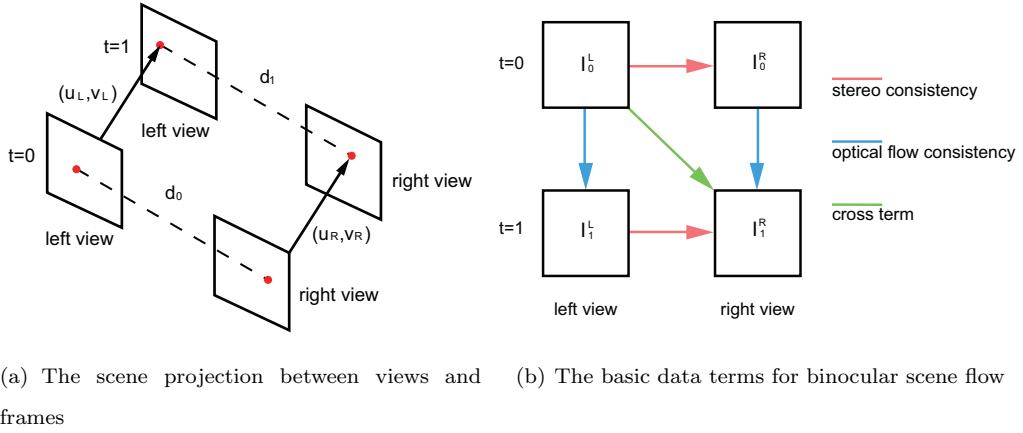


Figure 4: Brief sketches about scene flow under binocular settings

between two views but not the full-view 3D geometry knowledge. The relevance between views and frames is illustrated in Figure 4(a), and Figure 4(b) depicts the basic data terms in a binocular setting which consists of stereo consistency terms in time  $t$  and  $t + 1$  along with optical flow consistency terms in both views. The specific formulation is presented in Equation 7.

$$E_D = E_{fl} + \alpha E_{fr} + \beta E_{d0} + \gamma E_{dt} + \theta E_{cr} \quad (6)$$

where

$$E_{fl} = \sum_{\Omega} \Psi(I(x+u, y+v) - I(x, y)) \quad E_{fr} = \sum_{\Omega} \Psi(I(x+u+d_{t+1}, y+v) - I(x+d_t, y)) \quad (7a)$$

$$E_{d0} = \sum_{\Omega} \Psi(I(x+d_t, y) - I(x+d_t, y)) \quad E_{dt} = \sum_{\Omega} \Psi(I(x+u+d_{t+1}, y+v) - I(x+u, y+v)) \quad (7b)$$

$$E_{cr} = \sum_{\Omega} \Psi(I(x+u+d_{t+1}, y+v) - I(x, y)) \quad (7c)$$

$E_{fl}$  and  $E_{fr}$  are the optical flow consistency terms that assume the brightness of the same pixel stay constant between frames. Similarly,  $E_{d0}$  and  $E_{dt}$  are the stereo consistency terms that assume brightness constancy between views. Moreover,  $E_{cr}$  is the cross term to constrain the constancy between both frames and views.

Most binocular-based methods fused stereo and optical flow estimation into a joint framework [52, 43, 44, 6, 46, 77, 47, 27]. On the contrary, others decoupled motion from disparity estimation to estimate scene flow with stereo matching method replaceable at will [45, 7, 31, 4], and Basha utilized a point cloud scene representation as a three-dimensional parametrization version of scene

flow [54]. Moreover, local rigidity prior was presented along with segmentation prior and achieved promising results [78, 71, 72, 73]. Specifically, Valgaerts introduced a variational framework for scene flow estimation under an uncalibrated stereo setup by embedding an epipolar constraint [79], which makes it possible for scene flow estimation under two arbitrary cameras. In 2016, Richardt has made it a reality to compute dense scene flow from two handheld cameras with varying camera settings [60]. Scene flow was estimated under a variational framework with a DAISY descriptor [80] for wide-baseline matching.

Table 1 enumerates some typical methods under a binocular setting with diverse choices of data terms. Most methods chose optical flow consistency terms in both views and stereo consistency terms in both time  $t$  and time  $t + 1$  [52, 54, 78, 71], and few methods only take parts of terms mentioned above [43, 45, 46]. Cross term was utilized in [46, 78, 47]. Moreover, Huguet [44] and Hung [47] utilized additional gradient constancy assumption besides intensity to enhance robustness against illumination changes, which turns the image intensity value  $I(x, y)$  in energy function into image gradient  $G(x, y)$ . Additionally, extra RGB constancy terms  $\frac{1}{4}(I_G - I_R)$  and  $\frac{1}{4}(I_G - I_B)$  are taken in Hung’s paper as well, which extends gray value intensity into three-channel information. However, it is proposed that image gradient is sensitive to noise and is view dependent [54, 78]. Hence, the necessity of additional assumptions like gradient constancy remains further research to balance the pros and cons.

Literature	Year	Data term	Description
Li[52]	2005	$E_{fl}, E_{fr}, E_{d0}, E_{dt}$	First method under a binocular setting in the early stage.
Isard[43]	2006	$E_{fr}, E_{dt}, E_{cr}$	First utilizing cross term. Under a Markov random field(MRF) framework.
Huguet[44]	2007	$E_{fl}, E_{fr}, E_{d0}, E_{dt}$	Introducing a basic framework for scene flow under the binocular setting.
Wedel[45]	2008	$E_{fl}, E_{fr}, E_{dt}$	Decoupling motion and stereo.
Basha[54]	2010	$E_{fl}, E_{fr}, E_{d0}, E_{dt}$	First utilizing point cloud representation.
Cech[46]	2011	$E_{fl}, E_{fr}, E_{d0}, E_{dt}, E_{cr}$	Utilizing a seeded growing-propagation framework for fast implementation.
Hung[47]	2013	$E_{fl}, E_{fr}, E_{d0}, E_{dt}, E_{cr}$	Assuming additional RGB intensity and gradient constancy.

Table 1: Typical methods under the binocular setting

### 3.2.3 RGB-D data

Depth was regarded as a function of space and time by Spies [81, 82]. He added range flow motion constraint and introduced the range flow motion field. On basis of Spies' theory, Luckins added color channel constraint as an additional information to enhance the robustness [83]. Moreover, Schuchert added gradient constancy assumption and used pre-filtering to handle varying illumination [30].

With the development of RGB-D camera, depth can be acquired easily and accurately. RGB-D information can be seen as a cheap data source for geometry knowledge, and the depth information from RGB-D camera were seen as a cheap and efficient source for layering [49], which provided layer ordering straightly without exhaustive search. However, the application of RGB-D scene flow estimation will be restrained on account of the limited sensing range and the unstable performance under illumination or reflection, which is struggling in an outdoor scene. A comparison between the state-of-art consumer RGB-D cameras is presented in Table 2 to show the limitation of current RGB-D sensors in terms of range, frame rate and angle of depth measurement. Moreover, the quality of depth map is far from satisfactory due to the invalid data around object boundary region, noise and error pixels as Figure 5 presents.

Sensor	Res	Dis	FPS	FOV(H,V)
SR4000	176×144	5m/10m	50	(43°, 34°)
Kinect v1	640×480	4.5m	30	(57°, 43°)
Kinect v2	1920×1080	4.5m	30	(70°, 60°)
Camcube	204×204	7.5m	40	(40°, 40°)

Table 2: Performance of the state-of-the-art RGB-D cameras. Notation for the header: **Res**: the maximum resolution, **Dis**: the maximum sensing range of the sensor, **FPS**: the abbreviation of frame per second, **FOV(H,V)**: the field of view in both horizontal and vertical side.

Gottfried was the first to use Kinect sensor for scene flow estimation [84]. He addressed all essential stages including calibration, alignment and estimation. Afterwards, following the common optical flow optimization, scene flow was solved under the variational framework [38, 85], and was then modified by combining additional local rigidity priors [56, 59, 35]. Similarly, the pixel assignment methods are becoming more and more popular [57, 58, 86, 28]. In this way can the discontinuity be

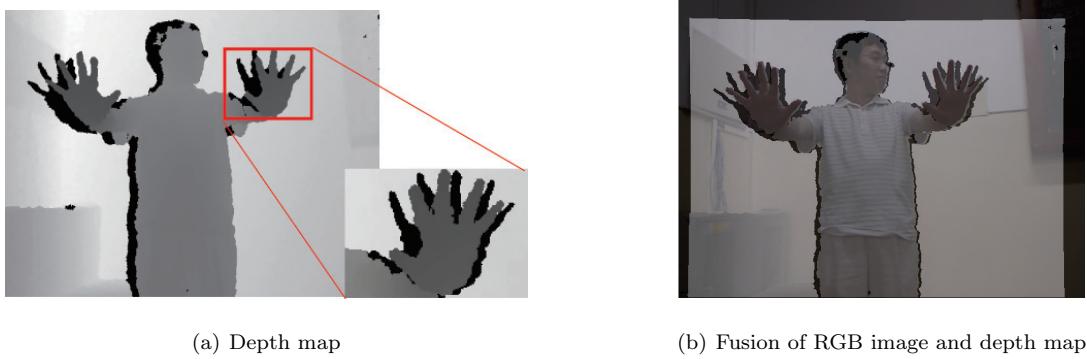


Figure 5: RGB-D data acquired by Kinect sensor. It is clear that the boundary of object is occupied with many missing data colored in black due to the reflection.

preserved smoothly. In addition, scene particle method [55, 87] and feature matching method [74, 29] are applied to scene flow estimation with pros and cons as supplements besides the common variational methods.

The basic data terms for RGB-D scene flow estimation consist of brightness constancy term(BC) and depth change consistency term(DCC):

$$E_{BC} = \sum_{\Omega} \Psi(I(x+u, y+v) - I(x, y)) \quad E_{DCC} = \sum_{\Omega} \Psi(Z(x+u, y+v) - Z(x, y) - W(x, y)) \quad (8)$$

which are utilized by most RGB-D scene flow estimation methods [81, 84, 38, 56, 57, 35]. For robustness against varying illumination, Luckins added diverse additional color constraint, e.g., RGB,  $l*a*b$ , and hue[83], and Schuchert combining intensity constancy along with gradient constancy for per-pixel estimation [30]. Sun added layering and occlusion reasoning penalty for better performance [49].

Table 3 enumerates the typical RGB-D scene flow estimation methods with diverse choices of data terms.

### 3.2.4 Light field data

Light field data has enabled image refocusing and depth estimation with rich information [88]. That is to say, it can not only be treated as a depth data source, but provides much more information for constraint and regularization. Srinivasan was the first and till now the only one that utilized a light field camera Lytro Illum for scene flow estimation [48]. He proposed an oriented light-field window

Literature	Year	Data term	Description
Spies[81]	2000	$E_{BC}, E_{DCC}$	Introducing the range flow.
Luckins[83]	2005	$E_{DCC}, E_{additional}$	Additional color constraint
Schuchert[30]	2010	$E_{BC}, E_{GC}, E_{DCC}$	Combining both intensity and gradient constancy constrains.
Gottfried[84]	2011	$E_{BC}, E_{DCC}$	First utilizing consumer RGB-D camera Kinect v1.
Quiroga[56]	2013	$E_{BC}, E_{DCC}$	Utilizing a duality-based optimization for efficient solution.
Sun[49]	2015	$E_{BC}, E_{DCC}$	Along with penalty for layering and occlusion.

Table 3: Typical RGB-D scene flow estimation methods

method as a matching manner and embedded it with the common RGB-D scene flow estimation framework, where depth data was acquired using the method proposed by Tao [89]. In terms of data term, he only took brightness constancy of the oriented light field window into consideration, which is illustrated in Equation 9.

$$E_D = \sum_{\Omega} \Psi(P_{Z_1, x_0+u, y_0+v}(x, y, u, v, t+1) - P_{Z_0, x_0, y_0}(x, y, u, v, t)) \quad (9)$$

where  $P$  is the full oriented light field window operator. The penalty function is the L2 norm  $\Psi(x) = x^2$ .

Srinivasan's paper brings us with new ideas to extend the scene flow estimation area on account that light field data compensates the shortage of depth data source in terms of the sensing range and the robustness under an outdoor scene.

### 3.3 Calculation scheme

Scene flow estimation can be viewed as an energy function minimization in general, however, it is solved under different framework with diverse strategies and ideas. We categorized calculation schemes into these four main kinds: *global variational method*, *pixel assignment method*, *feature matching method* and *learning based method*. The boundaries between each kind are not always clear, but we focus on the overall solution idea to show the difference between each kind. Besides, we point out differences between each method in the same category in particular.

### 3.3.1 Global variational method

Global variational method has always been a classical method for either the optical flow or the scene flow estimation. As Equation 1 presents, while data terms provide local constraints to keep consistency, regularization terms give global propagation that yield a dense estimation, the motion field is constrained both locally and globally under the total variational(TV) framework.

**Regularization term** Under the total variational framework, a total variational(TV) regularizer is commonly chosen as the regularization term, or in other words, smoothness term. TV regularizer in terms of binocular or RGB-D can be formulated in Equation 10.

$$E_R^{binocular} = \sum_{\Omega} \Psi(|\nabla u|^2 + |\nabla v|^2 + \lambda|\nabla d|^2 + \mu|\nabla \delta d|^2) \quad E_R^{RGB-D} = \sum_{\Omega} \Psi(|\nabla u|^2 + |\nabla v|^2 + |\nabla W|^2) \quad (10)$$

Besides, Basha estimated scene flow under a multi-view 3D point cloud[54]. Thus the optical flow constraint was replaced by the three-dimensional scene flow constraint, and the smoothness of depth was added to the regularization term to penalize the shape. Zhang modified the regularization term with an anisotropic smoothness term to choose a more reliable pixel change between depth and appearance[85]. A bilateral filtering was utilized for edge-preserving as well. And a tensor voting approach was utilized under the multi-view stereopsis [67], where scene flow was refined in terms of direction by tensor voting in the temporal neighborhood and in terms of magnitude by a physical property between two frames.

Nevertheless, Vogel stated that TV regularizer is not good for scene flow estimation on account that it cannot handle discontinuities in the depth direction [78]. In his paper, scene flow was estimated by simultaneously regularizing the global rigid motion and the local non-rigid residual, where the regularization term consists of TV regularizer and a local rigidity prior penalized by Lorentzian function  $\Psi(s) = \log(1 + \frac{s}{2\sigma^2})$ .

**Optimization** Global variational methods utilizes partial differential equation to turn the energy minimization problem into a Euler-Lagrange equation solving problem, where Euler-Lagrange equation can be linearized by the Taylor expansion and then solved with iterations, and a coarse-to-fine optimization is commonly implemented during the iterations to handle large displacement. Normally, the global variational methods are solved under two nested iterations. Taken the binocular-based

scene flow as an example, the inside iteration utilizes the successive over-relaxation(SOR) iteration to compute small increments of scene flow as  $(\delta u, \delta v, \delta d, \delta d')$ , and the coarse-to-fine optimization serves as the outside iteration to update the warping result  $u_0 + \delta u, v_0 + \delta v, d_0 + \delta d, d'_0 + \delta d'$  and leads to the final result  $(u, v, d, d')$  for the full resolution.

Moreover, Zach introduced a duality-based method to the optical flow estimation for optimization in 2007 and achieved real-time estimation [90]. It separates the energy function and holds them into the same framework for parallel optimization, which remarkably lowers the computational cost and complexity without accuracy loss. Following Zach's paper, Quiroga implemented an auxiliary flow in the energy function, which decompose the minimization into two simpler issues [59]. By alternating the updating of the scene flow and the auxiliary flow, the problem can be solved with great efficiency. A brief scheme is illustrated as follows:

**Step 1** With auxiliary flow  $\mathbf{V}'$  introduced to the problem, it can be solved by alternating the updating both  $\mathbf{V}'$  and  $\mathbf{V}$ . Then the energy function defined in Equation 1 turns to:

$$E(\mathbf{V}, \mathbf{V}') = E_D(\mathbf{V}) + \alpha E_R(\mathbf{V}') + \frac{1}{2\theta} |\mathbf{V} - \mathbf{V}'|^2$$

where  $\theta$  is a small constant.

**Step 2** By fixing the scene flow  $\mathbf{V}$ , the auxiliary flow  $\mathbf{V}'$  can be solved by minimizing:

$$\sum_{\mathbf{X}} \frac{1}{2\theta} |\mathbf{V}'(\mathbf{X}) - \mathbf{V}(\mathbf{X})|^2 + \alpha E_R(\mathbf{V}'(\mathbf{X}))$$

**Step 3** By fixing the auxiliary flow  $\mathbf{V}'$ , the scene flow  $\mathbf{V}$  can be solved by minimizing:

$$E_D(\mathbf{V}) + \sum_{\mathbf{X}} \frac{1}{2\theta} |\mathbf{V}(\mathbf{X}) - \mathbf{V}'(\mathbf{X})|^2$$

Similarly, Ferstl embedded primal-dual algorithm into a coarse-to-fine framework [57, 58]. A total generalized variation regularization(TGV) [91] along with an anisotropic diffusion tensor was utilized to preserve edges. Moreover, Jaimez achieved a real-time RGB-D scene flow estimation [8].

The brief summary of the typical global variational methods are illustrated in Table 4.

### 3.3.2 Pixel assignment method

pixel assignment methods assumes local rigidity where pixels in a small region shares the same motion. It consists of three steps: pixel assignment, region motion estimation, and the compensation for each

Literature	Year	Regularization term	Optimization
Huguet[44]	2007	smoothness of optical flow	SOR+coarse-to-fine
		smoothness of disparity and disparity change	
Gottfried[38]	2013	smoothness of optical flow	SOR+coarse-to-fine
		smoothness of depth	
Basha[54]	2010	smoothness of scene flow	SOR+coarse-to-fine
		smoothness of depth	
Jaesik[67]	2012	smoothness of scene flow magnitude	-
		tensor voting smoothness of scene flow direction	
Zhang[85]	2013	anisotropic smoothness of optical flow	SOR+coarse-to-fine
		anisotropic smoothness of depth	
Ferstl[57]	2014	anisotropic TGV [91] regularizer	duality-based method + coarse-to-fine
Quiroga[59]	2014	smoothness of rotation field $\omega$	duality-based method
		smoothness of translation field $\tau$	

Table 4: The brief summary of the typical global variational methods

pixel. Each pixel is assigned to a specific region with prior knowledges. Then the center motion of each region is estimated, while a small motion residual of each pixel is tolerated and compensated by refinement afterwards. In this way the method combines both the global denseness and the local computational effectiveness, with discontinuity preserved well simultaneously.

Zhang first introduced this idea by fitting an affine motion model to each segment with global smoothness constraint in the early stage [51]. The scene was set under a multi-view system without rigid assumption. It is followed by Li to apply this kind of method under a binocular setting [52].

In the last decade, this pixel-to-segment scene flow estimation draws people's attention for its superiority. Popham gave a pixel-to-patch assignment [70, 32]. The motion of each patch was estimated through a common variational method solved by Gauss-Seidel iteration, and the motion field of pixels in each patch were interpolated with a measurement covariance. Jaimez jointly estimated motion and segmentation [86]. The scene was assumed to be segmented into several labels and the pixel-to-segment issue was seen as a labelling problem, while lareweighedbelming is involved in the regularization. Scene flow for each segment is estimated with the global iteratively reweighed least squares(IRLS) minimization. Sun handled the issue with multiple hypothesis to jointly obtain oc-

clusion reasoning, motion estimation and the scene segmentation [49]. With a cheap acquisition of depth information, reliable layer ordering information can be obtained easily for the global-rigid, local-flexible motion field estimation.

Vogel solved the issue by simultaneously looking for a pixel-to-segment mapping and the segment motion [71]. The energy function can be formulated as:

$$E(\mathbf{P}, \mathbf{S}) = E_D(\mathbf{P}, \mathbf{S}) + \lambda E_R(\mathbf{P}, \mathbf{S}) + \mu E_S(\mathbf{S}) \quad (11)$$

where  $\mathbf{S} : I \rightarrow S$  is the pixel-to-segment mapping which assign each image pixel  $\mathbf{p} \in I$  to a segment  $s \in S$ .  $\mathbf{P} : S \rightarrow \Pi$  is the segment motion mapping which assign each segment to a 3D rigidly moving plane  $\pi \in \Pi$ .  $E_D$  is the data term and  $E_R$  is the TV regularizer, and  $E_S$  is an additional segmentation regularization term to refine segmentation during iterations.

Using a superpixel segmentation for initialization [92], the energy is alternatively optimized using fusion moves [93] and quadratic pseudo-boolean optimization (QPBO) [94], and it achieves the state-of-the-art performance. The general idea is depicted in Figure 6. Afterwards, he introduced a temporal window to enforce coherence over long time intervals [72], and proposed a detailed version with deep analysis and thorough comparisons later that forms the whole theoretical framework [73].

Similarly, in 2016, Lv follows the idea by representing the dynamic 3D scene as a collection of rigidly moving planar segments [95]. The complex assignment problem is formulated with a factor graph formulation [96], estimated as a non-linear least square problem and then optimized locally and globally.

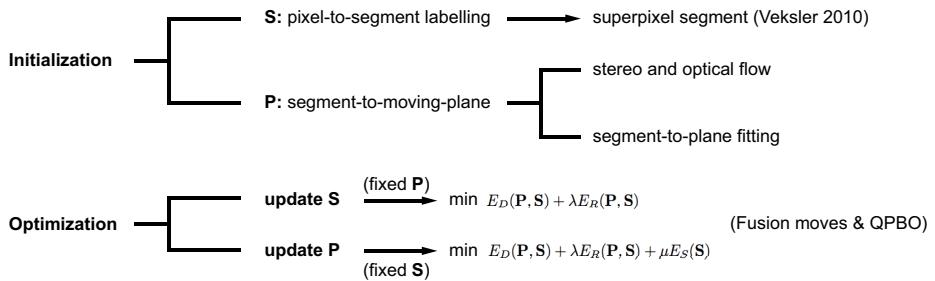


Figure 6: Flowchart of the pixel-to-segment assignment scene flow estimation [71]

The brief summary of the typical pixel assignment methods are illustrated in Table 5.

Literature	Year	Assignment	Optimization
Vogel[71]	2013	color pixel-to-segment (superpixel segmentation)	fusion moves + QPBO
Popham[32]	2014	color pixel-to-patch (belief propagation)	Gauss-Newton minimization
Sun[49]	2015	depth layering (K-means)	coordinate descent minimization
Jaimez[86]	2015	color pixel-to-segment (K-means)	IRLS + coarse-to-fine method
Lv[95]	2016	color pixel-to-segment (superpixel segmentation)	Levenberg-Marquardt

Table 5: The brief summary of the typical pixel assignment methods

### 3.3.3 Feature matching method

Feature matching method mainly consists of three steps: feature extraction, feature matching, and propagation. While stereo estimation aims to find the spatial correspondence, scene flow seeks for temporal coherence. Thus, this kind of feature matching methods for scene flow estimation is suitable for binocular-based methods with stereo estimation solved simultaneously. Cech [46] made use of Harris points as the feature and a multi-scale LK tracker [3] as a prematcher. The sparse estimation was then propagated to dense result using seeded growing method in terms of both stereo and motion. Sizintsev introduced a spatiotemporal quadric element(stequel) matching method [77]. Stereo correspondence and scene flow were estimated through the match cost solution with multiple constraints. Richardt utilized the DAISY descriptor [80] for computing correspondence between views and frames [60], and the dense scene flow was refined under the conventional variational framework.

Moreover, matching method is also implemented with an RGB-D camera. Letouzey matched SIFT features as a constraint and minimized the energy function as a preliminary estimation [66]. By re-projecting this preliminary estimation into the image plane, a preliminary map can be viewed as the initial value, where the SIFT features are the non-moving anchor points. Quiroga enforced consistency of scene flow with a sparse set of SURF features [56]. The features were extracted in the color image while depth information was taken as the matching constraint. Hornacek proposed a patch-wise estimation without assuming brightness constancy [74]. The matching cost takes both three-channel CIE l\*a\*b information and gradient information into consideration, initialized by SURF features. The pixel with tiny matching cost will be viewed with the same motion for propagation. On basis of Hornacek, Alhaija took edge as the sparse matching feature with a graph matching approach

to handle large displacement and acquired promising results [29].

The brief summary of the typical feature matching methods are illustrated in Table 6.

Literature	Year	Feature Function	Regularization term	Optimization
Cech[46]	2011	SIFT features	smoothness of optical flow	-
		seed for propagation	smoothness of disparity	
Letouzey[66]	2011	Harris point	smoothness of scene flow	Jacobi minimization
		sparse matching constraint		
Quiroga[56]	2013	SURF features	smoothness of scene flow	duality-based method
		sparse matching constraint	smoothness of depth	
Hornacek[74]	2014	SURF features	smoothness of scene flow	fusion moves + QPBO
		for patch generation		
Alhaija[29]	2015	SIFT features	smoothness of scene flow	fusion moves + QPBO
		for edge descriptor generation	projection consistency term	
Richardt[60]	2017	DAISY [80]	smoothness of optical flow	PatchMatch belief propagation
		sparse matching initialization	smoothness of disparity change	
			epipolar constraint	

Table 6: The brief summary of the typical feature matching methods

### 3.3.4 Learning-based method

Learning method has got enough attention over the past few years for solving computer vision tasks. To solve scene flow estimation, learning method can be utilized twofold. On the one hand, parameters as a part of the whole pipeline can be learned to enhance the robustness or the efficiency. Hadfield, for example, used machine learning technique to introduce an intelligent cost function as a penalization metric with limited improvement [97].

On the other hand, learning can be utilized for a per-pixel end-to-end estimation. Currently speaking, due to lack of large-scale dataset with ground truth and appropriate models, Mayer is the only one who utilized convolutional neural network(CNN) [98] to learn scene flow estimation [4], which reveals the embryo of learning scene flow. He also introduced a large scale dataset for training, which will be introduced in Section 4.3.6. Optical flow estimation and disparity estimation were decoupled, where the disparity estimation network named "*DispNet*" was presented on basis of

the proposed FlowNet optical flow network [99]. Each network consisted of a contracting part for feature contraction and an expanding part utilizing up-convolutional layers and un-pooling for final estimation. A loss weight scheme was implemented to balance the weight between high resolution and low resolution. The downsampling factor in total is 64, and the network consider a maximum displacement of 160 pixels in the input images, which is much superior to a 4-level pyramid with a 50% downsampling factor.

## 4 Evaluation

In this section, we enumerate and analyze the regular evaluation protocols and publicly-available datasets.

### 4.1 Evaluation protocols

Scene flow is made up of 3D motion vectors, which contain range and direction information of displacement. Thus, the error is measured twofold in terms of magnitude and angle. Currently, the error is measured in three ways: the end-point error(EPE), the root mean square error(RMSE), and the average angular error(AAE). A brief introduction is presented as follows.

#### 4.1.1 Two-dimensional error measure

Due to the fact that main datasets only provide ground truth under a two-dimensional representation in terms of optical flow and disparity, most methods re-project scene flow onto the image space or simply represent scene flow with a two-dimensional representation mentioned in Section 3.1.1, and the error is measured in terms of optical flow and disparity. But first, we'll present the fundamental error measure named absolute error:

**Absolute error** Absolute error describes the absolute magnitude difference, which is the euclidean distance between the endpoints of ground truth vector and the estimated vector. It is presented in terms of optical flow and disparity in Equation 12.

$$E_{of} = |(\mathbf{v}_e - \mathbf{v}_g)| = \sqrt{(u_e - u_g)^2 + (v_e - v_g)^2} \quad E_{disp} = |(d_e - d_g)| \quad (12)$$

where subscript  $e$  annotates the estimated value, and subscript  $g$  annotates the ground truth.

**The average endpoint error(EPE)** EPE is introduced by Otte [10]. It's the mean of absolute error among all the pixels as Equation 13 presents.

$$E_{EPE_{of}} = \frac{1}{n} \sum_{i \in \Omega} E_{of}^i = \frac{1}{n} \sum_{i \in \Omega} \sqrt{(u_e^i - u_g^i)^2 + (v_e^i - v_g^i)^2} \quad E_{EPE_{disp}} = \frac{1}{n} \sum_{i \in \Omega} E_{disp}^i = \frac{1}{n} \sum_{i \in \Omega} |(d_e^i - d_g^i)| \quad (13)$$

where  $n$  is the number of pixels, and  $\Omega$  is the entire image plane.

**The root mean square error(RMSE)** While EPE indicates the overall accuracy level, RMSE indicates both the error distribution and overall accuracy level as  $E_{RMSE}^2 = \sigma_{AE}^2 + E_{EPE}^2$ <sup>1</sup>.

RMSE in terms of optical flow and disparity can be presented as Equation 14a and 14b.

$$E_{RMSE_{of}} = \sqrt{\frac{1}{n} \sum_{i \in \Omega} E_{of}^i} = \sqrt{\frac{1}{n} \sum_{i \in \Omega} ((u_e^i - u_g^i)^2 + (v_e^i - v_g^i)^2)} \quad (14a)$$

$$E_{RMSE_{disp}} = \sqrt{\frac{1}{n} \sum_{i \in \Omega} E_{disp}^i} = \sqrt{\frac{1}{n} \sum_{i \in \Omega} |(d_e^i - d_g^i)|^2} \quad (14b)$$

In addition, to better measure the error in different scales, the normalization is needed. The normalized root mean square error(NRMSE) is scaled by the ground truth, which can be compared between different datasets. NRMSE in terms of optical flow and disparity can be presented as Equation 15.

$$E_{NRMSE_{of}} = \frac{E_{RMSE_{of}}}{\max \| \mathbf{v}_g \| - \min \| \mathbf{v}_g \|} \quad E_{NRMSE_{disp}} = \frac{E_{RMSE_{disp}}}{\max |d_g| - \min |d_g|} \quad (15)$$

**The average angular error(AAE)** The average angular error was introduced by Fleet in 1990 [100], which is introduced to measure the optical flow error deviation of angle:

$$E_{AAE_{of}} = \frac{1}{n} \sum_{i \in \Omega} \arctan\left(\frac{u_e^i v_g^i - u_g^i v_e^i}{u_e^i u_g^i + v_e^i v_g^i}\right) \quad (16)$$

It can also be calculated with  $\arccos(\mathbf{v}_g \cdot \mathbf{v}_e)$  or  $\arcsin(\mathbf{v}_g \times \mathbf{v}_e)$ .

$$\begin{aligned} \sigma_{AE}^2 &= \frac{1}{n} \sum_{i \in \Omega} [\sqrt{(u_e^i - u_g^i)^2 + (v_e^i - v_g^i)^2} - E_{EPE}]^2 = \frac{1}{n} \sum_{i \in \Omega} [(u_e^i - u_g^i)^2 + (v_e^i - v_g^i)^2 - 2E_{EPE} \sqrt{(u_e^i - u_g^i)^2 + (v_e^i - v_g^i)^2} + E_{EPE}^2] \\ &= \frac{1}{n} \sum_{i \in \Omega} [(u_e^i - u_g^i)^2 + (v_e^i - v_g^i)^2] - \frac{2}{n} E_{EPE} \sum_{i \in \Omega} \sqrt{(u_e^i - u_g^i)^2 + (v_e^i - v_g^i)^2} + \frac{1}{n} \sum_{i \in \Omega} E_{EPE}^2 \\ &= E_{RMSE}^2 - 2E_{EPE}^2 + E_{EPE}^2 = E_{RMSE}^2 - E_{EPE}^2 \end{aligned}$$

#### 4.1.2 Three-dimensional error measure

According to Equation 3:

$$(X, Y, Z) = \left( \frac{(x - c_x)b}{d}, \frac{(y - c_y)b}{d}, \frac{f * b}{d} \right) \quad (X + U, Y + V, Z + W) = \omega_c((x + u - c_x), (y + v - c_y), f) \quad (17)$$

where  $\omega_c = \frac{b}{d + \delta d}$ . The focal length of each camera is assumed the same, as  $f_x = f_y = f$ .

Hence, the transfer function between the three-dimensional point cloud scene flow representation and two-dimensional representation can be illustrated in Equation 18.

$$(U, V, W) = \omega_c(u - \frac{\delta d(x - c_x)}{d}, v - \frac{\delta d(y - c_y)}{d}, -\frac{f * \delta d}{d}) \quad (18)$$

We can see clearly that the projection between scene flow and optical flow along with disparity and disparity change information is complex, and the accuracy of disparity change  $\delta d$  can significantly affect the result. That is to say, the protocol mentioned in Section 4.1.1 is not sufficient enough to evaluate scene flow.

Wedel proposed a easy way for three-dimensional error measure[45]. The RMSE and AAE are modified as Equation 19a and 19b.

$$E_{RMSE_{wedge}} = \sqrt{\frac{1}{n} \sum_{i \in \Omega} \| (u_e^i, v_e^i, d_e^i, \delta d_e^i) - (u_g^i, v_g^i, d_g^i, \delta d_g^i) \|} \quad (19a)$$

$$E_{AAE_{wedge}} = \frac{1}{n} \sum_{i \in \Omega} \arccos\left(\frac{u_e^i u_g^i + v_e^i v_g^i + \delta d_e^i \delta d_g^i + 1}{\sqrt{(u_e^i)^2 + (v_e^i)^2 + (\delta d_e^i)^2 + 1)(u_g^i)^2 + (v_g^i)^2 + (\delta d_g^i)^2 + 1)}}\right) \quad (19b)$$

However, this may not precisely reveal the contribution to error measure between different unknowns. Basha provided a three-dimensional point cloud ground truth  $\mathbf{V}_g = (U_g, V_g, W_g)$ , which made the three-dimensional error measure feasible [54]. The absolute error can be measured in a three-dimensional way:

$$E_{sf} = |(\mathbf{V}_e - \mathbf{V}_g)| = \sqrt{(U_e - U_g)^2 + (V_e - V_g)^2 + (W_e - W_g)^2} \quad (20)$$

Hence, the EPE, RMSE, NRMSE, and AAE can be modified as:

$$\begin{aligned} E_{EPE_{sf}} &= \frac{1}{N} \sum_{P \in \mathbb{S}} E_{sf}^P & E_{RMSE_{sf}} &= \sqrt{\frac{1}{N} \sum_{P \in \mathbb{S}} E_{sf}^P} & E_{NRMSE_{sf}} &= \frac{E_{RMSE_{sf}}}{\max \|\mathbf{V}_g\| - \min \|\mathbf{V}_g\|} \\ E_{AAE_{sf}} &= \frac{1}{N} \sum_{P \in \mathbb{S}} \arccos\left(\frac{U_e^P U_g^P + V_e^P V_g^P + W_e^P W_g^P + 1}{\sqrt{(U_e^P)^2 + (V_e^P)^2 + (W_e^P)^2 + 1)(U_g^P)^2 + (V_g^P)^2 + (W_g^P)^2 + 1)}}\right) \end{aligned} \quad (21)$$

where  $\mathbb{S}$  is the surface in the three-dimensional space.

#### 4.1.3 Special metrics

Besides the protocols mentioned above, KITTI dataset [23] and Sintel dataset [101] analyzes EPE under different circumstances and introduced the special metric for evaluation.

**KITTI metric** KITTI metric is the specific criterion for performance evaluation on KITTI dataset [23]. It employed an EPE threshold of  $\tau \in (2, \dots, 5)$  pixel, and calculated the portion of pixels whose endpoint error in terms of optical flow and disparity is above the threshold among the entire image (threshold default is set as  $3px$ ). With occlusion ground truth, the official protocol is presented as:

Method	Setting	Out-Noc	Out-All	Avg-Noc	Avg-All	Density	Runtime	Environment
--------	---------	---------	---------	---------	---------	---------	---------	-------------

In 2015, Menze modified the dataset by adding background and foreground annotation [39], and percentage of outliers in terms of background and foreground are distinguished in the evaluation metric. Moreover, a scene flow error was introduced. if either the disparity or the optical flow end-point error is above the threshold(where the default is 3 pixels), then the pixel is viewed as the scene flow outlier. The official protocol is presented as:

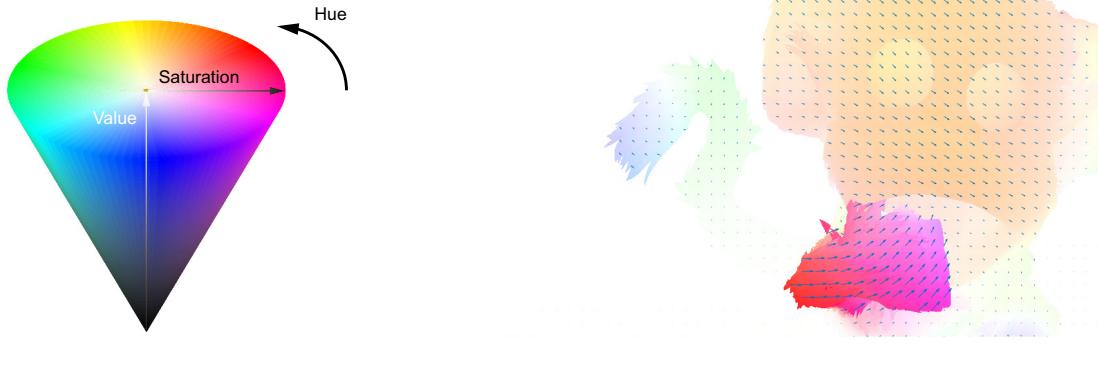
Method	D1-bg	D1-fg	D1-all	D2-bg	D2-fg	D2-all	F1-bg	F1-fg	F1-all	SF-bg	SF-fg	SF-all
--------	-------	-------	--------	-------	-------	--------	-------	-------	--------	-------	-------	--------

**Sintel metric** Sintel metric provides a thorough evaluation for Sintel benchmark [101]. It employed EPE as error measure as well. Particularly, it measure the error distribution in terms of both occlusion and large displacement with different threshold, which clearly reveal the performance under these two fundamental challenges. Moreover, the percentage of error pixels that remain visible in adjacent frames are taken as a criterion that reveal the temporal distribution of error. Thus, this metric provides sufficient information for evaluation and comparison. The official protocol is presented as:

Method	EPE all	EPE matched	EPE unmatched	d0-10	d10-60	d60-140	s0-10	s10-40	s40+
--------	---------	-------------	---------------	-------	--------	---------	-------	--------	------

## 4.2 Color decoding for visualization

For better evaluating and analyzing scene flow performance, the color decoding manners are proposed for visualizing optical flow and error map, which make the it clear and direct to quantitatively evaluate the performance.



(a) The HSV cone model

(b) An example of Monkaa optical flow ground truth colorization

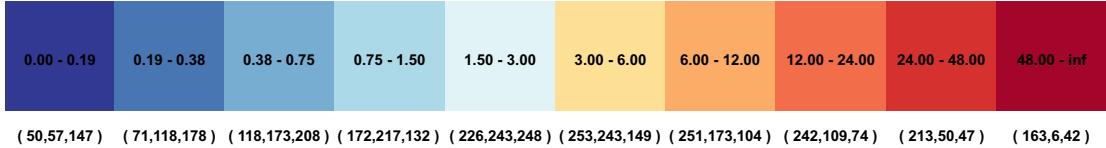
Figure 7: Brief sketches about optical flow colorization

**Optical flow** Optical flow is a motion field composed of two-dimensional vectors. Normally the optical flow are encoded with an HSV color space. "H" stands for hue, of which the range is  $[0, 360^\circ]$ . We use the angular between the optical flow vector and x axis in the image plane as the hue value. "S" stands for saturation, which is represented by the ratio between the magnitude of each vector and the maximum magnitude among the whole motion field. "V" stands for value. Normally the value in non-occluded area is set as 0, while the value in occluded area is set as 1. In this way the motion vectors in the same direction share the same hue with different saturation to indicate the magnitude degree, the non-moving region is pure white, and the occluded region is black. The HSV model for optical flow colorization is illustrated in Figure 7.

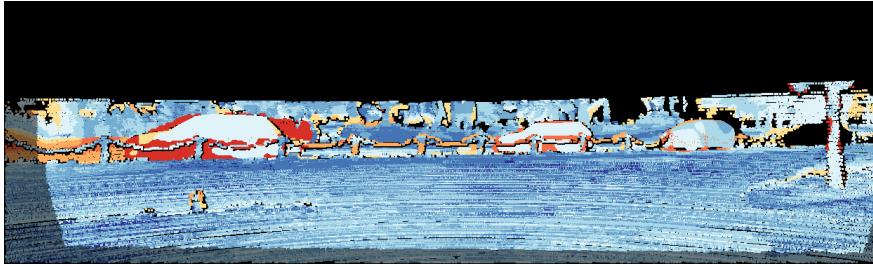
**Error map** Menze introduced an error visualization to directly distinguish the error distribution in the image plane. The correct pixel is depicted in blue while the error pixels are depicted in red. In this way can we directly find the rule for major error distribution. The corresponding relation between color and endpoint error for each pixel is presented in Figure 8.

### 4.3 dataset

Existing datasets mainly serve for scene flow evaluation under a binocular setting, which consist of optical flow ground truth and disparity ground truth. This can also be used for evaluating stereo and optical flow estimation. Due to the lack of RGB-D dataset, to evaluate RGB-D scene flow, the disparity map of each dataset should be converted to the depth map with Equation 2 as an input.



(a) Error map visualization. Each pixel in the error map is colorized by finding the specific interval where the EPE value is in, and the color is chosen with RGB value corresponding to the interval as illustrated below.



(b) An example [73] of scene flow error colorization in KITTI 2015 dataset

Figure 8: Brief sketches about error colorization map

As is mentioned in Section 3.1, scene flow can be represented as  $\mathbf{v}(u, v, \Delta d)$  or  $\mathbf{V}(\Delta X, \Delta Y, \Delta Z)$ . While most datasets only provided optical flow and disparity ground truth, Basha [54] provided the three-dimensional ground truth  $\mathbf{V}(\Delta X, \Delta Y, \Delta Z)$ , and Freiburg dataset [4] provided additional disparity change ground truth that truly represent scene flow. Moreover, Middlebury [102, 103], Basha [54] and KITTI [23, 39] provide occlusion ground truth so that the error out of occluded region can be evaluated separately.

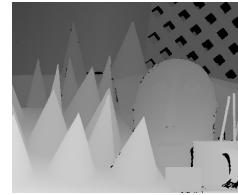
In the following section, we briefly introduced each commonly used dataset. A thorough information is illustrated in Table 7. Sample images including color image, optical flow ground truth and disparity ground truth of each dataset is presented in Figure 9 and 10.

#### 4.3.1 Middlebury dataset

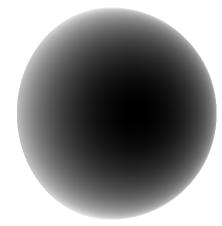
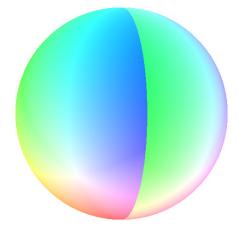
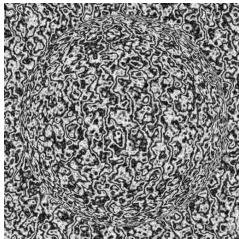
The Middlebury stereo dataset [102, 103] is commonly used as a quantitatively evaluation benchmark for optical flow and stereo matching. Particularly, the subsets named Teddy, Cones and Venus provided both optical flow and disparity ground truth, and hence it is utilized for scene flow evaluation [105, 44, 55, 54, 106, 56, 85, 57, 58, 87, 74, 59, 49, 48]. Each subset simulates a simple translational motion along the X axis. The 8 cameras are rectified and placed parallelly and equally along the X axis and thus the scene motion projected onto the image plane is the disparity between two cameras. Under a binocular setting, images from

Literature year	Ground truth	Resolution	Number[T] (tr, te)	Annotations	URL
Middlebury [102], [103] 2001, 2003	OF, Disp, Occ, Te, DC	450×375	3	Cones Teddy Venus	<a href="http://vision.middlebury.edu/stereo/data/">http://vision.middlebury.edu/stereo/data/</a>
Rotating sphere 2007, 2010	OF, Disp SF, Occ	512×512	1	Huguet Basha	<a href="http://devernay.free.fr/vision/varseneflow/">http://devernay.free.fr/vision/varseneflow/</a> <a href="http://people.csail.mit.edu/talidekel/MVSF.html">http://people.csail.mit.edu/talidekel/MVSF.html</a>
EISATS [104] 2008	OF, Disp, DispC	640×480	498	Set 2	<a href="http://ccv.wordpress.fos.auckland.ac.nz/eisats/">http://ccv.wordpress.fos.auckland.ac.nz/eisats/</a>
KITTI [23] 2012	OF, Disp, Occ, Lb	1238×374	384 (194,195)	-	<a href="http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=stereo">http://www.cvlibs.net/datasets/kitti/eval_stereo_flow.php?benchmark=stereo</a>
KITTI [39] 2015	OF, Disp, Occ	1242×375	400 (200,200)	-	<a href="http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php">http://www.cvlibs.net/datasets/kitti/eval_scene_flow.php</a>
MPI Sintel [101] 2012	OF, Dis, Dep Occ, Seg, CM	1024×436	1828 (1264,564)	-	<a href="http://sintel.is.tue.mpg.de/downloads">http://sintel.is.tue.mpg.de/downloads</a>
Freiburg [4] 2016	OF, Disp, DispC Occ, Seg, MB	960×540	39049 (34801,4248)	FlyingThings3D Driving Monkaa	<a href="http://lmb.informatik.uni-freiburg.de/resources/datasets/SceneFlowDatasets.en.html">http://lmb.informatik.uni-freiburg.de/resources/datasets/SceneFlowDatasets.en.html</a>

Table 7: A summary of the popular datasets. Notation for headers: **[T]**: number of scenes in total, **tr**: numbers of training images, and **te**: numbers of testing images. Notation for data format: OF: optical flow ground truth, Disp: disparity ground truth, DispC: disparity change ground truth, Dep: depth map ground truth, SF: point cloud scene flow ground truth, CM: camera motion, Seg: segmentation, Lb: semantic and instance labels and car labels, Occ: occlusion region ground truth, MB: motion boundaries, Te: textureless region ground truth, DC: discontinuity region ground truth.



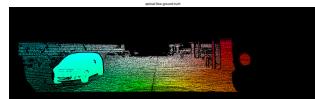
(a) Middlebury 2003 - Cones



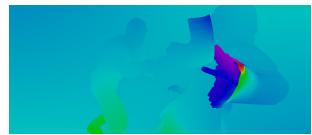
(b) Huguet - rotating sphere



(c) EISATS



(d) KITTI 2015



(e) MPI Sintel

Figure 9: Sample images with stereo and optical flow ground truth. The first row presents the left-view color image, the second row presents the color-encoded optical flow, and the third row presents the left-view corresponding disparity map.

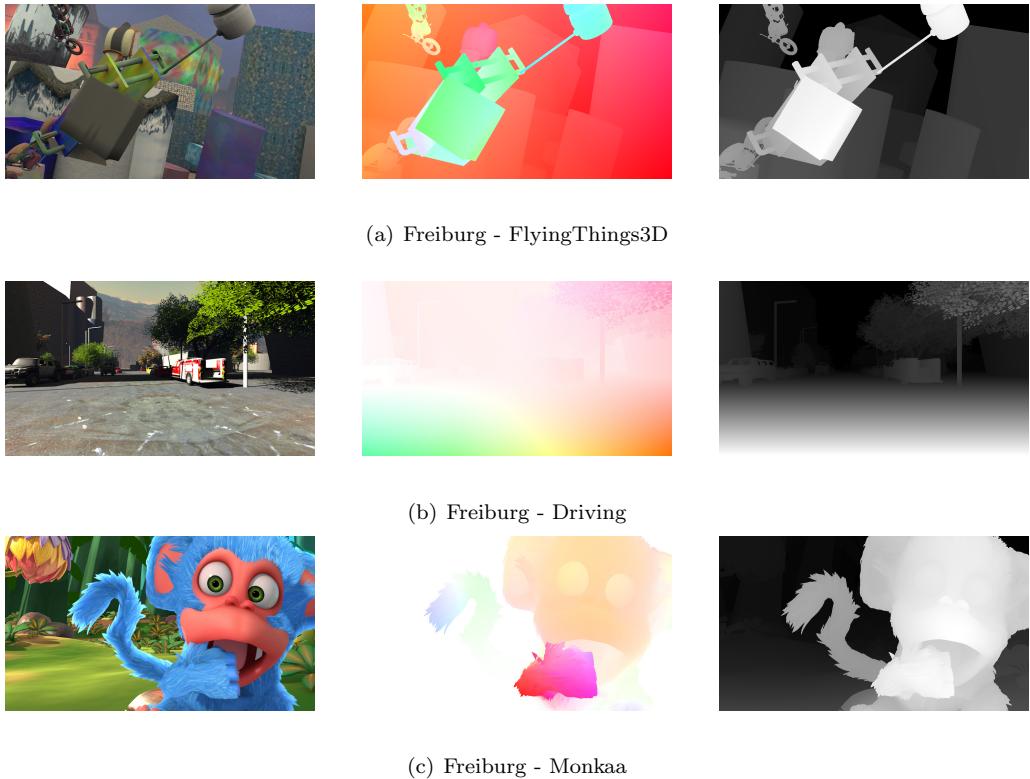


Figure 10: Sample images of Freiburg dataset. The first row: the left-view color image, the second row: the color-encoded optical flow ground truth, the third row: the left-view corresponding disparity ground truth.

camera 2 and 6 are taken as stereo pairs from time  $t$ , while images from camera 4 and 8 are taken as stereo pairs from time  $t+1$ . The disparity ground truth is the disparity from camera 2 to 6, and the optical flow ground truth is the disparity from camera 2 to 4. Similarly, when it comes to RGB-D scene flow, the ground truth disparity map is converted into depth channel for evaluation.

#### 4.3.2 Rotating sphere

In 2007, Huguet utilized the Pov-Ray to render a publicly available rectified synthetic rotating sphere with optical flow and stereo ground truth [44], which is commonly used as a benchmark [45, 79, 46, 78, 7, 71, 39]. Two hemispheres rotate in opposite directions which lead to strong discontinuity. Basha modified the rotating sphere by adding a rotating plane behind the rotating sphere with OpenGL [54]. It was used for evaluating three-dimensional parametrization scene flow on account of the ground truth it provided [67, 106]. Moreover,

it provided five rectified views which can also be evaluating multi-view scene flow, without full-view geometry ground truth though.

#### 4.3.3 EISATS dataset

EISATS traffic scene datasets is a synthetic stereo image sequence rendered by Pov-Ray with ground truth for both stereo and motion [104]. The sequence 1 consists of 100 frames while sequence 2 consists of 396 frames with ego-motion. The synthetic traffic scene consists of a few moving cars under an open environment. Only a few papers utilized this stereo pairs sequence for binocular scene flow estimation [45, 6, 7, 47].

#### 4.3.4 KITTI dataset

Geiger took advantage of their calibrated autonomous driving platform and developed the novel challenging KITTI benchmark with 194 training scenes and 195 test scenes in 2012 [23], and Menze annotated the dynamic scenes with 3D CAD models for all vehicles in motion and obtained a modified dataset with 200 training scenes and 200 test scenes in 2015 [39]. A novel evaluation methodology is also introduced as the KITTI metric, which is illustrated in Section 4.1.3. These two binocular-based dataset have been utilized by multiple papers over the years [44, 46, 71, 74, 72, 27, 39, 73]. The scene is much more realistic and challenging compared to the early Middlebury dataset, which is designed specifically for autonomous driving. However, due to the acquisition manner of data, there are missing value in both optical flow and disparity ground truth, as is illustrated in Figure 9. The density of ground truth is about 75% to 90%. Thus, KITTI dataset is not recommended for RGB-D scene flow evaluation on account that it needs dense disparity ground truth for simulating depth data.

#### 4.3.5 MPI Sintel dataset

MPI Sintel dataset is the largest dataset before 2015 [101], which consists of 23 training sequences with 1064 frames and 12 test sequences with 564 frames in total. It derived from an open source animated film and the resolution is  $1024 \times 436$ . The scenes are designed to be strictly realistic with fog and motion blur added. Moreover, beta version depth data were then added which can be a perfect dataset for RGB-D scene flow evaluation. Zanfir and Jaimez [86, 28] utilized this dataset for scene flow evaluation in 2015 and gave quantitative analysis. It is highly recommended for its naturalistic setting and density, as well as its comprehensive evaluation protocol. Moreover, the video sequence ensures a multi-frame implement. With the development dataset of scene flow estimation, this dataset which consists of non-rigid motion and large

displacement under a high resolution is reliable and challenging enough for evaluation.

#### 4.3.6 Freiburg dataset

Freiburg dataset is the up-to-date and largest dataset for optical flow, stereo and scene flow evaluation [4]. The disparity change ground truth make up for the blank of scene flow ground truth. It contains 34801 stereo training frames and 4248 test frames in  $960 \times 540$  resolution, which intended for deep learning. The dataset consists of three subsets named FlyingThings3D, Monkaa and Driving, which are all made by the 3D suite Blender [107].

FlyingThings3D contains 21818 training frames with 2247 scenes and 4248 test frames. The amount of data is its first priority. This scene contains abundant texture features and complex occlusion between multiple objects, which is highly challenging.

Monkaa resembles the MPI Sintel dataset since it derived from an animated short film Monkaa as well. It is less naturalistic as Sintel, but the data scale is three times larger. Besides, the articulated motion and non-rigid motion with furry effect are the major issues to tackle.

The Driving scene resembles the KITTI dataset since the dynamic street scenes are captured from the view point of a driving car. The long frame range and fully dense ground truth ensure the reliable evaluation for scene flow under the particular task for autonomous driving.

Compared to other datasets, this dataset contains the largest and most comprehensive scenes and aims for scene flow evaluation particularly compared to other datasets. Besides the 2.5 tera-gigabytes of data in total, a sample pack is also available for less than 100 megabytes. Hence, this dataset is highly recommended for its comprehensiveness.

#### 4.3.7 Other datasets

Many other datasets with ground truth are introduced as well. However, on account of limited reference, the comparisons under these datasets are deficient, and most of them lack public availability and innovation. We list them as follows just in case.

Similar to the rotating sphere dataset created by Huguet, in the early stage, Zhang introduced a synthetic deformable sphere using OpenInventor for quantitative analysis [50]. Spies modeled a structured light sensor to provide a synthetic textured sphere as well [82]. Valgaerts generated a general rotating sphere scene without rectification [79], while Cech added a fast moving bar and a slanted background plane and then textured the whole scene with white noise using Blender [46], which made it much more challenging for the

scene flow estimation.

Moreover, Ferstl created a translating and rotating cube in front of the static plane with white noise textured [57, 58], and Ghuffar created a noisy scene with two cubes moving on a plane in front of a static wall to testify the algorithm’s robustness towards noise and occlusion [35]. Vogel generated nine synthetic box dataset with ground truth [78], which consisted of pure rotation, translations in all axes and translation only in depth for independent analysis.

In addition, early in 2005, Luckins created a sloped plane and a sinusoidal plaid pattern named "splaid" for a rough evaluation [83], the sampling is the size of  $100 \times 100$  with depth and RGB color. Gong generated a synthetic 3D scene that consists of a rotating earth model textured with Phong illumination and bump mapping and a translating galaxy background [5]. The scene is rendered with Gaussian noise and the camera is moving against the earth which makes it really tough for scene flow estimation. Ruttle utilized the Human Eva II dataset for motion tracking and pose estimation and gave the quantitative analysis [53]. Popham [70, 32] evaluated his algorithm with a multi-view motion capture dataset named "Katy" and "Skirt" [108] with sparse motion trajectory ground truth. Sizintsev captured few sets of sequences with BumbleBee stereo camera and got ground truth with a structure light approach [103], and then used this dataset for evaluation [77]. Alhaija captured seven pairs of images through Kinect to specifically evaluate scene flow under large displacement [29]. The matching ground truth was given by manually labeling each segment.

#### 4.3.8 Performance

The performance of the methods that first utilizing these datasets for scene flow evaluation is presented in Table 8 as benchmarks.

## 5 Discussion

This paper has categorized and analyzed most of scene flow estimation methods, few questions are arisen as follows, with our own opinions given as well. A future vision for scene flow estimation methods and scene flow evaluation protocols is depicted.

Literature	Year	Dataset	Error type	Error	Notes
Huguet [44]	2007	sphere(Huguet)	OF(RMSE,AAE)	0.69, 1.75	introducing the Huguet rotating sphere dataset and the first to use Middlebury dataset.
		Middlebury(Teddy)		2.85, 1.01	
		Middlebury(Cones)		3.07, 0.39	
Wedel [45]	2008	EISATS	OF(RMSE, AAE)	0.59, 4.13	the first to use EISATS dataset.
Basha [54]	2010	sphere(Basha)	SF(NRMSE, AAE)	9.71, 3.39	introducing the Basha rotating sphere dataset.
Vogel [71]	2011	KITTI2012	OF(EPE , %3px)	1.6, 7.07%	the first to use KITTI2012 dataset.
			Disp(EPE , %3px)	1.0, 4.87%	
Zanfir [28]	2015	Sintel	OF(EPE)	4.6	the first to use Sintel dataset.
Menze [39]	2015	KITTI2015	%3px(OF,Disp,SF)	8.37, 5.79,	introducing the KITTI2015 dataset.
				9.44	
Mayer [4]	2016	Freiburg Driving	EPE	22.01,17.56,16.89	introducing Freiburg dataset.
		Freiburg Flying	(OF,Disp,DispC)	13.45,2.37,0.91	
		Freiburg Monkaa		7.68,6.16,0.81	

Table 8: Summary of the methods that firstly utilized the datasets for scene evaluation. Notation for data format: OF: optical flow error. Disp: stereo matching error. SF: scene flow error. %3px: the percentage of error pixels of which the EPE is larger than 3px.

## 5.1 Which kind of methods performs better?

As the taxonomy Section 3 presents, most methods share the similar energy minimization framework. Each kind has its pros and cons. Scene flow estimation under the multi-view stereopsis provides comprehensive geometry and motion information, which handles occlusion with less error. However, the large data size leads to the poor efficiency. Binocular-based scene flow estimation deals with two ill-posed problem in terms of stereo and motion, but it can be widely applied in diverse scenes. RGB-D scene flow estimation utilized the cheap depth information with better efficiency and better robustness under a low illumination condition, but it can only be applied in an indoor circumstance due to the limitation of sensors. And light field camera is just an emerging technique without the mature application.

Calculation scheme is the key property of each algorithm. With four different categories introduced in Section 3.3, the tiny difference between methods that belongs to the same kind are presented in Table 4, 5, 6. Global variation methods mentioned in Section 3.3.1 is a common and general solution. Pixel assignment methods mentioned in Section 3.3.2 aims to handle the inaccuracy around the boundaries for better performance, but additional ill-posed segmentation issue is involved in the framework. Feature matching methods aims for specific issue like large displacement and alleviating complexity for better efficiency, but the propagation from sparse features is worth attention. Learning based methods arouse extensive attention, but this emerging technique remains a long way to go.

A two-dimensional diagram is depicted in Figure 11 to show the performances between efficiency and accuracy of the representative algorithms. Table 9 summarizes the top-tier performance in some of the datasets, and Table 10 summarizes algorithms with great efficiency.

## 5.2 Which dataset to choose?

We calculated the statistics of the ground truth optical flow in terms of average magnitude and the max magnitude to show the challenging level and large displacement situation of each dataset, as Figure 12 presents. Where the bigger the average magnitude and standard deviation are, the more challenging the dataset is, and the bigger the max magnitude is, the more a robust large displacement handling is required.

We can see clearly that the up-to-date dataset is more challenging than datasets published before. Along with information provided in Table 7, we provide suggestion for datasets with different purpose as follows:

**Comprehensiveness** Taken multiple issues, e.g., categories of ground truth data, image resolution, challenging level, naturalism, scale, popularity, into consideration, we recommended MPI Sintel dataset and Freiburg dataset for their comprehensive property.

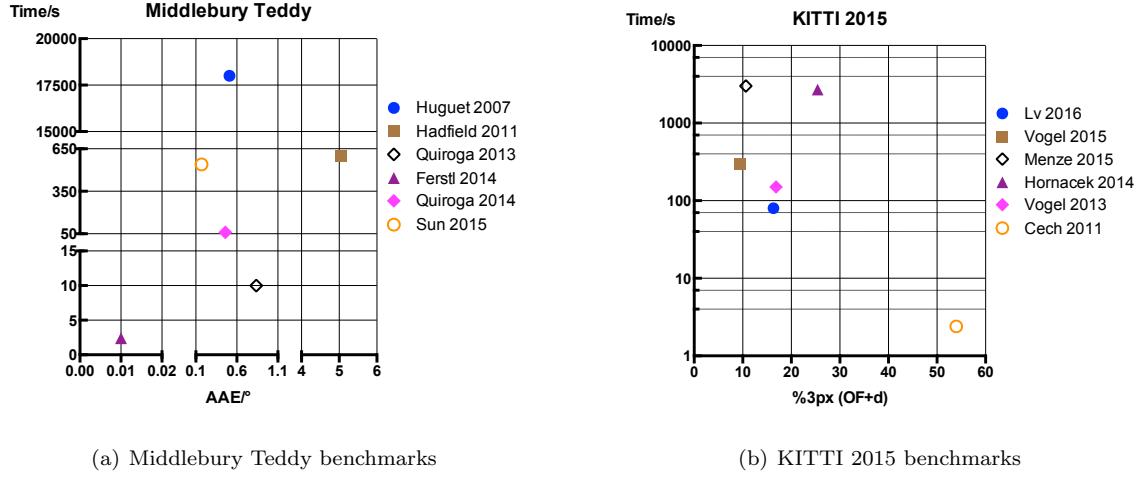


Figure 11: The two-dimensional accuracy-efficiency diagram. A point close to origin point is ideal with less error and less computational time.

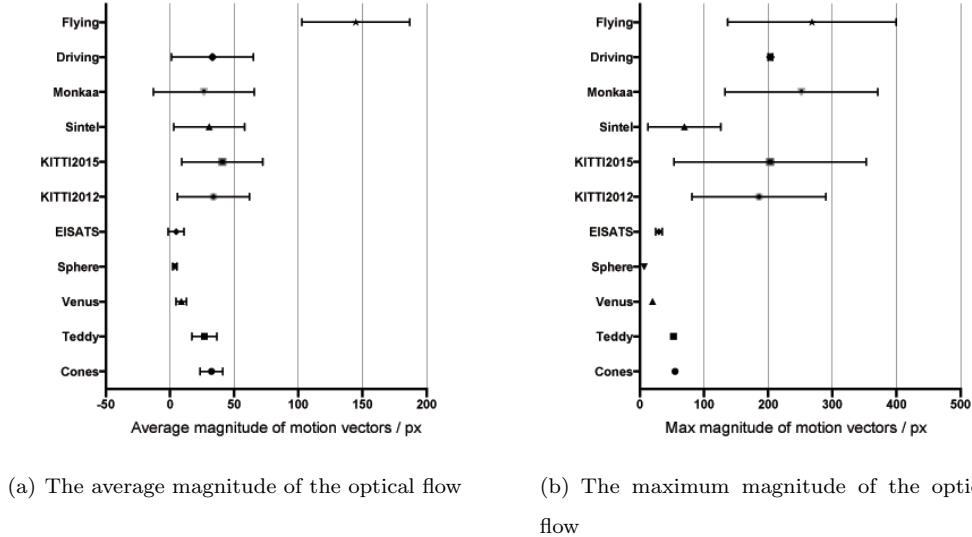


Figure 12: The statistics of the ground truth optical flow magnitude in each dataset. The point annotates the exact value, and the segments on left and right of the point annotates the standard deviation value. (Note: Due to the enormous volume of data, we haven't downloaded the full package of Freiburg dataset. We take the sampler package for analysis which includes three frames in each subset.)

Literature	Category	Dataset	Error	Notes
Year	So/Re/Sc	Error type		
Hornacek [74] 2014	RGB-D/ patch/ feature matching	Teddy Cones OF(RMSE, AAE)	0.35, 0.15 0.54, 0.52	leading performance on Middlebury dataset
Quiroga [59] 2014	RGB-D/ point cloud/ global variational	Teddy Cones OF(RMSE, AAE)	0.49, 0.46 0.45, 0.37	leading performance on Middlebury dataset
Sun [49] 2015	RGB-D/ depth/ pixel assignment	Teddy Cones OF(RMSE, AAE)	0.09, 0.17 0.12, 0.13	the best performance on Middlebury dataset
Vogel [73] 2015	binocular/ patch/ pixel assignment	KITTI2012 OF(EPE , %3px) Disp(EPE , %3px)		the best performance on KITTI dataset
Jaimez [86] 2015	RGB-D/ point cloud/ pixel assignment	Sintel OF(EPE)	1.203	the best performance on Sintel dataset

Table 9: Summary of the representative methods with leading accuracy performance

<b>Literature</b>	<b>Resolution</b>	<b>Time/s</b>	<b>Implement</b>	<b>Description</b>
Gong [5]	$320 \times 240$ (QVGA)	0.082(local) 0.217(global)	GPU	local: winner-take-all(WTA) scheme global: dynamic programming
Rabe [6]	$640 \times 480$ (VGA)	0.040(De6D) 0.100(Va6D)	GPU	De6D: fusing OF and stereo Va6D: coupling OF and stereo
Cech [46]	$640 \times 480$ (VGA)	1.5	CPU(Linux)	seeded growing algorithm complexity $\mathcal{O}(n^2)$
Wedel [7]	$320 \times 240$ (QVGA)	0.200/0.050	CPU/GPU	decoupling OF and stereo variational framework
Hung [47]	$640 \times 480$ (VGA)	28.000/1.400	CPU/OpenMP	voting for temporal constraint anisotropic smoothness
Jaimez [8]	$320 \times 240$ (QVGA)	7.150/0.042	CPU/GPU	primal-dual framework regularization on 3D surface

Table 10: Summary of the representative methods with leading efficiency performance

**Challenging** FlyingThings3D subset of Freiburg dataset shows the characteristic of large displacement, complex occlusion and diverse changes between frames, which is really challenging for scene flow estimation. Monkaa subset shows the similar characteristic which is recommended as well.

**Public popularity** Middlebury, KITTI(2012, 2015) and MPI Sintel dataset provide evaluation protocols and online ranking that can evaluate the performance of a method conveniently. Moreover, the evaluation can be compared with top tier methods in optical flow estimation field and stereo matching field to indicate the superiority of scene flow estimation.

**Multi-view and RGB-D data source** For multi-view stereopsis, Basha rotating sphere and KITTI2012/2015 provide multi-view extension for evaluation. To be noted, Basha provides only point cloud scene flow ground truth for quantitative analysis, while the ground truth of KITTI2012/2015 is sparse. In terms of RGB-D scene flow estimation, MPI Sintel dataset and Basha rotating sphere dataset provides depth ground truth so that disparity-to-depth conversion is no need. In addition, long range of distance may make the depth map transferred from disparity map unclear for visualization, which make image-based algorithm hard to work. The provided depth visualization map is a good option.

**Large scale for learning** Freiburg dataset is currently the only dataset with  $10^5$  order of scale that is designed for training optical flow, disparity, and scene flow.

### 5.3 Which protocol to choose?

Protocols usually varies from each other based on different datasets. The previous datasets like Middlebury and rotating sphere usually use RMSE/NRMSE and AAE for evaluation, while newly-introduced datasets like KITTI, MPI Sintel and Freiburg utilize EPE for evaluation. We recommend researchers to use EPE as a overall protocol, while RMSE and AAE can be supplementary means that reveal error distribution and angular error.

**Should we evaluate the 3D error?** As Equation 18 presents, the disparity in time  $t + 1$  or the disparity change do have influence on scene flow estimation. Hence, we highly recommended that for 2D parametrization scene flow estimation methods, the accuracy of disparity  $t + 1$  or the disparity change should be evaluated and provided. Considering the fact that Equation 19a and 19b is rarely utilized, and rotating sphere of Basha is the only dataset that provide three-dimension point cloud ground truth, we recommend researchers to provide EPE for optical flow, disparity in time  $t$  and time  $t + 1$  as a common protocol for scene flow evaluation.

## 5.4 Future vision

With 17 years of development, scene flow estimation has reached a promising status, while there are still many issues remains to be solved. In this section, we discuss the limitation of existing datasets, and present a brief vision on modification of algorithms.

### 5.4.1 Limitation of current datasets

The survey of existing evaluation methodologies reveals problems and limitations as elaborated below:

**Size** On account that real-time scene flow has been achieved with GPU implementation, high resolution data can be introduced for more challenging work.

**Ground truth** Point cloud is the real three-dimensional parametrization representation for scene flow, which reveals the difference between scene flow and optical flow significantly. Hence, the ground truth for scene flow under the point cloud representation is necessary. Meanwhile, occlusion, textureless region and discontinuity region ground truth are essential for evaluation due to the fact that errors mainly exist near these regions.

**Data source** Current datasets mainly focus on scene flow under the binocular setting, while multi-view extension and specific datasets for RGB-D and light field based scene flow is required. Otherwise, the properties of missing data in RGB-D cameras and the abilities like refocusing of light field cameras will be neglected with current datasets.

**Protocol** A protocol for evaluating performances under different datasets remains vacant. Moreover, the three-dimensional protocol isn't been applied due to the limitation of point cloud ground truth.

### 5.4.2 The modification of methods in the future

By checking the error map provided by KITTI benchmark, it's clear that inaccuracy mainly exists in the boundaries of objects. Since this is a common issue for all computer vision tasks, edge-preserving and reasonable filtering is the first priority.

GPU implementation has shown a great efficiency improvement, and the duality-based optimization has proved to enhance the efficiency of global variational methods without accuracy sacrifice. These kind of methods may be a routine in the future for better efficiency.

With the development of a robust and efficient estimation between two frames, some papers have studied motion estimation under a long sequence [36, 6, 63, 47]. The multi-fames estimation with temporal prior knowledges deserves more attention. A robust temporal constraint can benefit the methods with a better initial value or a better feature [47] to match. The challenges like varying illumination and occlusion can be

handled with the help of it.

The emerging learning based methods and light field technique has brought fresh blood to scene flow estimation. Learning method with CNN shows an upward tendency in the relevant issues of scene flow estimation like stereo matching and optical flow with promising accuracy and computational cost [24, 25, 99]. With the help of the up-to-date large scale training dataset [4], learning-based method has a profound potential to achieve an accurate and fast estimation. Light field camera provides more data than existing data source, which brings diverse possibilities for this field. Similar to the emergence of RGB-D cameras, this new source of data may lead to a new attractive branch.

On account of the fact that scene flow estimation relies highly on texture and intensity information, application will suffer in the night or an insufficient illumination circumstance. Moreover, the car headlights and lighting on the building that are frequent in the autonomous driving scene may interfere motion estimation significantly. Hence, the scene flow estimation with insufficient illumination is worth studying.

## 6 Conclusion

This paper presents a comprehensive and up-to-date survey on both scene flow estimation methods and the evaluation methodologies for the first time after 17 years since scene flow was introduced. We have discussed most of the estimation methods so researchers could have a clear view of this field and get inspired for their studies of interest. The representative methods are highlighted so the differences between these methods are clear, and the similarities between top-tier methods can be seen as a tendency for modification. The widely used benchmarks have been analyzed and compared, so are multiple evaluation protocols. This paper provides sufficient information for researchers to choose the appropriate datasets and protocols for evaluating performance of their algorithms.

There are still ample rooms for future research on accuracy, efficiency and multiple challenges. We wish our work could arise public interest in this field and bring it to a new stage.

## Acknowledgements

This work was supported by projects of National Natural Science Foundation of China [61401113]; and Natural Science Foundation of Heilongjiang Province of China [LC201426].

## References

- [1] S. Vedula, S. Baker, P. Rander, R. Collins, T. Kanade, Three-dimensional scene flow, in: IEEE International Conference on Computer Vision(ICCV), Vol. 2, 1999, pp. 722–729 vol.2.
- [2] B. K. Horn, B. G. Schunck, Determining optical flow, *Artificial intelligence* 17 (1-3) (1981) 185–203.
- [3] B. D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: IJCAI, Vol. 81, 1981, pp. 674–679.
- [4] N. Mayer, E. Ilg, P. Häusser, P. Fischer, D. Cremers, A. Dosovitskiy, T. Brox, A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation, arXiv preprint arXiv:1512.02134.
- [5] M. Gong, Real-time joint disparity and disparity flow estimation on programmable graphics hardware, *Computer Vision and Image Understanding (CVIU)* 113 (1) (2009) 90–100.
- [6] C. Rabe, T. Muller, A. Wedel, U. Franke, Dense, robust, and accurate motion field estimation from stereo image sequences in real-time, in: European Conference on Computer Vision (ECCV), Springer-Verlag, 2010, pp. 582–595.
- [7] A. Wedel, T. Brox, T. Vaudrey, C. Rabe, U. Franke, D. Cremers, Stereoscopic scene flow computation for 3d motion understanding, *International Journal of Computer Vision* 95 (1) (2011) 29–51.
- [8] M. Jaimez, M. Souiai, J. Gonzalez-Jimenez, D. Cremers, A primal-dual framework for real-time dense rgb-d scene flow, in: IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 98–104.
- [9] S. S. Beauchemin, J. L. Barron, The computation of optical flow, *ACM Comput. Surv.* 27 (3) (1995) 433–466.
- [10] M. Otte, H.-H. Nagel, Optical flow estimation: advances and comparisons, in: European Conference on Computer Vision (ECCV), Springer, 1994, pp. 49–60.
- [11] T. Brox, A. Bruhn, N. Papenberg, J. Weickert, High accuracy optical flow estimation based on a theory for warping, in: European conference on computer vision (ECCV), Springer, 2004, pp. 25–36.
- [12] S. Deqing, S. Roth, M. J. Black, Secrets of optical flow estimation and their principles, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 2432–2439.
- [13] S. Baker, D. Scharstein, J. P. Lewis, S. Roth, M. J. Black, R. Szeliski, A database and evaluation methodology for optical flow, *International Journal of Computer Vision* 92 (1) (2011) 1–31.

- [14] Z. Chen, H. Jin, Z. Lin, S. Cohen, Y. Wu, Large displacement optical flow from nearest neighbor fields, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 2443–2450.
- [15] O. U. N. Jith, S. A. Ramakanth, R. V. Babu, Optical flow estimation using approximate nearest neighbor field fusion, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014, pp. 673–6577.
- [16] H. A. Rashwan, M. A. Garca, D. Puig, Variational optical flow estimation based on stick tensor voting, *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society* 22 (7) (2013) 2589–2599.
- [17] L. Xu, J. Jia, Y. Matsushita, Motion detail preserving optical flow estimation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (9) (2012) 1744–1757.
- [18] M. Stoll, S. Volz, A. Bruhn, Adaptive integration of feature matches into variational optical flow methods, in: Asian Conference on Computer Vision (ACCV), 2013, pp. 1–14.
- [19] M. Stoll, S. Volz, A. Bruhn, Joint trilateral filtering for multiframe optical flow, in: IEEE International Conference on Image Processing (ICIP), 2013, pp. 3845–3849.
- [20] W. Li, D. Cosker, M. Brown, R. Tang, Optical flow estimation using laplacian mesh energy, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2013, pp. 2435–2442.
- [21] H. Hirschmuller, Stereo processing by semiglobal matching and mutual information, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 30 (2) (2008) 328–341.
- [22] D. Scharstein, R. Szeliski, R. Zabih, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, in: IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV), 2001, pp. 131–140.
- [23] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3354–3361.
- [24] J. bontar, Y. LeCun, Computing the stereo matching cost with a convolutional neural network, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1592–1599.
- [25] W. Luo, A. G. Schwing, R. Urtasun, Efficient deep learning for stereo matching, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 5695–5703.
- [26] C. Zhang, Z. Li, Y. Cheng, R. Cai, H. Chao, Y. Rui, Meshstereo: A global stereo model with mesh alignment regularization for view interpolation, in: IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2057–2065.

- [27] K. Yamaguchi, D. McAllester, R. Urtasun, Efficient joint segmentation, occlusion labeling, stereo and flow estimation, in: European Conference on Computer Vision (ECCV), 2014, pp. 756–771.
- [28] A. Zanfir, C. Sminchisescu, Large displacement 3d scene flow with occlusion reasoning, in: IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4417–4425.
- [29] H. A. Alhaija, A. Sellent, D. Kondermann, C. Rother, Graphflow–6d large displacement scene flow via graph matching, in: German Conference on Pattern Recognition, Springer, 2015, pp. 285–296.
- [30] T. Schuchert, T. Aach, H. Scharr, Range flow in varying illumination: Algorithms and comparisons, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32 (9) (2010) 1646–1658.
- [31] P. F. U. Gotardo, T. Simon, Y. Sheikh, I. Matthews, Photogeometric scene flow for high-detail dynamic 3d reconstruction, in: IEEE International Conference on Computer Vision (ICCV), 2015, pp. 846–854.
- [32] T. Popham, A. Bhalerao, R. Wilson, Estimating scene flow using an interconnected patch surface model with belief-propagation inference, *Computer Vision and Image Understanding (CVIU)* 121 (2014) 74–85.
- [33] Z. Ye, C. Kambhamettu, On 3-d scene flow and structure recovery from multiview image sequences, *IEEE Transactions on Systems, Man, and Cybernetics* 33 (4) (2003) 592–606.
- [34] F. Liu, V. Philomin, Disparity estimation in stereo sequences using scene flow, in: British Machine Vision Conference (BMVC), Vol. 1, 2009, p. 2.
- [35] S. Ghuffar, N. Brosch, N. Pfeifer, M. Gelautz, Motion estimation and segmentation in depth and intensity videos, *Integrated Computer-Aided Engineering* 21 (3) (2014) 203–218.
- [36] U. Franke, C. Rabe, H. Badino, S. Gehrig, 6d-vision: Fusion of stereo and motion for robust environment perception, in: Joint Pattern Recognition Symposium, Springer, 2005, pp. 216–223.
- [37] P. F. Alcantarilla, J. J. Yebes, J. Almaz, x00E, L. M. Bergasa, On combining visual slam and dense scene flow to increase the robustness of localization and mapping in dynamic environments, in: IEEE International Conference on Robotics and Automation (ICRA), 2012, pp. 1290–1297.
- [38] E. Herbst, X. Ren, D. Fox, Rgb-d flow: Dense 3-d motion estimation using color and depth, in: IEEE International Conference on Robotics and Automation (ICRA), 2013, pp. 2276–2282.
- [39] M. Menze, A. Geiger, Object scene flow for autonomous vehicles, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3061–3070.
- [40] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2008, pp. 1–8.

- [41] N. Dalal, B. Triggs, C. Schmid, Human detection using oriented histograms of flow and appearance, in: European Conference on Computer Vision (ECCV), 2006, pp. 428–441.
- [42] F. Yasutaka, J. Ponce, Dense 3d motion capture for human faces, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 1674–1681.
- [43] M. Isard, J. MacCormick, Dense motion and disparity estimation via loopy belief propagation, in: Asian conference on computer vision, Springer, 2006, pp. 32–41.
- [44] F. Huguet, F. Devernay, A variational method for scene flow estimation from stereo sequences, in: IEEE International Conference on Computer Vision (ICCV), 2007, pp. 1–7.
- [45] A. Wedel, C. Rabe, T. Vaudrey, T. Brox, U. Franke, D. Cremers, Efficient dense scene flow from sparse or dense stereo data, in: European Conference on Computer Vision (ECCV), Vol. 5302 LNCS, 2008, pp. 739–751.
- [46] J. Cech, J. Sanchez-Riera, R. Horaud, Scene flow estimation by growing correspondence seeds, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 3129–3136.
- [47] C. H. Hung, L. Xu, J. Y. Jia, Consistent binocular depth and scene flow with chained temporal profiles, International Journal of Computer Vision 102 (1-3) (2013) 271–292.
- [48] P. P. Srinivasan, M. W. Tao, R. Ng, R. Ramamoorthi, Oriented light-field windows for scene flow, in: IEEE International Conference on Computer Vision (ICCV), 2015, pp. 3496–3504.
- [49] D. Sun, E. B. Sudderth, H. Pfister, Layered rgbd scene flow estimation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2015, pp. 548–556.
- [50] Y. Zhang, C. Kambhamettu, Integrated 3d scene flow and structure recovery from multiview image sequences, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 2, 2000, pp. 674–681.
- [51] Z. Ye, C. Kambhamettu, On 3d scene flow and structure estimation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 2, 2001, pp. 778–785.
- [52] R. Li, S. Sclaroff, Multi-scale 3d scene flow from binocular stereo sequences, in: IEEE Workshops on Application of Computer Vision (WACV), Vol. 2, 2005, pp. 147–153.
- [53] J. Ruttle, M. Manzke, R. Dahyot, Estimating 3d scene flow from multiple 2d optical flows, in: International Machine Vision and Image Processing Conference (IMVIP), 2009, pp. 1–6.
- [54] T. Basha, Y. Moses, N. Kiryati, Multi-view scene flow estimation: A view centered variational approach, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 1506–1513.

- [55] S. Hadfield, R. Bowden, Kinecting the dots: Particle based scene flow from depth sensors, in: IEEE International Conference on Computer Vision (ICCV), 2011, pp. 2290–2295.
- [56] J. Quiroga, F. Devernay, J. Crowley, Local/global scene flow estimation, in: IEEE International Conference on Image Processing (ICIP), 2013, pp. 3850 – 3854.
- [57] D. Ferstl, C. Reinbacher, G. Riegler, R. M. x00Fc, ther, H. Bischof, atgv-sf: Dense variational scene flow through projective warping and higher order regularization, in: International Conference on 3D Vision, Vol. 1, 2014, pp. 285–292.
- [58] D. Ferstl, G. Riegler, M. Ruether, H. Bischof, Cp-census: A novel model for dense variational scene flow from rgbd data, in: British Machine Vision Conference, 2014.
- [59] J. Quiroga, T. Brox, F. Devernay, J. Crowley, Dense semi-rigid scene flow estimation from rgbd images, in: European Conference on Computer Vision (ECCV), Vol. 8695 LNCS, 2014, pp. 567–582.
- [60] C. Richardt, H. Kim, L. Valgaerts, C. Theobalt, Dense wide-baseline scene flow from two handheld video cameras, arXiv preprint arXiv:1609.05115.
- [61] J. Neumann, Y. Aloimonos, Spatio-temporal stereo using multi-resolution subdivision surfaces, in: Stereo and Multi-Baseline Vision (SMBV), 2001, pp. 103–108.
- [62] J. Neumann, Y. Aloimonos, Spatio-temporal stereo using multi-resolution subdivision surfaces, International Journal of Computer Vision 47 (1) (2002) 181–193.
- [63] F. Yasutaka, J. Ponce, Dense 3d motion capture from synchronized video streams, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008, pp. 1–8.
- [64] J. Courchay, J.-P. Pons, P. Monasse, R. Keriven, Dense and accurate spatio-temporal multi-view stereovision, in: Asian Conference on Computer Vision (ACCV), 2010, pp. 11–22.
- [65] N. Birkbeck, D. Cobza, M. Jagersand, Basis constrained 3d scene flow on a dynamic proxy, in: IEEE International Conference on Computer Vision (ICCV), 2011, pp. 1967–1974.
- [66] A. Letouzey, B. Petit, E. Boyer, Scene flow from depth and color images, in: British Machine Vision Conference, BMVA Press, 2011, pp. 46:1–11.
- [67] J. Park, T. H. Oh, J. Jung, Y.-W. Tai, I. S. Kweon, A tensor voting approach for multi-view 3d scene flow estimation and refinement, in: European Conference on Computer Vision (ECCV), Vol. 7575 LNCS, 2012, pp. 288–302.

- [68] R. L. Carceroni, K. N. Kutalakos, Multi-view scene capture by surfel sampling: from video streams to non-rigid 3d motion, shape and reflectance, in: IEEE International Conference on Computer Vision (ICCV), Vol. 2, 2001, pp. 60–67 vol.2.
- [69] F. Devernay, D. Mateus, M. Guilbert, Multi-camera scene flow by tracking 3-d points and surfels, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 2, 2006, pp. 2203–2212.
- [70] T. Popham, A. Bhalerao, R. Wilson, Multi-frame scene-flow estimation using a patch model and smooth motion prior, in: British Machine Vision Conference (BMVC), 2010.
- [71] C. Vogel, K. Schindler, S. Roth, Piecewise rigid scene flow, in: IEEE International Conference on Computer Vision (ICCV), 2013, pp. 1377–1384.
- [72] C. Vogel, K. Schindler, S. Roth, View-consistent 3d scene flow estimation over multiple frames, in: European Conference on Computer Vision (ECCV), Vol. 8692, 2014, pp. 263–278.
- [73] C. Vogel, K. Schindler, S. Roth, 3d scene flow estimation with a piecewise rigid scene model, International Journal of Computer Vision 115 (1) (2015) 1–28.
- [74] M. Hornecek, A. Fitzgibbon, C. Rother, Sphereflo: 6 dof scene flow from rgb-d pairs, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 3526–3533.
- [75] J. P. Pons, R. Keriven, O. Faugeras, G. Hermosillo, Variational stereovision and 3d scene flow estimation with statistical similarity measures, in: IEEE International Conference on Computer Vision (ICCV), 2003, pp. 597–602 vol.1.
- [76] J. P. Pons, R. Keriven, O. Faugeras, Multi-view stereo reconstruction and scene flow estimation with a global image-based matching score, International Journal of Computer Vision 72 (2) (2007) 179–193.
- [77] M. Sizintsev, R. P. Wildes, Spatiotemporal stereo and scene flow via stequel matching, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (6) (2012) 1206–1219.
- [78] C. Vogel, K. Schindler, S. Roth, 3d scene flow estimation with a rigid motion prior, in: IEEE International Conference on Computer Vision (ICCV), 2011, pp. 1291–1298.
- [79] L. Valgaerts, A. Bruhn, H. Zimmer, J. Weickert, C. Stoll, C. Theobalt, Joint estimation of motion, structure and geometry from stereo sequences, in: European Conference on Computer Vision (ECCV), Vol. 6314 LNCS, 2010, pp. 568–581.
- [80] E. Tola, V. Lepetit, P. Fua, Daisy: An efficient dense descriptor applied to wide-baseline stereo, IEEE transactions on pattern analysis and machine intelligence 32 (5) (2010) 815–830.

- [81] H. Spies, B. Jähne, J. L. Barron, Dense range flow from depth and intensity data, in: International Conference on Pattern Recognition (ICPR), Vol. 1, 2000, pp. 131–134 vol.1.
- [82] H. Spies, B. Jähne, J. L. Barron, Range flow estimation, Computer Vision and Image Understanding (CVIU) 85 (3) (2002) 209–231.
- [83] T. C. Lukins, R. B. Fisher, Colour constrained 4d flow, in: British Machine Vision Conference (BMVC), 2005, pp. 340–348.
- [84] J.-M. Gottfried, J. Fehr, C. S. Garbe, Computing range flow from multi-modal kinect data, in: International Symposium on Visual Computing, Springer, 2011, pp. 758–767.
- [85] X. Zhang, D. Chen, Z. Yuan, N. Zheng, Dense scene flow based on depth and multi-channel bilateral filter, in: X. Zhang, D. Chen, Z. Yuan, Z. Hu (Eds.), Asian Conference on Computer Vision (ACCV), 2012, pp. 140–151.
- [86] M. Jaimez, M. Souiai, J. St, x00Fc, ckler, J. Gonzalez-Jimenez, D. Cremers, Motion cooperation: Smooth piece-wise rigid scene flow from rgb-d images, in: International Conference on 3D Vision (3DV), 2015, pp. 64–72.
- [87] S. Hadfield, R. Bowden, Scene particles: Unregularized particle-based scene flow estimation, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (3) (2014) 564–576.
- [88] E. H. Adelson, J. Y. A. Wang, Single lens stereo with a plenoptic camera, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (2) (1992) 99–106.
- [89] M. W. Tao, S. Hadap, J. Malik, R. Ramamoorthi, Depth from combining defocus and correspondence using light-field cameras, in: IEEE International Conference on Computer Vision (ICCV), 2013, pp. 673–680.
- [90] C. Zach, T. Pock, H. Bischof, A duality based approach for realtime tv-l1 optical flow, in: Joint Pattern Recognition Symposium, 2007, pp. 214–223.
- [91] K. Bredies, K. Kunisch, T. Pock, Total generalized variation, SIAM Journal on Imaging Sciences 3 (3) (2010) 492–526.
- [92] O. Veksler, Y. Boykov, P. Mehrani, Superpixels and supervoxels in an energy optimization framework, in: European conference on Computer vision (ECCV), Springer, 2010, pp. 211–224.
- [93] V. Lempitsky, S. Roth, C. Rother, Fusionflow: Discrete-continuous optimization for optical flow estimation, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, 2008, pp. 1–8.

- [94] C. Rother, V. Kolmogorov, V. Lempitsky, M. Szummer, Optimizing binary mrf's via extended roof duality, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8.
- [95] Z. Lv, C. Beall, P. F. Alcantarilla, F. Li, Z. Kira, F. Dellaert, A continuous optimization approach for efficient and accurate scene flow, in: European Conference on Computer Vision, Springer, 2016, pp. 757–773.
- [96] F. R. Kschischang, B. J. Frey, H.-A. Loeliger, Factor graphs and the sum-product algorithm, *IEEE Transactions on information theory* 47 (2) (2001) 498–519.
- [97] S. Hadfield, R. Bowden, Scene flow estimation using intelligent cost functions, in: British Machine Vision Conference (BMVC), 2014.
- [98] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, Backpropagation applied to handwritten zip code recognition, *Neural Computation* 1 (4) (1989) 541–551.
- [99] A. Dosovitskiy, P. Fischer, E. Ilg, H. P. xE, usser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, T. Brox, Flownet: Learning optical flow with convolutional networks, in: IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2758–2766.
- [100] D. J. Fleet, A. D. Jepson, Computation of component image velocity from local phase information, *International Journal of Computer Vision* 5 (1) (1990) 77–104.
- [101] D. J. Butler, J. Wulff, G. B. Stanley, M. J. Black, A naturalistic open source movie for optical flow evaluation, in: European Conference on Computer Vision (ECCV), 2012, pp. 611–625.
- [102] D. Scharstein, R. Szeliski, R. Zabih, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, in: IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV), 2001, pp. 131–140.
- [103] D. Scharstein, R. Szeliski, High-accuracy stereo depth maps using structured light, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1, 2003, pp. 195–202.
- [104] T. Vaudrey, C. Rabe, R. Klette, J. Milburn, Differences between stereo and motion behavior on synthetic and real-world stereo sequences, in: International Conference of Image and Vision Computing New Zealand (IVCNZ), 2008, pp. 1–6.
- [105] R. Li, S. Sclaroff, Multi-scale 3d scene flow from binocular stereo sequences, *Computer Vision and Image Understanding (CVIU)* 110 (1) (2008) 75–90.
- [106] T. Basha, Y. Moses, N. Kiryati, Multi-view scene flow estimation: A view centered variational approach, *International Journal of Computer Vision* 101 (1) (2013) 6–21.

- [107] Blender. <https://www.blender.org/>
- [108] J. Gall, C. Stoll, E. d. Aguiar, C. Theobalt, B. Rosenhahn, H. P. Seidel, Motion capture using joint skeleton tracking and surface estimation, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 1746–1753.