Robust Stereo Matching with Surface Normal Prediction

Shuangli Zhang†, Weijian Xie†, Guofeng Zhang*†, Hujun Bao† and Michael Kaess‡

***Abstract*— Traditional stereo matching approaches generally have problems in handling textureless regions, strong occlusions and reflective regions that do not satisfy a Lambertian surface assumption. In this paper, we propose to combine the predicted surface normal by deep learning to overcome these inherent difficulties in stereo matching. With the selected reliable disparities from stereo matching method and effective edge fusion strategy, we can faithfully convert the predicted surface normal map to a disparity map by solving a least squares system which maintains discontinuity on object boundaries and continuity on other regions. Then we refine the disparity map iteratively by bilateral filtering-based completion and edge feature refinement. Experimental results on the Middlebury dataset and our own captured stereo sequences demonstrate the effectiveness of the proposed approach.**

I. INTRODUCTION

Three dimension range sensing is important for robotics since it can help robots to navigate and even understand their environments. A simple way to estimate depth information is to use stereo cameras. The last two decades have witnessed various research works, and significant progress has been made on stereo matching. However, limited by the low-level features these methods rely on, almost all prior methods have difficulties in handling challenging scenarios such as regions with strong occlusion, reflection or insufficient texture.

In the past few years, deep learning and convolutional neural networks (CNNs) have achieved great successes on many computer vision problems. Various tasks, like image classification and object detection, now perform significantly better than ever. Very recently, Zbontar and LeCun [39] proposed to use a CNN to compute the stereo matching cost and get very decent results. However, this method still does not address the inherent difficulties in stereo matching mentioned above.

In this paper, we propose to use the predicted surface normal by CNN to improve stereo matching. Single image-based surface normal or depth prediction work [40], [34], [8], [7], [1] have achieved great success in recent years. These methods do not involve matching and thus do not have the limitations for handling regions with strong occlusion, reflection or insufficient textures. However, since the predicted surface normals are not always reliable and have ambiguity in determining the depth, we propose to use the reliable disparity estimate from stereo matching and use the fused edge information to convert the surface normal map to the disparity map by solving a linear system. To achieve this goal, we propose a novel plane-based disparity confidence measurement to select reliable disparity pixels. We also provide an edge fusion algorithm locating edges that indicate potential disparity discontinuities. Finally, we refine the disparity map iteratively by bilateral filtering-based completion and edge feature refinement. We evaluate our method on both indoor stereo data captured by ourselves and the public Middlebury stereo dataset [29], [30], [18], [28]. The experimental results demonstrate that the proposed approach is able to address the matching ambiguities in textureless and occluded regions, and improve the stereo matching result.

*Corresponding author.

†Shuangli Zhang, Weijian Xie, Guofeng Zhang, and Hujun Bao are with the State Key Lab of CAD&CG, Zhejiang University. Emails: {zhangguofeng, bao}@cad.zju.edu.cn.

‡Michael Kaess is with the Robotics Institute, Carnegie Mellon University. Email: kaess@cmu.edu.

II. RELATED WORK

Binocular stereo matching can be used to recover disparity/depth maps from stereo image pairs, which are categorized into local and global methods [29]. Disparity is predicted by measuring the dissimilarity of pixels or local patches and choosing the pairs with least local or global matching cost. Local methods compute the disparity that only counts the matching with local aggregation. These algorithms [21], [19], [13], [38] generally compute the disparity for each pixel independently, so the recovered disparity maps may easily have artifacts around textureless and occluded regions. Global methods generally formulate stereo matching as an optimization problem by involving smoothness constraints for neighboring pixels. Graph Cuts [4] or Belief Propagation [10] are usually used to minimize the energy function. Semi-Global Matching (SGM) [17] both considers pixelwise matching and global smoothness constraints. Since SGM is much more efficient than other methods with similar accuracy, it is widely used for real-time stereo matching [9][12].

Many methods have been proposed to reduce the matching ambiguity for textureless regions. One representative type of these work are segmentation-based methods, which generally use an image segmentation method to cluster the neighboring pixels with similar colors into the same segment, and then fit the 3D surface for each segment as a 3D plane [23], [33], a B-spline or a quadratic surface [3], [41]. Another representative type is to use color-aware filtering techniques in a larger window [36] or even the whole image [37] to adaptively aggregate the matching cost based on color similarity.

Since outliers are inevitable in practice, several methods have been proposed to compute the confidence for the estimated disparity map. An evaluation of confidence measures of disparity can be found in [20]. The confidence prediction is often followed by a Markov Random Field (MRF) or

Conditional Random Field (CRF) framework, where disparity smoothness constraints only exist between neighbor pixels, and used as soft constraints in a global optimization framework to reduce the disparity ambiguity [32].

Recently, learning methods also have been used to assist depth estimation in stereo. Zbontar and LeCun [39] proposed a new method to compute matching cost in stereo with a CNN. Following the classical depth estimation way by comparing small image patches, they proposed a new measurement of patch similarity via training a CNN. There are also several methods proposed to improve the stereo matching results with the information provided by deep learning methods. For instance, Hane et al. [15] proposed to use the response of a trained surface normal classifier as a regularization term when adding it into the energy minimization of a global disparity labeling problem. Guney et al. [14] proposed to make use of semantic segmentation and object knowledge, especially regular structures of cars, to constrain the disparity and attain good performance for textureless, saturated or semitransparent surfaces.

The last three years have witnessed significant progress in surface normal prediction from a single image. Zeisl et al. [40] proposed to build a regressor to train a surface normal classifier while combining both context-based and segment-based features. Their method is applicable to both indoor and outdoor environments. Wang et al. [34] train a CNN built upon the understanding and constraints of 3D scenes and use meaningful intermediate representations to achieve good performance. Eigen et al. [8] proposed to estimate surface normal, depth and semantic segmentation by a single multiscale convolutional network architecture. Most recently, Bansal et al. [1] proposed a novel surface normal estimation method using a new skip-network architecture. This method achieves state-of-the-art accuracy and can faithfully recover fine surface details for indoor images. We use this method to predict the surface normal in our stereo matching framework.

III. Approach Overview

Figure 1 gives an overview of our approach. Given a stereo pair, we first use the stereo matching method of [39] or [17] to estimate the disparity map D for the left image. Given disparity d, the depth can be computed by $z = bf/d$ (b is the stereo baseline and f is the focal length.) For simplicity, the terms "depth" and "disparity" are used interchangeably in our paper. Then we compute the disparity confidence by taking into account Left Right Difference (LRD) [20] and the disparity distance to the fitted local planes. We also use the normal prediction method proposed in [1] to recover the normal maps for the left image. After recovering the normal maps, we first use mean shift segmentation method [5] to segment the whole image into a set of segments. Then we combine edge features from the left image, its disparity map and segmentation to maintain effective and continuous edge features on discontinuous regions. Finally, we convert the normal to disparity with reliable disparities from stereo matching and fused edge information by solving a linear system.

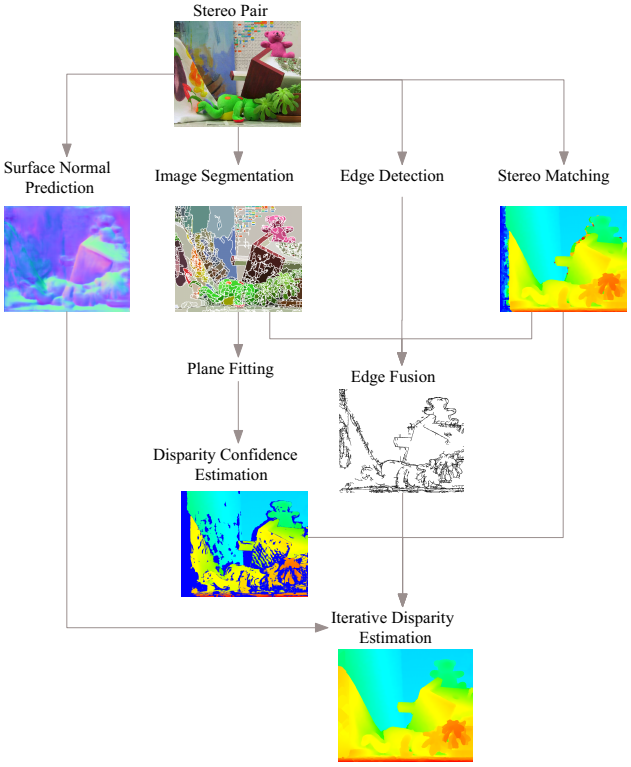


Fig. 1. Framework overview.

IV. Disparity Confidence Generation and Edge Fusion

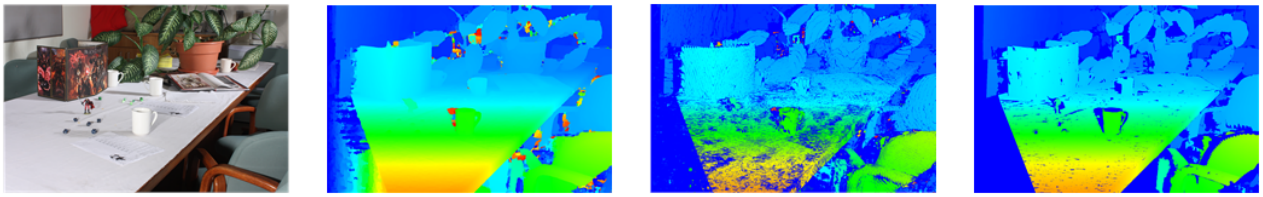
Given a surface normal map, we can use the integral method as in [35], [26] to convert it to a depth map. However, there will be depth ambiguity if there are no depth priors provided. Therefore, we propose to use the recovered disparity information from stereo matching to help convert surface normal to disparity. For robust conversion, we need to measure the disparity confidence to exclude outliers first.

A. Depth Confidence

We employ LRD measurement proposed in [20] to estimate the initial depth confidence. If the difference between the smallest and the second smallest matching cost is large, the confidence measured by LRD should be large. Meanwhile, the consistency of least matching costs is checked across the left and right images implicitly to impose the assumption that a reliable depth confidence should have consistent matching cost. The matching cost maps of left and right images are denoted as c^l and c^r, respectively. For the pixel $\mathbf{x} = (x, y)^\top$ in the left image, its first and second smallest matching costs are defined as $c_1^l(x, y)$ and $c_2^l(x, y)$, respectively. Then we define the LRD-based depth confidence as

$$C_{LRD}(x, y) = \min\left(1, \log_{10}\left(\frac{c_2^l(x, y) - c_1^l(x, y)}{|c_1^l - \min_{d_r} c^r(x - d_1, y, d_r)|} + 1\right)\right). \tag{1}$$

We select the pixels that satisfy $C_{LRD}(x, y) > \tau_1$ as initial reliable pixels, as illustrated in Figure 2(c). τ_1 is an adaptive threshold, which is set to 1 as default, and will be



(a) Left image (b) Original disparity (c) Disparity selected by C_{LRD} (d) Disparity selected by C_P

Fig. 2. Disparity confidence measurements.

automatically increased or decreased to make the ratio of reliable pixels to fall within the range of $[40\%, 60\%]$. This filtering can effectively select sufficient accurate disparity pixels and remove most outliers. Then we propose to further remove outliers by plane fitting.

By assuming the estimated disparities are locally smooth, it is reasonable to compute its deviation with its local surface to measure the disparity error. We use the mean shift segmentation method [5] to segment the image. For more robust segmentation, we first make a preprocessing to adjust the image brightness. The image intensity is firstly normalized to [0, 1], and the median intensity of all pixels is denoted as I_m. We lighten the whole image if $I_m < 0.5$ and darken the whole image if $I_m > 0.5$. The image intensity is scaled as

$$I' = \begin{cases} I^{1+(I_m-0.5)} & \text{if } I_m < 0.5, \\ I^{1+3(I_m-0.5)} & \text{if } I_m > 0.5. \end{cases} \quad (2)$$

The above preprocessing can make image brightness more balence for better segmentation.

For each segment, we use the RANSAC algorithm [11] to perform plane fitting. However, segmentation may be imperfect, and a segment may contain multiple objects. So we first use the RANSAC method [11] to fit a plane with most inliers. Then we remove the inliers from the segment and use the remaining pixels to fit another best plane with most inliers. The above procedure is repeated for several times. So for each segment S_i, we may obtain multiple 3D planes. Let vector $\mathbf{P}_j$ denote the jth plane of this segmentation, where $\{\mathbf{P}_j = (a_j, b_j, c_j, d_j)^\top | j = 1, 2, 3, ...\}$. The related plane equation is $a_j x + b_j y + c_j z + d_j = 0$.

For each segment S where at least one plane is successfully fitted, we can measure the disparity distance of inside pixel (x, y) and its original disparity z obtained by any stereo matching algorithm to all fitted planes and select the lowest distance $\epsilon(x, y)$ by

$$\epsilon(x, y) = \min_{(a_i, b_i, c_i, d_i)} \frac{|(a_i x + b_i y + c_i z + d_i)|}{\sqrt{{a_i}^2 + {b_i}^2 + {c_i}^2}}. \quad (3)$$

There is a negative correlation between $\epsilon(x, y)$ and its confidence. If the disparity fits some plane well, $\epsilon(x, y)$ is a small value, and its confidence will be high.

Specifically, if pixel (x, y) is considered as an outlier in all the planes fitted, then it is considered as noise, and $\epsilon(x, y)$ is assigned with INT_MAX.

If a segment does not have sufficient reliable disparities to fit a 3D plane, we compute its bounding box, and find other segmentation regions within the bounding box to help address the confidences of this segmentation. After collecting all the pixels (x, y) in the bounding box with $\epsilon(x, y)$ less than INT_MAX, we select the minimal and maximal disparities $(d_{\min}, d_{\max})$ from them to compute the residual error. $\epsilon(x, y)$ here is defined as

$$\epsilon(x, y) = \begin{cases} \alpha |D(x, y) - d_{\min}| + \rho & \text{if } D(x, y) < d_{\min}, \\ \alpha |D(x, y) - d_{\max}| + \rho & \text{if } D(x, y) > d_{\max}, \\ \rho & \text{otherwise.} \end{cases} \quad (4)$$

where $D(x, y)$ is the disparity of pixel (x, y), α is a weight and set to 2 in our experiment. As the bounding box strategy above only provides rough borderlines of disparity, we add a small value ρ (set to 1 in our experiments) to $\epsilon(x, y)$ to decrease the confidence of this kind of segmentation. Then, our segment based confidence is defined as

$$C_P(x, y) = \exp(-\epsilon(x, y)). \quad (5)$$

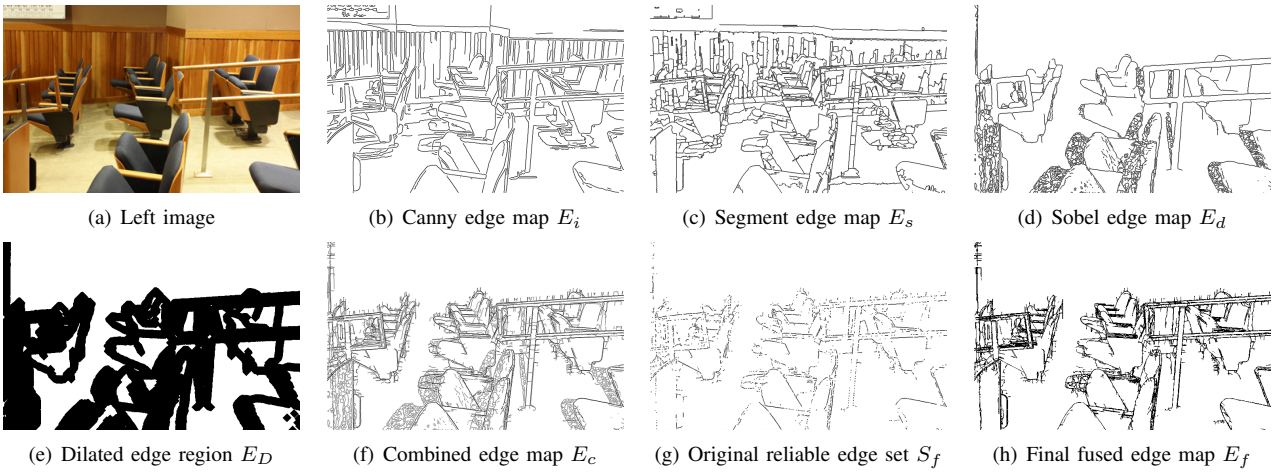
The filtered disparity map by C_p is shown in Figure 2(d).

By combining the LRD-based and fitting-based confidences, we can select the reliable disparities by satisfying $C_{LRD}(x, y) > \tau_1$ and $C_P(x, y) > \tau_2$, where τ_1 and τ_2 are two thresholds. Either a bigger τ_1 or τ_2 would cause a sparser result. Both thresholds should be determined according to the performance of stereo matching algorithm that provide original disparity map and the character of the scenario. A high τ_1 is suitable for a noisy disparity map to prevent outliers, while a low τ_1 is better for a good disparity result since it could output denser reliable pixels. And a high τ_2 is suitable for a scene with a lot of planes, while a low τ_2 could tolerate more curve surfaces.

B. Edge Extraction and Fusion

The normal constraints should not be imposed on discontinuous boundaries. However, the extracted edges do not always reflect the discontinuous boundaries. We propose to fuse the edge information generated by three different methods (i.e. Canny image edges, segment boundaries and disparity edges). An edge denoising process is applied and edge tracking is used to enhance edge connectivity.

First, we use Canny edge detector to extract edges from RGB images, denoted as E_i. With the obtained segments by mean shift, we also can directly get the edges according to the segment boundaries, denoted as E_s. The disparity edge map E_d is computed by applying the Sobel operator to the



(a) Left image (b) Canny edge map E_i (c) Segment edge map E_s (d) Sobel edge map E_d

(e) Dilated edge region E_D (f) Combined edge map E_c (g) Original reliable edge set S_f (h) Final fused edge map E_f

Fig. 3. Edge fusion.

estimated disparity map (we use combination of Sobel and Laplace operators in iterative process of depth refinement).

Second, we only maintain edges which probably imply true disparity discontinuity. The method of separating effective edges from those in smooth regions is inspired by [22]. We dilate E_d to get E_D, which covers all potential depth discontinuity regions. Then we fuse these edge maps as

$$E_c = E_i \cap E_D + E_s \cap E_D + E_d. \tag{6}$$

If a pixel is marked as edge pixel in two or more edge maps above, we consider it as a reliable edge pixel. Let S_f denote the set of reliable edges. All pixels with $E_c > 1$ are added to the set.

Then an edge tracking is applied to connect reliable edges. We compute gradient direction of every potential edge pixel $p(x, y)$ with $E_c(x, y) > 0$. Then for every reliable pixel (x_1, y_1) in S_f, it is assigned with a reliability level r. In the beginning, let $r(x, y) = 0$ for all the pixels in S_f. Also, set $\vec{n}$ to be a unit vector vertical to its gradient direction. We then search potential edge pixels within a neighborhood search region. Pixel (x_2, y_2) will be added to S_f, if $E_f(x_2, y_2) > 0$ and

$$\frac{|(x_2 - x_1, y_2 - y_1) \times \vec{n}|}{|(x_2 - x_1, y_2 - y_1)|} < \sin\frac{\pi}{6}. \tag{7}$$

The reliability level $r(x_2, y_2) = r(x_1, y_1) + 1$, which means (x_2, y_2) is less reliable than (x_1, y_1).

The search stops when there are no pixels in S_f left with reliability level less than 5 in our experiments. A bigger threshold will result in more redundant edges while a smaller value may miss important discontinuity information. The neighborhood size is set to 7×7 in our experiments. Then edge map E_f is extracted from S_f.

V. Disparity Conversion from Surface Normal

We would like to convert the surface normal to disparity. The 3D point cloud extracted from the disparity map and camera intrinsics is not with homogeneous distribution on x and y, i.e. any depth refinement in the real space would cause changes on all three coordinates. For convenience, we propose to transform normal space to disparity space. The axes of disparity space are x- and y-coordinate of the image pixel and its disparity, respectively.

The transformation of points from real world coordinate (X, Y, Z) to disparity space coordinate (x, y, z) is

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} f\frac{X}{Z} + c_x \\ f\frac{Y}{Z} + c_y \\ \frac{bf}{Z} \end{bmatrix}, \tag{8}$$

where f is the focal length, b is the baseline, and (c_x, c_y) is the principal point. The related normal transformation ([31], [15]) is given by

$$\begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix} = \begin{bmatrix} \frac{Z}{f} & 0 & 0 \\ 0 & \frac{Z}{f} & 0 \\ -\frac{XZ}{bf} & -\frac{YZ}{bf} & -\frac{Z^2}{bf} \end{bmatrix} \begin{bmatrix} n_X \\ n_Y \\ n_Z \end{bmatrix}. \tag{9}$$

Without loss of generality, each grid in disparity space (i.e. four neighboring pixels in the image) contains four points $\{\mathbf{p}_i = [x_i, y_i, z_i]^\top | i = 1, 2, 3, 4\}$. Here, (x_i, y_i) are the fixed pixel coordinates, and only z_i needs to be determined. Their corresponding normals are denoted as $\{\mathbf{n}_i = [n_{x_i}, n_{y_i}, n_{z_i}]^\top | i = 1, 2, 3, 4.\}$. The normal of $\mathbf{n}_i$ should be orthogonal to the vector $\mathbf{l}_{i,j} = \mathbf{p}_j - \mathbf{p}_i$. So we have the following four equations:

$$\begin{aligned} \mathbf{n}_1^\top \mathbf{l}_{1,2} &= 0, \\ \mathbf{n}_1^\top \mathbf{l}_{1,3} &= 0, \\ \mathbf{n}_2^\top \mathbf{l}_{2,4} &= 0, \\ \mathbf{n}_4^\top \mathbf{l}_{3,4} &= 0. \end{aligned}$$

And for normal $[n_{x_i}, n_{y_i}, n_{z_i}]^T$ and horizontal vector $[1, 0, z_{i+1} - z_i]^T$, we have

$$z_{i+1} - z_i = \frac{n_{x_i}}{n_{z_i}}. \tag{10}$$

Similarly, for normal and vertical vector $[0, 1, z_{i+w} - z_i]^T$, where w is image width, we have

$$z_{i+w} - z_i = \frac{n_{y_i}}{n_{z_i}}. \tag{11}$$

For each grid, we have the above equations. So for a whole image with M pixels and homogeneous normal constraints, we can construct a linear system $\mathbf{Az} = \mathbf{B}$, where $\mathbf{A}$ is a $2M \times M$ sparse matrix, $\mathbf{z}$ is a variable vector containing the disparity variables $\{z_i | i = 1 \dots M\}$ and $\mathbf{B}$ is a vector.

Imposing normal constraints will make the generated disparity more smooth but may cause problems for discontinuous boundaries. We use the edge information from Section IV-B to delete normal constraints on edges.

E_f is dilated by 1 pixel so as to ensure disconnection of pixels on different sides of the edge. All equations related to normals of pixels on the edge region, where $E_f > 0$, are removed. Relative elements in A and B are also set to 0.

We also notice that the above linear system has infinite solutions. Sparse points with reliable disparity need to be applied as boundary conditions to constrain the solution. We use the disparity confidence introduced in Section IV-A to select a set of pixels with reliable disparities as additional soft constraints, and denote their weights as their disparity confidences C_P. Weights of unreliable disparities are all set to 0. So the depth prior constraint can be defined as

$$E_{obs} = ||\mathbf{W}(\mathbf{z} - \mathbf{z_0})||_{\mathbf{2}}^{\mathbf{2}}, \tag{12}$$

where $\mathbf{z}$ denotes the variable vector of disparity, $\mathbf{z_0}$ denotes the vector of pre-computed disparity, and $\mathbf{W}$ is a diagonal matrix indicating weights, which is defined as

$$\mathbf{W} = diag(w_1, w_2, \dots, w_M), \tag{13}$$

$$w_i = \begin{cases} C_P(x_i, y_i) & \text{if } C_{LRD}(x_i, y_i) > \tau_1 \\ & \text{and } C_P(x_i, y_i) > \tau_2, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

Here x_i and y_i are the x, y coordinates of pixel i, respectively.

We propose a simple noise-proof strategy to guarantee zero weight of noisy disparity at the far left of image without abandoning too many valid disparity pixels. We consider disparity of the leftmost m columns, where m is the maximum disparity allowed. These disparities are sorted by rows. 95th percentile of each row is picked up as a bound. Pixels on the left of bound are considered invalid, and their weights are set to 0.

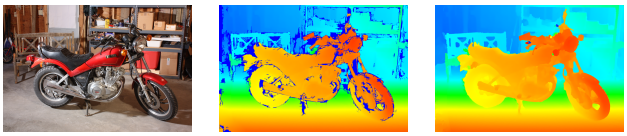
Finally, with a least squares method, we optimize the following energy function to convert the normal map to the disparity map

$$E(\mathbf{z}) = \lambda ||\mathbf{Az} - \mathbf{B}||_2^2 + ||\mathbf{W}(\mathbf{z} - \mathbf{z_0})||_{\mathbf{2}}^{\mathbf{2}}, \tag{15}$$

where λ is a weight and generally set to 1.0.

A. Iterative Disparity Refinement

Since sometimes edge fusion leads to closed regions which may not have reliable disparities, the depth conversion for these areas will fail, resulting in a few holes, as shown in Figure 4(b). Also, edge pixels could be assigned with invalid disparities due to the lack of normal and disparity constraints. We apply a bilateral filter to complete the disparities of the missing pixels.



(a) Left image (b) First iteration (c) Final result

Fig. 4. Iterative disparity refinement.

After disparity completion, we apply another reliable disparity selection for the next iteration. Very sparse disparity constraints may cause slight differences between these disparities and ground truth as the normal we used is roughly estimated and the conversion from normal to depth brings deviation if only normal constraints exist in some areas. Therefore, we would like to add more disparity constraints in each iteration to achieve more accurate results.

Let D_0 denote the pre-computed disparity map, D_k denote the disparity map after kth iteration. The element of weight matrix $\mathbf{W}$ in E_{obs} of $k+1$th iteration is

$$w_i = \begin{cases} C_P(x_i, y_i) & \text{if } C_P(x_i, y_i) > \tau_2 \\ & \text{and } |D_0(x, y) - D_k(x, y)| < \phi_k, \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Here ϕ_k is a threshold that initial value is set to 0.02 times of the maximum disparity allowed, and then decreased by 0.5 in each interation until it reaches 1.

As we have attained disparity map with less obvious noise, edge features with less noise could be extracted from the disparity map. To prevent redundant edge on planar region with large disparity gradient, after the first iteration, we use a combination of both disparity map edges from gradient and Laplace operators as well as image edges.

Then an edge tracking similar with the method in IV-B is performed, where all edge pixels that come from the Laplace operation are regarded as reliable edges at first, then gradient and image edges within the search region are added into the final edge set. The iterative process terminates when all pixels are assigned with valid disparities and no more new disparity constraints are added. Figure 4(c) shows a final converted disparity map.

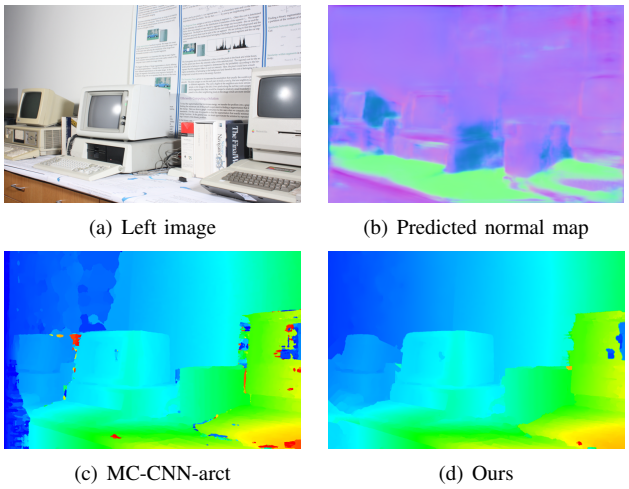
VI. Exprimental Results

To evaluate the performance of the proposed method, we have conducted experiments on both real captured indoor stereo pairs/sequences and Middlebury stereo datasets. We use MC-CNN method [39] or SGM [16] to compute matching cost and generate the original disparity maps. For normal prediction, we request the authors of [1] to help estimating the surface normal maps.

A. Evaluation of Accurate Architecture

We make comparisons with MC-CNN methods and other state-of-the-art methods on Middlebury stereo benchmark[1] [29]. Table I shows the error statistics for both Middlebury training dataset (denoted as TR) and test dataset (denoted as TE). We compare the percentage of bad pixels with

[1]http://vision.middlebury.edu/stereo/eval3/



(a) Left image (b) Predicted normal map

(c) MC-CNN-arct (d) Ours

Fig. 5. Vintge example. (a) Left image. (b) The predicted surface normal map of (a). (c) The estimated disparity map by MC-CNN-arct. (d) The refined disparity map by our method.

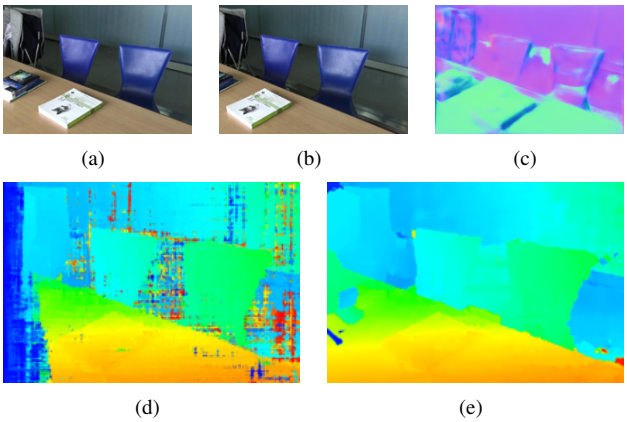
TABLE I

COMPARISON WITH OTHER METHODS ON THE MIDDLEBURY STEREO BENCHMARK

	Average Error					
	bad 4.0 (%)		bad 2.0 (%)		bad 1.0 (%)	
	TR	TE	TR	TE	TR	TE
Ours	12.0	11.4	18.0	16.6	27.6	26.3
MC-CNN+RBS[2]	14.5	13.9	19.3	18.1	28.0	27.5
MC-CNN-acrt[39]	15.7	15.8	19.7	19.1	27.7	27.3
MC-CNN-fst[39]	17.7	17.7	21.5	20.6	29.8	28.4
MDP[25]	14.6	15.6	20.1	20.2	36.3	37.4
MeshStereo[42]	15.2	14.5	20.9	19.8	32.1	32.9
TMAP[27]	16.5	18.2	22.7	24.6	35.7	38.6
IDR[24]	17.3	18.7	22.8	25.0	33.0	36.4

disparity error greater than 4.0, 2.0 and 1.0 pixels, respectively. Due to the limited memory size of graphics card, we use half resolution images to perform stereo matching and surface normal prediction. Before error computation, the half resolution images are upsampled to the full resolution images first. We firstly use MC-CNN-acrt to generate the initial disparity maps. As can be seen, our method achieves the lowest average error compared to the original MC-CNN based methods and other methods. As shown in Fig. 5(c), the recovered disparity map by MC-CNN [39] have obvious artifacts in occluded and textureless regions. With our method, the outliers are significantly reduced, as shown in Fig. 5(d). Fig. 6 shows another indoor example captured by ourselves. More examples can be found in our supplementary video.

Although our approach is mainly designed based on MC-CNN, it also can benefit other stereo matching methods. We further use SGM to generate original disparity maps, and evaluate our disparity improvement. We compare the original SGM results and ours on Middlebury stereo benchmark (training dataset). As SGM disparity result contains many holes and invalid pixels, we also compare our results with the disparity maps obtained by further applying median filter and bilateral filter to the original SGM disparity map. Table II shows the ratio of bad pixels in our result to those



(a) (b) (c)

(d) (e)

Fig. 6. An indoor example. (a) Left image. (b) Right image. (c) The predicted surface normal map of (a). (d) The estimated disparity map by MC-CNN-acrt. (e) The refined disparity map by our method.

TABLE II

RATIO OF BAD PIXELS

	Ours / SGM			Ours / filtered SGM		
	bad 4.0	bad 2.0	bad 1.0	bad 4.0	bad 2.0	bad 1.0
Adiron	0.61	0.78	0.91	0.79	0.89	0.97
ArtL	0.73	0.83	0.96	0.89	0.95	0.99
Jadepl	0.73	0.84	0.94	0.81	0.90	0.97
Motor	0.72	0.85	0.94	0.86	0.92	0.97
MotorE	0.62	0.79	0.92	0.86	0.95	1.01
Piano	0.82	0.90	0.96	0.91	0.94	0.97
PianoL	0.84	0.90	0.95	0.98	0.98	1.00
Pipes	0.87	0.95	1.01	0.94	0.97	1.00
Playrm	0.78	0.91	0.98	0.94	0.98	1.00
Playt	0.75	0.87	0.94	0.89	0.94	0.97
PlaytP	0.65	0.80	0.91	0.77	0.87	0.93
Recyc	0.60	0.79	0.93	0.81	0.91	0.98
Shelvs	0.86	0.93	0.97	0.96	0.98	0.99
Teddy	0.69	0.86	1.01	0.83	0.93	0.98
Vintge	0.79	0.91	0.96	0.94	0.99	0.99
Avg	0.74	0.86	0.95	0.88	0.94	0.98

in SGM and filtered SGM. Although the generated disaprity map by SGM has obvious artifacts around discontinuous boundaries which influence the performance of our edge fusion, our approach still produces better results than SGM and its filtered version in most cases.

B. Evaluation of Fast Architecture

MC-CNN generally takes about 20 seconds to compute a stereo pair with resolution 694×554, and SGM takes about 0.34 second. Normal estimation is around 0.2 second with Nvidia Titan X graphics card for any input images resized to $224 \times 224 \times 3$. Based on the estimated initial disparity map and surface normal map, our method needs to further take about 99 seconds to process a stereo image with 694×554 resolution, which cannot be applied to real-time applications. In order to satisfy real-time applications for robotics, we propose to simplify our approach with parallel computation.

For our fast architecture, we use SGM to compute initial disparity map and use Left Right Consistency (LRC) proposed in [6] to estimate disparity confidence. The related elements of weight matrix $\mathbf{W}$ in E_{obs} are all set to 1 if these pixels have small LRC error and their disparity variances are

TABLE III

ERROR STATISTICS COMPARISON BETWEEN OUR METHOD WITH FAST ARCHITECTURE (1 AND 2 ITERATIONS RESPECTIVELY) AND FILTED SGM.

	Bad pixel 1.0 (%)		Time (s)	
	filtSGM	Ours	filtSGM	Ours
Adiron	65	52 / 49	30.9	8.4 / 15.0
ArtL	57	65 / 56	8.2	1.8 / 3.3
Jadepl	75	75 / 67	29.8	8.6 / 13.7
Motor	67	43 / 39	32.3	8.6 / 15.5
MotorE	59	55 / 47	32.3	7.9 / 14.4
Piano	71	52 / 48	31.2	7.7 / 14.5
PianoL	80	67 / 63	29.5	7.1 / 13.3
Pipes	58	51 / 40	31.1	7.7 / 13.7
Playrm	69	64 / 59	28.9	7.4 / 13.4
Playt	82	72 / 65	29.3	6.7 / 12.0
PlaytP	82	53 / 47	41.5	6.7 / 12.5
Recyc	69	47 / 41	31.1	8.2 / 15.1
Shelvs	75	65 / 64	32.9	8.6 / 15.7
Teddy	37	40 / 35	14.8	4.2 / 7.3
Vintge	88	65 / 63	33.9	9.9 / 16.3
Avg	69	58 / 52	29.2	7.3 / 13.0

small compared to those in the last iteration. The edge is formed by intersection of canny edge of image and dilated sobel edge of disparity map only. As SGM often results in fat edges on discontinuous region, we tend to abandon disparity constraints of pixels near edges in this case to reduce artifacts. The limited quality of disparity map from SGM may result in more redundant edges and closed regions. To reduce the proportion of failed pixels in solving the linear equations, we set a small weight (0.0001 in our experiments) to the depth constraints of all unreliable pixels except those invalid ones on the left boundary. We apply a median filter instead of bilateral filter to complete the disparities of the missing pixels. Compared to the original disparity map, our refined disparity map is more complete without obvious holes. The final disparity map is obtained by solving (15) with 1 or 2 iterations (λ is set to 0.1).

We test our method with fast architecture on both Middlebury stereo dataset and indoor stereo sequences. Table III shows comparisons on bad pixel percentage and running time of our fast version (1 and 2 iterations respectively with Matlab code) and filtered SGM. As can be seen, our simplified version not only produces better results than those by applying median filter and bilateral filter to the original SGM disparity map, but also runs much faster. Even with only one iteration conversion, our method can get better result than that of filtered SGM. More iterations can improve the result but computation time increases. Figure 7 shows the results of some selected frames from a video sequence captured by stereo camera with resolution of 640×480. The produced disparity maps by our method with fast architecture are obviously better than those by filtered SGM. We notice that our fast architecture outperforms our accurate version in some cases here. Due to the fat disparity edges in SGM mentioned above, the accurate architecture sometimes introduces blurring artifacts around discontinuous boundaries.

The normal prediction runs on a PC with Nvidia Titan X graphics card and takes about 0.2s per frame. The running

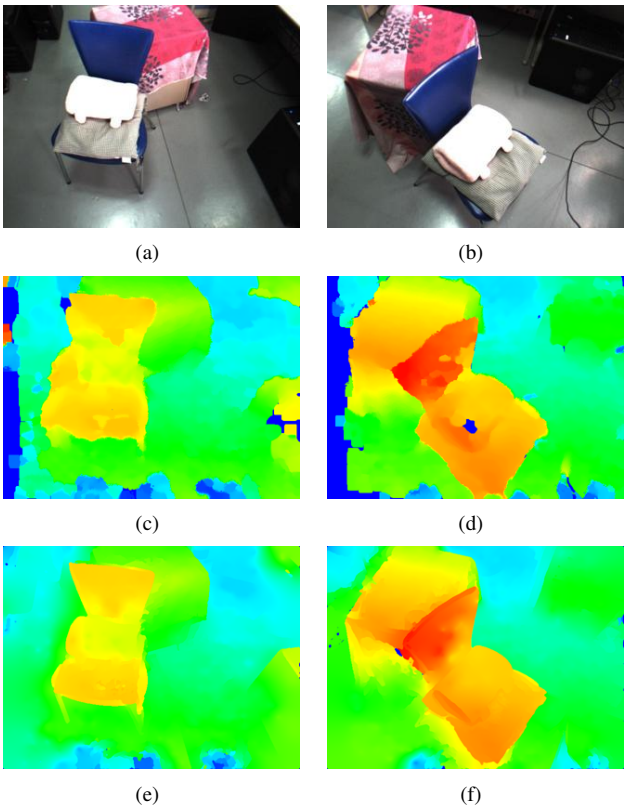


Fig. 7. The disparity result of an indoor stereo sequence. (a-b) The left images of selected frames. (c-d) The estimated disparity maps by filtered SGM. (e-f) The estimated disparity maps by our method.

time of other components with fast architecture is about 1.4 seconds per stereo pair (640 × 480 resolution) with Matlab code on a 4 core i5-4590 3.30GHz CPU without GPU acceleration. We also implement C++ code and accelerate the solving of the sparse linear system by conjugate gradient algorithm with GPU implementation. We found that using conjugate gradient algorithm with 200 iterations is generally enough, which takes 0.23s on a PC with GTX 960 display card. Thus, with GPU acceleration of linear equation solver, the whole process of our fast version takes about 0.75s per frame with resolution of 640×480, which is acceptable for some real-time applications.

VII. CONCLUSIONS

In this paper, we propose a robust stereo matching method with surface normal prediction, which can significantly improve the disparity estimation result especially in the occluded and textureless regions. In order to achieve this goal, we first use a single image based surface normal prediction method to recover the normal map of left image. Simultaneously, we perform stereo matching and measure the disparity confidence to select the reliable disparities. Using the estimated disparity map with confidence and the fused edge information from different cues, we faithfully convert the surface normal to disparity by solving a linear system. Finally, we complete the missing holes by iteratively applying bilateral-filtering based completion and edge feature

refinement. Our method with fast architecture can achieve over 1 fps for a stereo sequence with resolution of 640×480 and still obviously improve the disparity results compared to SGM. The experimental results of both real captured indoor stereo pairs/sequences and Middlebury stereo dataset demonstrate the effectiveness of the proposed approach.

ACKNOWLEDGMENTS

We would like to thank Aayush Bansal for his kind help in computing surface normal maps. This work was partially supported by the National Key Research and Development Program of China (No. 2016YFB1001501), NSF of China (Nos. 61232011 and 61672457), and a Foundation for the Author of National Excellent Doctoral Dissertation of PR China (Grant No. 201245).

REFERENCES

[1] A. Bansal, B. Russell, and A. Gupta. Marr revisited: 2D-3D alignment via surface normal prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2016.

[2] J. T. Barron and B. Poole. The fast bilateral solver. In *European Conference on Computer Vision*, volume 3, pages 617–632, 2016.

[3] M. Bleyer, C. Rother, and P. Kohli. Surface stereo with soft segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1570–1577. IEEE, 2010.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.

[6] G. Egnal, M. Mintz, and R. P. Wildes. A stereo confidence metric using single view imagery with comparison to five alternative approaches. *Image Vision Computing*, 22(12):943–957, 2004.

[7] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[8] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems*, pages 2366–2374, 2014.

[9] I. Ernst and H. Hirschmüller. Mutual information based semi-global stereo matching on the GPU. In *International Symposium on Visual Computing*, pages 228–239. Springer, 2008.

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54, 2006.

[11] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[12] S. K. Gehrig, F. Eberli, and T. Meyer. A real-time low-power stereo vision engine using semi-global matching. In *IEEE International Conference on Computer Vision*, pages 134–143. Springer, 2009.

[13] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *Asian Conference on Computer Vision*, pages 25–38. Springer Berlin Heidelberg, 2010.

[14] F. Guney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4165–4175, 2015.

[15] C. Hane, L. Ladicky, and M. Pollefeys. Direction matters: Depth estimation with a surface normal classifier. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 381–389, 2015.

[16] H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 807–814. IEEE, 2005.

[17] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):328–341, 2008.

[18] H. Hirschmüller and D. Scharstein. Evaluation of cost functions for stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007.

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

[20] X. Hu and P. Mordohai. A quantitative evaluation of confidence measures for stereo vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2121–2133, 2012.

[21] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, 1994.

[22] K. R. Kim and C. S. Kim. Adaptive smoothness constraints for efficient stereo matching using texture and edge information. In *IEEE International Conference on Image Processing*, pages 3429–3433, Sept 2016.

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

[24] J. Kowalczuk, E. Psota, and L. C. Pérez. Real-time stereo matching on CUDA using an iterative refinement method for adaptive support-weight correspondences. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(1):94–104, 2013.

[25] A. Li, D. Chen, Y. Liu, and Z. Yuan. Coordinating multiple disparity proposals for stereo computation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4022–4030, 2016.

[26] V. Nozick. Pyramidal normal map integration for real-time photometric stereo. In *EAM Mechatronics*, pages 128–132, 2010.

[27] E. T. Psota, J. Kowalczuk, M. Mittek, and L. C. Pérez. MAP disparity estimation using hidden markov trees. In *IEEE International Conference on Computer Vision*, pages 2219–2227, 2015.

[28] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition*, pages 31–42, 2014.

[29] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1-3):7–42, 2002.

[30] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 195–202, 2003.

[31] D. Shreiner, M. Woo, J. Neider, and T. Davis. *OpenGL Programming Guide, Sixth Edition: The Official Guide to Learning OpenGL, Version 2.1*. Addison-Wesley, 5 edition, 2008.

[32] A. Spyropoulos, N. Komodakis, and P. Mordohai. Learning to detect ground control points for improving the accuracy of stereo matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1621–1628, 2014.

[33] H. Trinh and D. McAllester. Unsupervised learning of stereo vision with monocular cues. In *British Machine Vision Conference*, pages 72–1, 2009.

[34] X. Wang, D. Fouhey, and A. Gupta. Designing deep networks for surface normal estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 539–547, 2015.

[35] T.-P. Wu and C.-K. Tang. Visible surface reconstruction from normals with discontinuity consideration. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1793–1800. IEEE, 2006.

[36] Q. Yang. Hardware-efficient bilateral filtering for stereo matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):1026–1032, 2014.

[37] Q. Yang. Stereo matching using tree filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(4):834–846, 2015.

[38] K.-J. Yoon and I. S. Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):650–656, 2006.

[39] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17:1–32, 4 2016.

[40] B. Zeisl and M. Pollefeys. Discriminatively trained dense surface normal estimation. In *European Conference on Computer Vision*, pages 468–484. Springer International Publishing, 2014.

[41] C. Zhang, Z. Li, R. Cai, H. Chao, and Y. Rui. As-rigid-as-possible stereo under second order smoothness priors. In *European Conference on Computer Vision*, pages 112–126. Springer International Publishing, 2014.

[42] C. Zhang, Z. Li, Y. Cheng, R. Cai, H. Chao, and Y. Rui. Meshstereo: A global stereo model with mesh alignment regularization for view interpolation. In *IEEE International Conference on Computer Vision*, pages 2057–2065, 2015.