PM-PM: PatchMatch With Potts Model for Object Segmentation and Stereo Matching

Shibiao Xu, *Member, IEEE*, Feihu Zhang, *Student Member, IEEE*, Xiaofei He, *Fellow, IEEE*, Xukun Shen, *Member, IEEE*, and Xiaopeng Zhang, *Member, IEEE*

***Abstract*— This paper presents a unified variational formulation for joint object segmentation and stereo matching, which takes both accuracy and efficiency into account. In our approach, depth-map consists of compact objects, each object is represented through three different aspects: 1) the perimeter in image space; 2) the slanted object depth plane; and 3) the planar bias, which is to add an additional level of detail on top of each object plane in order to model depth variations within an object. Compared with traditional high quality solving methods in low level, we use a convex formulation of the multilabel Potts Model with PatchMatch stereo techniques to generate depth-map at each image in object level and show that accurate multiple view reconstruction can be achieved with our formulation by means of induced homography without discretization or staircasing artifacts. Our model is formulated as an energy minimization that is optimized via a fast primal-dual algorithm, which can handle several hundred object depth segments efficiently. Performance evaluations in the Middlebury benchmark data sets show that our method outperforms the traditional integer-valued disparity strategy as well as the original PatchMatch algorithm and its variants in subpixel accurate disparity estimation. The proposed algorithm is also evaluated and shown to produce consistently good results for various real-world data sets (KITTI benchmark data sets and multiview benchmark data sets).**

***Index Terms*— Object segmentation, stereo matching, Potts model, PatchMatch, multiple view reconstruction.**

I. Introduction

OBJECT class segmentation and dense stereo matching have been long-standing problems in computer vision, there are many high-quality methods treat them as two independent problems. In fact, the problems of object class segmentation, which assigns an object label to every pixel in the image, and dense stereo matching, in which every pixel within an image is labelled with a disparity (inversely proportional to the depth), are well suited for being solved jointly. Both approaches formulate the problem of providing a correct labelling of an image as one of Maximum a Posteriori (MAP) estimation. As a correct labelling of object class can inform depth labelling, and stereo matching can also improve object labelling. In intuition, the object class boundaries are more likely to occur at a sudden transition in depth and vice versa. While these two problems are *mutually informative*, few attempts have been made to jointly optimise their labellings, that is exactly what we concern.

Many existing segment-based stereo matching methods (see [1]–[6]) typically use low level segmentation methods to over-segment the image in a pre-processing step, and use the resulting segments either as a hard or soft constraint. Thus, to some extent, the two tasks are solved separately, and depth estimating results are subject to the quality of the final resulting segments. In contrast, inspired by the minimal partitions approaches [7], [8], we employ object-level segmentation as a soft constraint, which can aid depth estimation in powerful ways. That means our method jointly computes segmentation and depth iteratively, aiming to recover large segments corresponding to entire objects, and using a symmetric process that yields a consistent segmentation of both images. In our approach, we assume that depth-map consists of compact objects, which means each object both has a smooth boundary and smooth appearance variations at the same time. To enforce local smoothness of the segmentation, we use a Potts prior [9] with label costs [10] minimizing the contour length of each segment. Hence the depth borders are enforced to be smooth as the depth within segments is modeled with object depth plane in a parametric form.

Most stereo correspondence algorithms (see [11]–[14]) match support windows at integer-valued disparities and assume a constant disparity value within the support window. The recently proposed PatchMatch stereo algorithm [15] can overcome these limitations by directly estimating highly slanted planes, and achieve impressive disparity details with sub-pixel precision. In contrast to this local algorithm, we smartly traverse parts of it and enable a global optimization where depth-map and object planes are estimated jointly. Based on the above compact objects assumption, we propose a planar bias model to illustrate that objects may have a bias towards being planar in 3D space. Our model is enforced to compute an individual depth plane at each pixel onto which

Manuscript received September 28, 2014; revised January 26, 2015; accepted March 12, 2015. Date of publication March 25, 2015; date of current version April 6, 2015. This work was supported by the National Natural Science Foundation of China under Grant 61332017, Grant 61331018, Grant 91338202, and Grant 61271430. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Chang-Su Kim. *(Corresponding author: Xiaopeng Zhang.)*

S. Xu and X. Shen are with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China (e-mail: shibiao.xu@buaa.edu.cn; xkshen@buaa.edu.cn).

F. Zhang is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: feihu.zhang@nlpr.ia.ac.cn).

X. He is with the College of Computer Science, Zhejiang University, Hangzhou 310027, China (e-mail: xiaofeihe@cad.zju.edu.cn).

X. Zhang is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: xiaopeng.zhang@ia.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2416654

the support object region is projected, together with the planar bias, it allows to extend depth plane in the same object region and to precisely capture depth discontinuities planar.

In this paper, we draw a novel connection between two kinds of existing computer vision problems, object class segmentation and dense stereo matching, to estimate the correspondence fields between images in object level, where both *accuracy* and *efficiency* are taking into account. The key of our method is the joint optimization process for an efficient convex formulation of the multi-label Potts Model with global PatchMatch stereo techniques to generate precise depth-map at each image in object level. Furthermore, we do not assume that the images were captured in a certain environment, such as indoor or outdoor, our only assumption is that the scene is assembled of compact objects. Compared to state-of-the-art methods [16], [17], the proposed method has three main advantages: 1) It is able to reconstruct highly object slanted surfaces, and achieve impressive disparity details with sub-pixel precision simultaneously, which outperforms other patch-based methods. 2) It is a computational efficient method, which could be easily parallelized for the computation of object plane and depth plane at each pixel, this makes the proposed method 10 to 20 times faster than the state-of-the-art methods while attaining the better accuracy. 3) It is suitable for accurate large-scale scene reconstruction without discretization or staircasing artifacts for high resolution images in object level.

II. Previous Works

Considering that the proposed approach allows for simultaneously segmenting complex scenes and improving the robustness of the depth estimation, both object segmentation methods and stereo matching methods are reviewed in this section.

Segmentation denotes the task of dividing an image into meaningful nonoverlaping regions. In general, there are two types of segmentation methods: unsupervised segmentation and supervised segmentation. K-means [18], Mean shift [19] and Normalized cuts [20] are three major unsupervised segmentation methods that do not need predefined parameters to describe each segment determines which category it belongs to. Over the last few years, we observed a number of breakthroughs in the supervised segmentation regarding algorithmic approaches to efficiently compute the minimum energy solutions for respective cost functions, using graph cuts [21], [22], level set method [23], random walks [24], and convex relaxation techniques [25]. Potts Model [26]–[28] is a popular supervised segmentation model. To find the optimal solution, three convex relaxations of Potts Model were proposed by Lellmann et al. [29], by Zach et al. [30] and by Pock et al. [7]. In this paper, we use recent advances in solving the Potts Model to derive a novel primal-dual energy minimization problem. A standard primal-dual algorithm allows to use several hundred labels in a reasonable time using a GPU implementation. For the first time, it is possible to handle such a big amount of labels in a depth-map segmentation approach making the method applicable to complex scenes.

Stereo matching algorithms usually takes four steps [31]: matching cost computation, cost aggregation, disparity computation and disparity refinement. The matching costs are initialized for each pixel at all possible disparity levels; and the costs are aggregated over each pixel's support region [32], [33]; then the disparities are computed with a local [12], [34] or global [35] optimizer; finally the disparity results are refined with various post-processing techniques. In general, many stereo matching algorithms are based on the assumption that the pixels within the matching window share the same disparity value, and discrete disparity are often considered leading to discrete depth layers. To avoid the above problems, the PatchMatch stereo algorithm [15] was initially introduced as a computationally efficient way to compute correspondence fields between images. It shows that the PatchMatch algorithm can be applied for stereo matching using slanted support windows so that instead of just estimating a single depth value for each pixel a complete depth plane estimation is made. There, depth plane is overparameterized by a 3D vector at each node $\mathbf{h_p} = \left[a_p, b_p, c_p\right]^T$, parameterizing a planar depth surface $d(\mathbf{p}) = a_p x + b_p y + c_p$. It is important that the PatchMatch stereo algorithm does not try to discretize the space of the likelihood function, instead it relies on randomized sampling for the solution space and propagation of good estimates using the spatial neighborhood. Benefit from the complementary advantages of PatchMatch and edge-aware filtering (EAF) [12], PatchMatch Filter (PMF) [36] was proposed to perform random search, label propagation and efficient cost aggregation collaboratively. This kind of patch-based method has also been integrated into global optimization framework with different considerations and requirements. Heise et al. [17] integrate the PatchMatch stereo algorithm into a variational smoothing formulation using quadratic relaxation, which allows the explicit Huber regularization (PM-Huber) of the disparity and normal gradients using the estimated plane parameters. Besse et al. [37] point out a close relationship between PatchMatch and belief propagation, and present a unified method called PatchMatch belief propagation (PMBP) for pairwise continuous MRFs, which is more accurate than PatchMatch stereo algorithm. In this paper, we smartly traverse parts of PatchMatch stereo algorithm and enable a global optimization using local sampling labels, which is solved via graph cuts [4], [21] instead of belief propagation [1], [38]. Therefore, our method is able to take advantage of better convergence of graph cuts for achieving greater accuracy. Moveover, our method allows the parallel computation of matching costs.

The proposed objective function is embedded in a variational framework that allows for simultaneously segmenting complex scenes and improving the robustness of the depth estimation. Therefore, the approach which is closest to ours is ObjectStereo [16]. ObjectStereo algorithm jointly estimates objects and depth. To our knowledge, it is the only work which has shown a synergy effect between object class segmentation and dense stereo matching. It is worth of noting that the focus of ObjectStereo and our method are to show that depth estimation can be improved by introducing the notion of objects. In this context, Bleyer et al. [39] and Ladicky et al. [40] have also shown the synergy effect between depth estimation

and object-class extraction respectively, however, they either focus on the extraction of objects or rely on a priori defined object class. Besides that, Häne et al. [41] solve jointly for semantic segmentation and 3D reconstruction that leads to a better reconstruction result for textureless region, however, the computational efficiency needs to be improved for spatial resolution with abundant regular voxel grids. As for stereo matching, the depth ordering is essential, many complex approaches that jointly try to optimize for segmentation and depth ordering were recently proposed by Sun et al. [42] for multiple layers or Zhang et al. [43] for two layers. However, most of these mentioned methods are defined in a discrete setting. In contrary, the proposed method will be defined in a continuous setting, which can achieve more accurate results.

A different research direction is to solve the two tasks separately, that is depth-map is estimated based on pre-computed objects class. These kind of methods such as segment-based stereo [1]–[6] usually assign a 3D depth plane for each of over-segmented image regions. The candidate planes are generated by fitting planes to a roughly estimated depth-map, and then the optimal assignment of the planes is estimated by, e.g., graph cuts with expansion [4], [21] moves or belief propagation [1], [38]. Although this approach yields continuous-valued disparities, it strictly limits the reconstruction to a piecewise planar representation. Also, the results are subject to the quality of the segmentation.

III. Problem Statement

In this section, we will describe the underlying energy formulation for our proposed joint stereo matching and object segmentation. Our goal is to generate depth-map at each image in object level, while taking both *accuracy* and *efficiency* into account.

A. Depth Plane Representation

When estimating the disparity field, only the left and right image pairs are used. We start with these two given rectified stereo color images $I_1, I_2 : \Omega \rightarrow \mathbb{R}^3$. We define correspondence field $d : \Omega \rightarrow \mathbb{R}$ as the disparity computed from image I_1 to image I_2.

According to [15], our model is enforced to compute an individual slanted depth plane that has minimal aggregated matching cost between image pairs for each pixel, and each image point $\mathbf{p} = [x, y, 1]^T$ is assigned with the following parameters: (1) The disparity value d and (2) an normal $\mathbf{n} = [n_x, n_y, n_z]^T$. In our method, we randomly initialize the disparity value and normal vector within the reasonable ranges, for more details on computing a random plane at each pixel, please refer to Section IV-B. Then we calculate the parameter υ of a plane $\pi = [\mathbf{n}^T, \upsilon]^T$ with $\upsilon = \mathbf{n}^T X_{\mathbf{p}} = -n_x x - n_y y - n_z d$, where $X_{\mathbf{p}}$ is the corresponding 3D space point of image point $\mathbf{p}$. This follows from $\pi^T [x, y, d, 1]^T = 0$, which must hold if the point $X_{\mathbf{p}}$ lies on the plane π. Therefore, the disparity value d_π of any image point $[x_\pi, y_\pi, 1]^T$ (including image point $\mathbf{p} = [x, y, 1]^T$) on the plane is given by

$$d_\pi = \frac{-n_x x_\pi - n_y y_\pi + (n_x x + n_y y + n_z d)}{n_z} \tag{1}$$

We can reformulate this as a linear transformation $H_\pi(d, \mathbf{n})$ in dense stereo matching assuming that the corresponding point of $\mathbf{p}$ in the second image is given by $\mathbf{p}' = \mathbf{p} - [d_\pi, 0, 0]^T = [x - d_\pi, y, 1]^T$ with $d_\pi = d$ being the disparity as

$$\mathbf{p}' = \begin{pmatrix} 1 + \frac{n_x}{n_z} & \frac{n_y}{n_z} & -\frac{n_x}{n_z}x - \frac{n_y}{n_z}y - d \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{p} = H_\pi(d, \mathbf{n})\mathbf{p} \tag{2}$$

It is worth of noting that the proposed depth plane representation is also appropriate for accurate multi-view reconstruction with some adjustments. In the general case, the camera parameters of the selected image pair $I_i, I_j : \Omega \rightarrow \mathbb{R}^3$ are $\{K_i, R_i, C_i\}$ and $\{K_j, R_j, C_j\}$, where K is the intrinsic parameters, R is the rotation matrix, and C is the camera center. Thus, the general projection matrices for the two cameras are $P = K_i[I_{3\times3}|\mathbf{0}_3]$ and $P^{'} = K_j[R_{3\times3}|\mathbf{t}] = K_j[R_j R_i^{-1}|R_j(C_j - C_i)]$ with the world origin set at the first camera. In this case, the correspondence field $d : \Omega \rightarrow \mathbb{R}$ needs to be represented as the depth instead of disparity, that means the corresponding 3D space point of image point $\mathbf{p}$ is computed in I_i's coordinate as $X_p = dK_i^{-1}\mathbf{p}$. Then, the resulting induced homography mapping [44] in multi-view reconstruction is given by

$$H_\pi(d, \mathbf{n}) = K_j(R_j R_i^{-1} - \frac{R_j(C_j - C_i)\mathbf{n}^T}{\mathbf{n}^T X_p})K_i \tag{3}$$

for a depth plane $\pi = [\mathbf{n}^T, \upsilon]^T$ with normal $\mathbf{n}$ and parameter $\upsilon = \mathbf{n}^T X_{\mathbf{p}}$. In special, it is obvious that the induced homography mapping will become the "linear transformation" in stereo matching when the camera is set with identity rotation and translation in horizontal direction.

In the matching process, we set a square window $N(\mathbf{p})$ centered on pixel p. For each pixel $\mathbf{q}$ in $N(\mathbf{p})$ we find its corresponding pixel in image I_2 using homography mapping $H_\pi(d, \mathbf{n})$. Then the aggregated matching cost $m(\mathbf{p}, (d, \mathbf{n}))$ for pixel $\mathbf{p}$ is computed as

$$m(\mathbf{p}, (d, \mathbf{n})) = \frac{1}{Z} \sum_{\mathbf{q} \in N(\mathbf{p})} \omega(\mathbf{p}, \mathbf{q}) \rho(\mathbf{q}, H_\pi(d, \mathbf{n})\mathbf{q}) \tag{4}$$

where the function ω computes a weighting based on the color similarity between the corresponding pixels $\omega(\mathbf{p}, \mathbf{q}) = e^{-\mu \|I_1(\mathbf{p}) - I_1(\mathbf{q})\|_1}$, μ is user-defined parameter, and $\|I_1(\mathbf{p}) - I_1(\mathbf{q})\|_1$ computes the L_1 distance of $\mathbf{p}$ and $\mathbf{q}$'s colors in integer RGB vector space. The pixel similarity is measured by

$$\begin{aligned} \rho(\mathbf{q}, H_\pi(d, \mathbf{n})\mathbf{q}) = {} & (1 - \alpha) \cdot min(\|I_1(\mathbf{q}) - I_2(\mathbf{q}')\|_1, \tau_{col}) \\ & + \alpha \cdot min(\|\nabla I_1(\mathbf{q}) - \nabla I_2(\mathbf{q}')\|_1, \tau_{grad}) \end{aligned} \tag{5}$$

where $\|\nabla I_1(\mathbf{q}) - \nabla I_2(H_\pi(d, \mathbf{n})\mathbf{q})\|_1$ denotes the absolute difference of gray-value gradients computed at $\mathbf{q}$ and $\mathbf{q}' = H_\pi(d, \mathbf{n})\mathbf{q}$. Since the x-coordinate of $\mathbf{q}$ lies in the continuous domain, we derive its color and gradient values by linear interpolation. The user-defined parameter α balances the influence of color and gradient term. Parameters τ_{col} and τ_{grad}

truncate costs for robustness in occlusion regions. Z is an normalization constant with

$$Z = \sum_{\mathbf{q}\in N(\mathbf{p})} \omega(\mathbf{p}, \mathbf{q}). \tag{6}$$

B. Object Representation

As stated above, we represent depth-map as a collection of compact objects and partition the image domain Ω into K pairwise disjoint object regions Ω_i. In general, it is oversimplified that object's depth is estimated by a 3D plane roughly. Therefore, we add an additional level of detail on top of the object planes in order to model depth variations within an object and compute a depth bias value at each pixel $\mathbf{p}$ within an object Ω_i. Then, each object contains the following components: (1) Perimeter, (2) Planar bias and (3) an object plane.

The first component of each object, perimeter is used to smooth the object boundaries as

$$\text{Per}\ (\Omega_i; \Omega), \quad s.t. \quad \Omega = \bigcup_{i=1}^{K} \Omega_i, \Omega_i \bigcap \Omega_j = \emptyset \quad \forall i \neq j \tag{7}$$

where $\text{Per}(\Omega_i; \Omega)$ denotes the boundary length of Ω_i, and the perimeter of a satisfactory partition should be as small as possible. Let

$$u_i = \mathbb{1}_{\Omega_i} = \begin{cases} 1 \ \ if \quad \mathbf{p} \in \Omega_i \\ 0 \ \ otherwise \end{cases}$$

be the characteristic function of Ω_i, we have $\text{Per}(\Omega_i) = \int_\Omega g(\mathbf{p}) \|\nabla u_i\|_1$, which is the Total Variation of u_i (discussed in detail later). Note that the norm in $\|\nabla u_i\|_1$ is point wise. Similar to the weighted Total Variation studied in [45], we add a g-weighting term to pull the segmentation boundaries towards strong image gradients, which should be possible depth edges, we calculate the per-pixel weighting function $g(\mathbf{p})$ with

$$g(\mathbf{p}) = e^{-\epsilon \|\nabla I(\mathbf{p})\|_1^{\nu}} \tag{8}$$

where ϵ and ν are user-defined parameters.

The latter two components of each object, planar bias and object plane are used to regularize the depth-map by

$$-log(\pi_i(d_{\Omega_i}(\mathbf{p}) - d(\mathbf{p}))) \tag{9}$$

where $d_{\Omega_i}(\mathbf{p}) - d(\mathbf{p})$ represents the disparity bias at pixel $\mathbf{p}$, which is obtained by subtracting $\mathbf{p}$'s disparity $d_{\Omega_i}(\mathbf{p}) = h_{\Omega_i}\mathbf{p} = a_i x + b_i y + c_i$ according to the object plane h_{Ω_i} (discussed in detail later) from its disparity $d(\mathbf{p})$ according to the depth-map. In our solution, we enforce that bias values have a consistent distribution within object Ω_i, which is called Planar bias model, and we implement the bias distribution as a Gaussian Mixture Model (GMM), which is updated in each optimization iteration in Section IV. The function $\pi_i(d_{\Omega_i}(\mathbf{p}) - d(\mathbf{p}))$ returns the probability of planar depth bias in the GMM of object Ω_i. In fact, this probability is maximized if an object's GMM captures very similar disparity biases with low variance among them. Hence, in order to achieve unified disparity inside object Ω_i, the regularization term (9) is minimized for the occurrence of a specific bias in object Ω_i and tries to avoid disparity biases that have low probabilities.

In order to determine the parameters of object plane h_{Ω_i} in the Planar bias model, a straightforward way is to solve a least square system, which is very sensitive to outliers. We give a robust method by applying a decomposition method to solve each parameter separately. First, the horizontal slant a_i is estimated using a set of reliable pixels lying in the same horizontal line within the segment Ω_i, and these reliable pixels can be obtained with disparity left-right consistency checking. The disparity gradients $\delta d/\delta x$ for all reliable pixels are computed and inserted to a list, then the robust estimation of the horizontal slant a_i is determined by sorting the disparity gradient list in descending order and applying Gaussian filtering (Gaussian weighted average) from the middle of the list. Second, the vertical slant b_i is estimated in a similar manner by considering the disparity gradients $\delta d/\delta y$ of all reliable pixels lying on the same vertical line. Third, the determined slant c_i is obtained directly with Gaussian filtering for all the reliable disparities from the center of the segment Ω_i.

In summary, we define the joint segmentation and reconstruction problem as

$$\begin{aligned} \mathcal{E}(\Omega_i, d, \mathbf{n}) = \text{Per}\ (\Omega_i) + \gamma \|\mathbb{1}_{\Omega_i}\|_\infty + \sigma \int_{\Omega_i} f_i^c(\mathbf{p}) d\mathbf{p} \\ + \lambda \int_{\Omega_i} (-log\pi_i(d_{\Omega_i}(\mathbf{p}) - d(\mathbf{p})))d\mathbf{p} \\ + \tau \int_{\Omega_i} m(\mathbf{p}, (d, \mathbf{n}))d\mathbf{p} \end{aligned} \tag{10}$$

where $\|\mathbb{1}_{\Omega_i}\|_\infty$ is a label cost term to constrain the maximal partition number as in [10], which ensures that the number of used segments is as small as necessary. The infinity norm penalizes the maximum value of the characteristic function, and will minimize the number of non-empty segments. Moreover, we evaluate the color costs at each pixel using $f_i^c(\mathbf{p}) = \|I_{mean}(\Omega_i) - I(\mathbf{p})\|_1$, which enforces that an object has a compact set of colors. We then try to obtain the desired Ω_i, d and $\mathbf{n}$ by minimizing the following energy function

$$\begin{aligned} &\min_{\Omega_i, d, \mathbf{n}} \sum_{i=1}^{K} \mathcal{E}(\Omega_i, d, \mathbf{n}), \\ &s.t. \quad \Omega = \bigcup_{i=1}^{K} \Omega_i, \quad \Omega_i \bigcap \Omega_j = \emptyset \quad \forall i \neq j. \end{aligned} \tag{11}$$

IV. Proposed Joint Optimization

The optimization problem in (11) poses a difficult non-convex optimization problem. Fortunately, the model is convex in the segmentation Ω_i and convex in the disparity d and normal $\mathbf{n}$ respectively. Hence, we can split up the optimization into two subproblems, Potts Model based object segmentation and PatchMatch based stereo matching, and use alternative direction method [46] to obtain a good approximate solution of the proposed model. The overall framework of the proposed method is illustrated in Fig. 1. In the following, each step of the algorithm is described in detail.

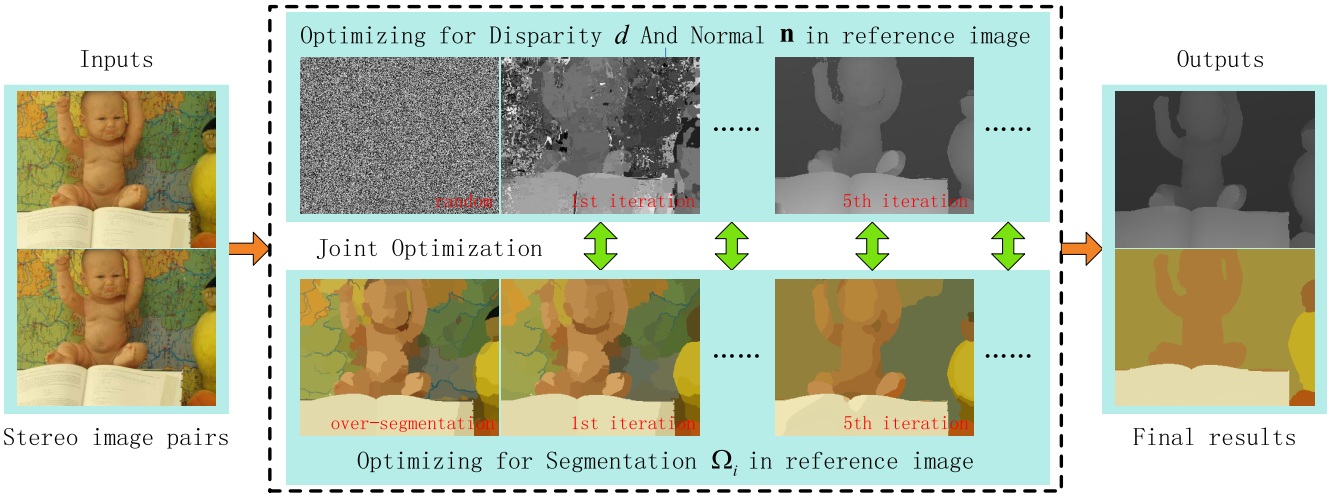


Fig. 1. Overall framework of the proposed method, which captures the interdependence between the object class segmentation problem and the dense stereo matching problem by allowing interactions between them. From left to right: Stereo image pairs, our disparity map and object map after initialization, disparity map and object map after the 1st iteration, disparity map and object map after the 5th iteration, the final disparity map and object map.

A. Optimizing for Segmentation Ω_i

For the fixed disparity d and normal $\mathbf{n}$ in (11), the first proposed subproblem, segmentation formulates the problem of providing a correct labelling of an image as one of Maximum a Posteriori (MAP) estimation, which is typically a generalised Potts truncated linear model [47].

$$\min_{\Omega_i}\left\{\sum_{i=1}^{K}\mathrm{Per}\ (\Omega_i)+\gamma\,\|\mathbb{1}_{\Omega_i}\|_\infty+\sum_{i=1}^{K}\int_{\Omega_i}f_i(\mathbf{p})d\mathbf{p}\right\}$$
$$s.t.\quad \Omega=\bigcup_{i=1}^{K}\Omega_i,\quad \Omega_i\bigcap\Omega_j=\emptyset\quad\forall i\neq j \tag{12}$$

where $f_i(\mathbf{p})=\lambda(-log(\pi_i(d_{\Omega_i}(\mathbf{p})-d(\mathbf{p}))))+\tau m(\mathbf{p},(d,\mathbf{n}))+\sigma f_i^c(\mathbf{p})$ represents the potential function in each region. To solve the Potts based energy with label cost, we first have to apply some relaxations. Similar to the work of Zach et al. [30], we calculate the perimeter of the regions Ω_i using Total Variation $\int_\Omega g(\mathbf{p})\|\nabla u_i\|_1$, and relax the characteristic function $u_i=\mathbb{1}_{\Omega_i}$ to the continuous range $[0,1]$. Thus, we can give a convex formulation of (12) for $K>=1$, by rewriting it in terms of the variational model

$$\min_{u_i}\left\{\sum_{i=1}^{K}\int_\Omega g\|\nabla u_i(\mathbf{p})\|_1 d\mathbf{p}+\gamma\,\|u_i(\mathbf{p})\|_\infty+\sum_{i=1}^{K}\int_\Omega u_i(\mathbf{p})f_i(\mathbf{p})d\mathbf{p}\right\} \tag{13}$$

where the characteristic function of pixel $\mathbf{p}$ in each region $u_i(\mathbf{p})$ satisfies that $\sum_{i=1}^{K}u_i(\mathbf{p})=1, u_i(\mathbf{p})\geq 0$, $\forall i=1,\ldots,K$. Using Fenchel duality, the infinity norm $\|u_i(\mathbf{p})\|_\infty$ becomes $<u_i(\mathbf{p}),\zeta_i(\mathbf{p})>$, s.t. $\|\zeta_i(\mathbf{p})\|_1\leq 1$ with the dual variable $\zeta_i(\mathbf{p})\in\mathbb{R}$, and the L_1 norm $\|\nabla u_i(\mathbf{p})\|_1$ becomes $<\boldsymbol{\xi}_i(\mathbf{p}),\nabla u_i(\mathbf{p})>\ =\ -<u_i(\mathbf{p}),\mathrm{div}\boldsymbol{\xi}_i(\mathbf{p})>$, s.t. $\|\boldsymbol{\xi}_i(\mathbf{p})\|_1\leq 1$ with the dual variable $\boldsymbol{\xi}_i(\mathbf{p})\in\mathbb{R}^2$. In the primal-dual formulation the multiplication with the edge function g can be easily integrated by replacing the constraint $\|\boldsymbol{\xi}_i(\mathbf{p})\|_1\leq 1$ with $\|\boldsymbol{\xi}_i(\mathbf{p})\|_1\leq g$. Then, we can obtain a general formulation of the Total Variation of $u_i(\mathbf{p})$ with

$$\sum_{i=1}^{K}\int_\Omega g\|\nabla u_i(\mathbf{p})\|_1 d\mathbf{p}=\sup_{\xi\in B_0}\left\{\sum_{i=1}^{K}-\int_\Omega u_i(\mathbf{p})\mathrm{div}\xi_i(\mathbf{p})d\mathbf{p}\right\} \tag{14}$$

where the dual variables $\boldsymbol{\xi}=(\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_K):\Omega\rightarrow R^{2K}$ are constrained to lie in the convex set B_0 defined as

$$B_0=\left\{\boldsymbol{\xi}=(\boldsymbol{\xi}_1,\ldots,\boldsymbol{\xi}_K):\Omega\rightarrow\mathbb{R}^{2K},\|\boldsymbol{\xi}_i(\mathbf{p})\|_1\leq g,\forall\mathbf{p}\in\Omega\right\} \tag{15}$$

Finally, we arrive at the following primal-dual saddle point formulation of Potts Model in (12)

$$\min_{u_i}\max_{\boldsymbol{\xi}_i,\zeta_i}\sum_{i=1}^{K}\left\{-<u_i,\mathrm{div}\boldsymbol{\xi}_i>+\gamma<u_i,\zeta_i>+<u_i,f_i>\right\}$$
$$s.t.\quad\begin{cases}\sum_{i=1}^{K}u_i(\mathbf{p})=1,u_i(\mathbf{p})\geq 0,\quad\forall i=1,\ldots,K\\ \|\boldsymbol{\xi}_i(\mathbf{p})\|_1\leq g\\ \|\zeta_i(\mathbf{p})\|_1\leq 1\end{cases} \tag{16}$$

The saddle point problem (16) can be directly solved by using the primal-dual Arrow-Hurwicz algorithm in [8]. Basically the algorithm consists of alternating a gradient descend in the primal variable and a gradient ascend in the dual variable. After each update step the primal and dual variables are re-projected to the respective sets. The projections of ζ_i onto $Z=\{\zeta_i|\|\zeta_i\|_1\leqslant 1\}$ and u_i onto $U=\left\{u_i|\sum_{i=1}^{K}u_i=1,\ u_i\geqslant 0\right\}$ are easy, which can be performed by simple point-wise truncation operations. The projection of $\boldsymbol{\xi}_i$ onto $\Xi=\{\boldsymbol{\xi}_i|\|\boldsymbol{\xi}_i\|_1\leqslant g\}$ is more complicated, since it involves constraints over several levels, in order to perform the projection, we use the iterative projection algorithm of Dykstra [48]. Let Π_U, Π_Ξ and Π_Z

Algorithm 1 Segmentation Algorithm

Initialize $\boldsymbol{\xi}^0 = \mathbf{0}$, $\boldsymbol{\zeta}^0 = \mathbf{0}$ and $\mathbf{u}^0 = \mathbf{0}$.
for $n = 1$ to N **do**
  for $i = 1$ to K **do**
    $\boldsymbol{\xi}_i^{n-\frac{1}{2}} = \boldsymbol{\xi}_i^{n-1} + \eta \nabla u_i^{n-1}$
    $\boldsymbol{\xi}_i^{n} = \Pi_{\Xi}(\boldsymbol{\xi}_i^{n-\frac{1}{2}}) = \frac{\boldsymbol{\xi}_i^{n-\frac{1}{2}}}{\max\left(g, \left\|\boldsymbol{\xi}_i^{n-\frac{1}{2}}\right\|\right)}$
    $\zeta_i^{n-\frac{1}{2}} = \zeta_i^{n-1} + \kappa\gamma u_i^{n-1}$
    $\zeta_i^{n} = \Pi_{Z}(\zeta_i^{n-\frac{1}{2}}) = \max\left(0, \zeta_i^{n-\frac{1}{2}}\right)$
  end for
  for $i = 1$ to K **do**
    $u_i^{n-\frac{1}{2}} = u_i^{n-1} + \delta(\mathrm{div}\boldsymbol{\xi}_i^n - \gamma\zeta_i^n - f_i^n)$
    $u_i^{n} = \Pi_{U}(u_i^{n-\frac{1}{2}}) = \max(0, u_i^{n-\frac{1}{2}})$
    $\bar{u}_i^n = 2u_i^n - \bar{u}_i^{n-1}$
  end for
end for

Algorithm 2 Stereo Matching Algorithm

Initialize $S(\mathbf{p}) = \{S_N(\mathbf{p}), S_{rnd\Omega}(\mathbf{p}), S_{view}(\mathbf{p})\}$ randomly.
repeat
  ♢ **Optimize labeling** s **for current local label sets:**
  $s^{(t)} = \text{argmin } E(d, \mathbf{n})$ with local label sets $S(\mathbf{p})$
  ♢ **Refine local label sets** $S(\mathbf{p})$**:**
  for all pixels $\mathbf{q} \in N(\mathbf{p})$ **do**
    $S_N(\mathbf{p}) \leftarrow \{s_{\mathbf{q}}^{(t)}\}$
  end for
  for all regions $r \in \Omega$ **do**
    $S_{rnd\Omega}(\mathbf{p}) \leftarrow$ four completely randomly samples from $\{\mathbf{q} | \mathbf{q} \in r, \mathbf{p} \in r\}$ in horizontal and vertical line respectively
  end for
  for all pixels $\mathbf{q} \in N_{view}(\mathbf{p})$ **do**
    $S_{view}(\mathbf{p}) \leftarrow \{s_{\mathbf{q}}^{(t)}\}$
  end for
until convergence

stand for the projection operators to the respective convex sets U, Ξ and Z, the iterative optimization algorithm for the object class segmentation subproblem can be summarized in Algorithm 1: firstly, we fix the step parameters $\eta > 0$, $\kappa > 0$ and $\delta > 0$; secondly, let N be the maximum iteration number and K represent the segments number, we initialize $\boldsymbol{\xi}^0 = \mathbf{0}$, $\boldsymbol{\zeta}^0 = \mathbf{0}$ and $\mathbf{u}^0 = \mathbf{0}$; finally, in each iteration $n \in N$, for all $i \in K$ we update the dual variables $\boldsymbol{\xi}_i$, ζ_i and primal variable u_i respectively by projected gradient schemes. We perform the object map estimation in both images of the stereo pair. Both object maps are used to perform consistent segmentation constraint in a symmetric process, that means once pixels belong to the same object in one image, their corresponding pixels in the other image also belong to the same object. Therefore, the consistent segmentation results of the stereo image pair are always assumed to be the initial segments of the corresponding images in each primal-dual iteration. For more details on the primal-dual Arrow-Hurwicz algorithm, we refer the interested readers to [8] and [48].

B. Optimizing for Disparity d and Normal ***n***

For fixed segmentation $\boldsymbol{\Omega}_i$ in (11), we can get the second subproblem, patch-based stereo matching. Instead of performing an exhaustive search (random initialization, spatial propagation, view propagation and temporal propagation) as done in [15], we employ a variant of the PatchMatch Stereo algorithm. Our algorithm is based on minimizing an energy of the form as

$$E(d, \mathbf{n}) = E_{data}(d, \mathbf{n}) + E_{regularize}(d, \mathbf{n}) \tag{17}$$

where it is consisting of a data term describing the similarity between pointwise matches in the stereo pair

$$E_{data}(d, \mathbf{n}) = \sum_{i=1}^{K} \left\{ \tau \int_{\Omega_i} m(\mathbf{p}, (d, \mathbf{n})) d\mathbf{p} \right\} \tag{18}$$

and a regularization term favoring unified depth and normal variations within the same object region

$$E_{regularize}(d, \mathbf{n}) = \sum_{i=1}^{K} \left\{ \lambda \int_{\Omega_i} (-log(\pi_i(d_{\Omega_i}(\mathbf{p}) - d(\mathbf{p})))) d\mathbf{p} \right\} \tag{19}$$

In order to keep efficiently optimizing continuous energy function, we introduce local sampling labels. The local sampling labels are the combination of pixel labels and object region labels, in which label spaces are shared among neighbors, and they enable per-pixel estimation of continuous solutions as well as fast propagations. Pixel labels are a small number of candidate discrete disparity labels and normal labels defined at each pixel p, which contains $N \times N$ patch of samples centered around $\mathbf{p}$ from the previous iteration (in this paper we set $N = 3$), and we refer to as a pixel label set $S_N(\mathbf{p})$. In addition, we define four completely randomly chosen object region samples $S_{rnd\Omega}(\mathbf{p})$ in horizontal and vertical line that give additional candidate labels for accelerating spatial propagation and avoiding stuck at a local minima. Formally, we define the set of samples as

$$S(\mathbf{p}) = S_N(\mathbf{p}) \bigcup S_{rnd\Omega}(\mathbf{p}) \bigcup S_{view}(\mathbf{p}) \tag{20}$$

where the set $S_{view}(\mathbf{p})$ contains the propagated disparity from the other view. Our method chooses the best candidate label s from the set $S(\mathbf{p})$ that are shared for the pixel $\mathbf{p}$. By sharing local label sets among neighbors, good candidate labels are spatially propagated to nearby pixels.

For efficient and accurate optimization, we use the standard graph cuts approaches [21], [22] in our algorithm. The overview of our optimization procedure is summarized in Algorithm 2, where $N_{view}(\mathbf{p})$ represents the corresponding point set of $\mathbf{p}$ in the other view. Our optimization uses an iterative framework. Firstly, we assign each pixel of both views to a random plane by selecting a random pixel disparity d that lies in the range of allowed continuous disparity values,

in general, the allowed disparity range is provided by the relevant benchmark data set (e.g., [0, 120] will be suitable for Middlebury stereo benchmark [31]), and for more details on the calculation of this disparity range, we refer the interested readers to [49]. Secondly, we assign the normal $\mathbf{n}$ of the plane randomly in spherical coordinate as $\mathbf{n} = [n_x, n_y, n_z]^T = [cos(\theta)sin(\phi), sin(\theta)sin(\phi), cos(\phi)]^T$, where θ is a random angle in the range [0°, 360°], and ϕ is a random angle in the range [0°, 60°]. These range settings come from a simple assumption that a plane is visible in the reference image when the angle between the plane normal $\mathbf{n}$ and the z axis of the reference image's coordinate system is below a certain threshold, here we set this threshold as 60° in our optimization for fast convergence in most cases, and it is certain that a truly random rotation would be the best choice. Finally, we alternately optimize the labeling $s = \{d, \mathbf{n}\}$ with given local sets $S(\mathbf{p})$, and refine the local label sets $S(\mathbf{p})$ locally with the labeling s fixed. We perform the disparity map estimation in both images of the stereo pair. Both disparity maps are used to perform the view propagation of samples and also left-right consistency checking, which allows the removal of inconsistent disparity results. After each PatchMatch iteration, occluded areas with inconsistent disparity are estimated by the closest distance valid points of the MST in the same object region (discussed in Section IV-C). The occlusion-estimation is also performed for the final result and is the only post-processing step we perform.

C. MST Based Occlusion Handling

According to tree filtering technique [50], a great advantage of using Minimum Spanning Tree (MST) [51] is that it gives a more natural image pixel similarity measurement metric. It is more accurate than the previous methods so that every pixel in the image can correctly contribute to all the other pixels during the whole stereo matching process. Thus, in order to perform an efficient non-local occlusion handling strategy, the input stereo image pairs are represented as connected, undirected MST $G = (V, E)$, where each node in V corresponds to a pixel in corresponding image, and each edge in E connects a pair of neighboring pixels.

However, due to that current tree filtering technique for occlusion handling [50] still have to step through the entire cost volume exhaustively, which makes the solution speed scale linearly with the label space size. When the label space is huge, which is often the case for subpixel accurate stereo estimation, their computational complexity becomes quickly unacceptable. Therefore, a spatially regularized labeling space is favored, and we novelly decompose the global MST into subtrees for each segment following the segment-tree algorithm [52], and take segmentation as the basic primitive which also effectively denotes the label search range. More importantly, this segment-based strategy creates desired chances for computation reduce and speedup.

Given the left-right consistency checking disparity results, we extract a set of candidate labels $C(\mathbf{p})$ for invalid pixel $\mathbf{p}$. This candidate set consists of the labels of the closest valid pixels for the current invalid target pixel. Instead of choosing the candidates labels from the left or right pixels in the same row of the current invalid pixel [15], our closest candidates pixels are defined on each segmentation subtree, which can be achieved as follows: firstly, for each invalid pixel $\mathbf{p}$, we suppose $\mathbf{p}$ as the root of the subtree T; secondly, we try to find $\mathbf{p}$'s *closest and valid* nodes in every branch of T. These are candidates used to recover $\mathbf{p}$'s final disparity values.

When all of the closest candidates are located, we will use them to recover the invalid regions. Instead of giving each of the candidates different weights and calculating the weighted average, we treat each candidate equally to select the new labels for invalid pixels. This is mainly because that the weighted average usually bring about the edge blurring. Our strategy is implemented as follows:

Firstly, considering that it is compact for each pixel in the same region, and the invalid target pixel share the same label among neighboring candidate set. For the current target invalid pixel $\mathbf{p}$, once there are more than half of the candidates in $C(\mathbf{p})$ give the same label value, then the target pixel $\mathbf{p}$ can be assigned with this value directly for the integer-disparity estimation. While for our patch-based continuous stereo estimation method, we follow the slant plane assumption of PatchMatch method [15], given each valid candidate $\mathbf{q}$ in $C(\mathbf{p})$, the recover disparity value $d'_{\mathbf{p}}$ is required to be calculated with $d'_{\mathbf{p}} = a_{\mathbf{q}}\mathbf{p}_x + b_{\mathbf{q}}\mathbf{p}_y + c_{\mathbf{q}}$, where $a_{\mathbf{q}}$, $b_{\mathbf{q}}$, $c_{\mathbf{q}}$ represents the slant plane parameters of pixel $\mathbf{q}$. When more than half of these candidate values drop into the same value range, the minimum value of these values are assigned to the target invalid pixel $\mathbf{p}$.

Secondly, when the above state fails, the lowest disparity value given by these candidates is chosen to recover the target invalid pixel $\mathbf{p}$. Selecting the lowest one is motivated by the fact that occlusion occurs at the background, which plays a key role in preventing the foreground from eroding the occlusion region.

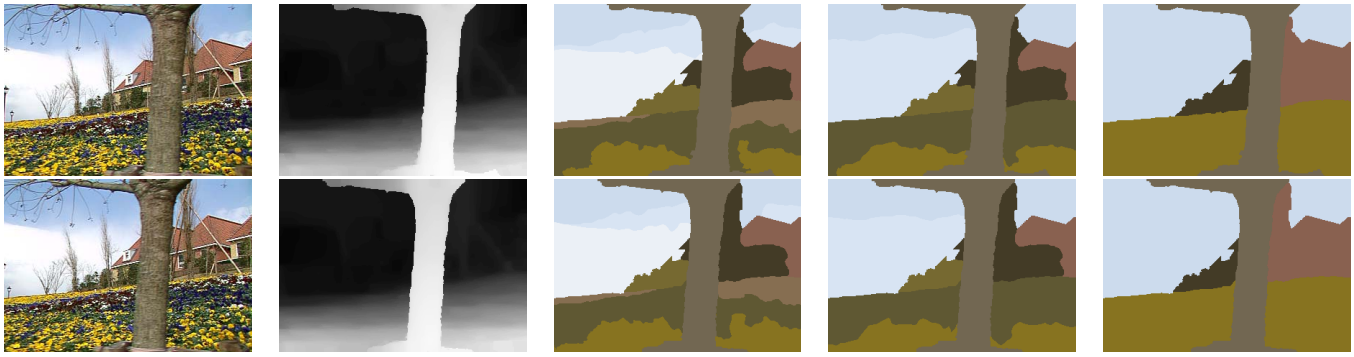
Unlike most occlusion handling methods [15], our strategy will not lead to the horizontal streaks results, thus a time-consuming mean filter procedure is not needed which sometimes bring the structure destroying and edge blurring.

V. Experimental Results

In this section, we first evaluate our method on the commonly-accepted Middlebury stereo benchmark [31]. We further compare with the original PatchMatch algorithm [15] and its variants (PM-Huber [17] and PMBP [37]) in sub-pixel accurate disparity estimation, especial with the ObjectStereo method [16] that is closely related to our approach. Finally, we evaluate the proposed algorithm on two real-world data sets (KITTI benchmark data sets [53] and multi-view benchmark data sets [54]) to show the robustness of our joint solution and verify the accurate large-scale scene reconstruction quality of our method. All the experiments are implemented on a PC platform with an Intel i7 3.60GHz CPU, 16GB memory and an NVIDIA Geforce GTX 280 GPU.

A. Experimental Settings

In the experiments, there are many parameters for our multi-label Potts model and global PatchMatch



(a) (b) (c) (d) (e)

Fig. 2. The effect of the the label cost prior. From left to right, we give different weights for the label cost term, with increasing γ the number of labels used to explain the scene decreases. In the last column, (e) with increasing K five segments are always obtained as the label cost prior expected, which ensures that the number of used segments is as small as necessary. (a) Stereo image pair. (b) Stereo disparity pair. (c) $\gamma = 30$, $K = 15$. (d) $\gamma = 50$, $K = 15$. (e) $\gamma = 70$, $K = 15, 10, 5$.

TABLE I

OBJECTIVE EVALUATION FOR THE PROPOSED METHOD WITH THE MIDDLEBURY BENCHMARK IN SUB-PIXEL ACCURACY

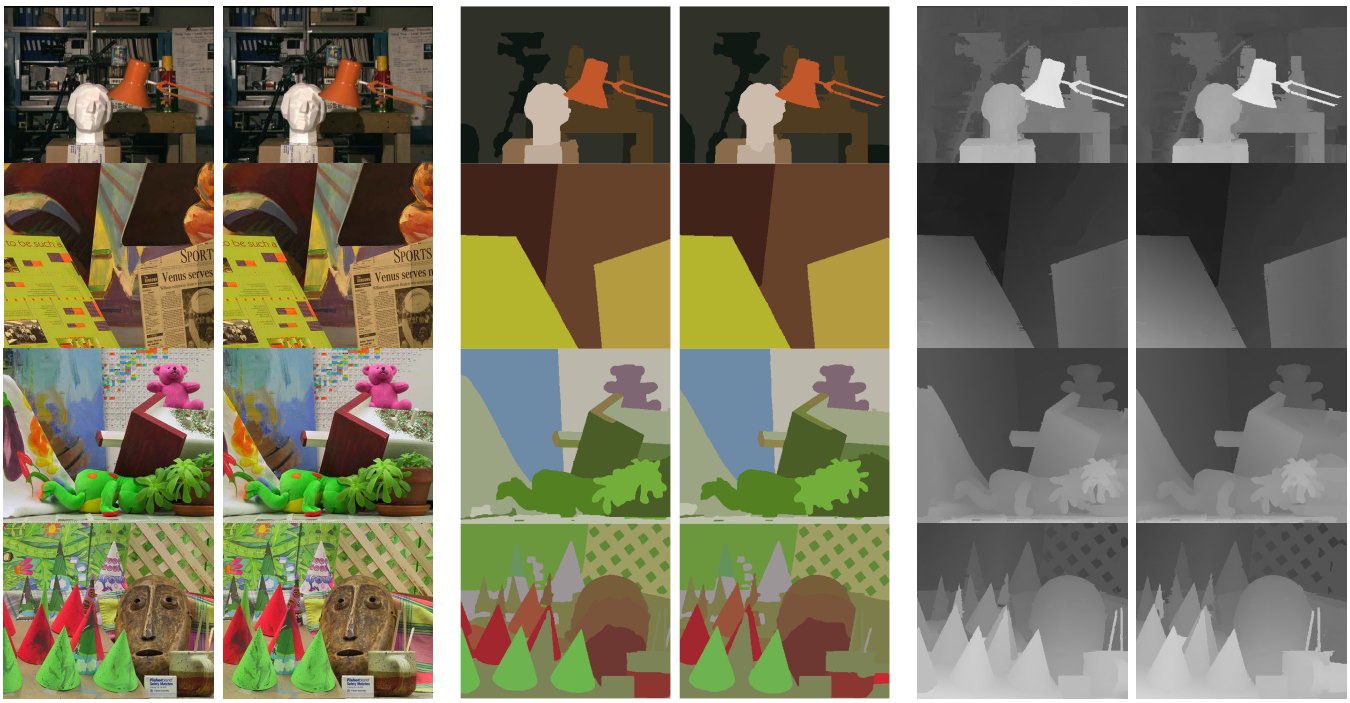
Algorithm	**Avg. Rank**	**Tsukuba**			**Venus**			**Teddy**			**Cones**			**Avg. Error**
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	
Our Method	6.9	7.16_{11}	7.66_{8}	15.1_{21}	0.58_{3}	0.79_{2}	4.67_{3}	5.21_{4}	11.9_{8}	15.9_{6}	3.51_{6}	8.86_{5}	9.58_{6}	7.58%
PM-Huber [17]	7.1	7.12_{9}	7.80_{11}	13.7_{9}	1.00_{11}	1.40_{12}	7.80_{14}	5.53_{6}	9.36_{3}	15.9_{7}	2.70_{1}	7.90_{1}	7.77_{1}	7.33%
PMBP [37]	17.2	11.9_{48}	12.3_{44}	17.8_{54}	0.85_{9}	1.10_{7}	6.45_{9}	5.60_{7}	12.0_{9}	15.5_{4}	3.48_{5}	8.88_{6}	9.41_{5}	8.77%
PatchMatch [15]	25.0	15.0_{68}	15.4_{67}	20.3_{81}	1.00_{12}	1.34_{11}	7.75_{13}	5.66_{8}	11.8_{7}	16.5_{8}	3.80_{8}	10.2_{9}	10.2_{8}	9.91%
ObjectStereo [16]	73.0	16.4_{77}	16.8_{74}	16.1_{31}	2.56_{22}	2.67_{18}	7.69_{12}	19.6_{125}	22.7_{96}	30.3_{83}	16.3_{133}	20.7_{115}	19.7_{90}	16.0%

stereo matching. In order to assess the sensitivity of our algorithm to variations in these parameters, we test 21 image pairs on Middlebury stereo benchmark [31] (including Aloe, Bowling 1 - 2, Baby 1 - 3, Cloth 1 - 4, Flowerpots, Lampshade 1 - 2, Rocks 1 - 2, Wood 1 - 2, Art, Teddy, Cones, Venus) for a variety of parameter settings, and the performance of our method is rather stable to keep the disparity average error rate lower than 15% when the explicit parameters γ, σ, λ and τ in (10) are in the range $[10, 100]$, $[5, 20]$, $[2, 15]$ and $[20, 50]$, the implicit parameters m, ϵ, ν, η, κ, δ, α, μ, τ_{col} and τ_{grad} used in the optimization are in the range $[20, 70]$, $[2, 8]$, $(0, 1)$, $(0, 0.5)$, $(0, 0.5)$, $(0, 0.5)$, $(0, 1)$, $(0, 1)$, $[5, 15]$ and $[2, 8]$. Furthermore, among the parameters to be set, the matching window size m, the matching weight μ, the data term weight τ and the regularization term weight λ seem to have the largest impact on the final results, other parameters do affect the final results slightly, which could be fixed in the whole experiments. For a convenient and fair comparison with other similar methods, we use the same parameters settings throughout the experiments. The explicit parameters in (10) are set to $\{\gamma, \sigma, \lambda, \tau\} = \{30, 10, 10, 30\}$. The implicit parameters of Potts Model optimization are set to $\{\epsilon, \nu, \eta, \kappa, \delta\} = \{2, 0.55, 0.1, 0.1, 0.1\}$. The size of the patch m considered in the data term is 41×41 pixels centered around the pixel $\mathbf{p}$, which is the same setting as PM-Huber [17] and PMBP [37]. For setting the implicit parameters of PatchMatch optimization, we mainly follow the original PatchMatch stereo algorithm [15] and set them to $\{\alpha, \mu, \tau_{col}, \tau_{grad}\} = \{0.9, 0.1, 10, 2\}$.

To observe the effect of the the label cost prior [10] used in (12), we assess the segmentation performance using different settings for the weight γ and the label number K of the label cost term $\sum_{i=1}^{K} \|\mathbb{1}_{\Omega_i}\|_{\infty}$. The effectiveness of the label cost term and the label number is demonstrated in Fig. 2. With increasing γ, less labels are used resulting in a simplified segmentation in Fig. 2(c) - (e). In contrast, with increasing K, our algorithm always partitions the scene into five segments as the label cost prior expected in Fig. 2(e), which ensures that the number of used segments is as small as necessary.

B. Quantitative Evaluation

We use the following data sets: 'Tsukuba', 'Venus', 'Teddy' and 'Cones' to quantitatively evaluate the five method, and select the evaluation rankings on the Middlebury stereo benchmark [31] for sub-pixel accuracy, as Table I shown. The results for each data set are evaluated by measuring the percentage of bad matching pixels (where the absolute disparity error is greater than sub-pixel). The measurement is computed for three subsets of an image: nonocc (the pixels in the nonoccluded regions), all (the pixels in both the nonoccluded and half-occluded regions), and disc (the visible pixels near the occluded regions). In this benchmark, our method currently (January 2015) achieves the second rank amongst more than 150 stereo methods, and outperforms all the relevant methods with lowest average error rate (7.58%) and the highest average ranking (6.9). Our method consistently outperforms PMBP [37], PatchMatch [15] and ObjectStereo [16] in the four evaluations. Although results of PM-Huber [17] for 'Cones' data set are slightly better than ours, our method works particularly well on the 'Venus' data set, which is the top performer, and our method is compatible with PM-Huber [17]



(a) (b) (c)

Fig. 3. Our results on the standard Middlebury benchmark stereo image pairs (from top to bottom) 'Tsukuba', 'Venus', 'Teddy' and 'Cones': (a) original stereo image pair, (b) final segmentation results, (c) final disparity results.

in the other evaluations. The key of our method is to provide a unified variational formulation for joint object segmentation and stereo matching. We show the final segmentation results and final disparity results for the four Middlebury images in Fig. 3. Note that already our method is very successful in sub-pixel depth estimation when compared against the state-of-the-art of patch-based stereo matching methods. What is more, our method can also accomplish to segment the complex scene into a small number of good objects based on their color and depth values.

For visual comparison, we further evaluate the error maps of the five methods with the Middlebury stereo benchmark [31] in Fig. 4. For these data sets, more detailed evaluation on various regions can be performed. The approach closest to ours is ObjectStereo [16], which also shows that depth estimation can be improved by introducing the notion of objects. Although it can achieve perfect object segmentation results, the final disparity results of ObjectStereo [16] is quantized to integer disparities, thus we consider that not to be appropriate for sub-pixel accurate disparity estimation.

In order to compare disparity quality with PatchMatch [15], PMBP [37] and PM-Huber [17] and show the advantage of the proposed joint solution, we firstly take 'Teddy' data set for example, a particularly interesting case is marked by the blue rectangle, the uniform background pattern misleads these mentioned pixel level algorithms [15], [17], [37] into computing an unreliable pixel matching cost, disparity estimation using only the pixel label set (including randomly chosen sample labels in the whole image view) further propagates this error, leading to patches of incorrectly estimated disparity around teddy bear. In our algorithm, we apply an iterative process to gradually cluster small neighboring segments to object-level segmentation and employ the object-level segmentation as a soft constraint, which can enable the label propagation between small separated segments of the same object and aid disparity estimation to make good candidate labels of the solution without bad local minima, therefore, the integration of the region constrained label set into disparity estimation helps to suppress the errors propagation.

We further challenge our algorithm with additional difficult case to observe the effect of region labels for large textureless regions in Fig. 5, note that sub-pixel errors are tagged in red, and in contrast to the Middlebury benchmark evaluation, which takes no account of the occlusion regions, all the areas (including occlusions) of both views are used to calculate these sub-pixel errors in Fig. 5. After estimating the disparity with pixel label set and region label set respectively in Fig. 5(b) - (c), we can now see the shortcoming of the pixel labels when it alone defines the candidate label set. It can be seen that the pixel label set accurately estimated the disparity values of pixels near color and depth discontinuities in Fig. 5(b). However, in the textureless region (e.g., the arrow point region), the disparity values of neighboring pixels were split among patches with different disparity ranges, which leads to the wrong disparity estimation inside the textureless region. The region label set can outperform pixel label set in this case in Fig. 5(c), as it is able to correctly match from segment constrains within the uniformly color region and more accurately estimate the disparity for this region. Ideally, the region level disparity estimating at a coarser

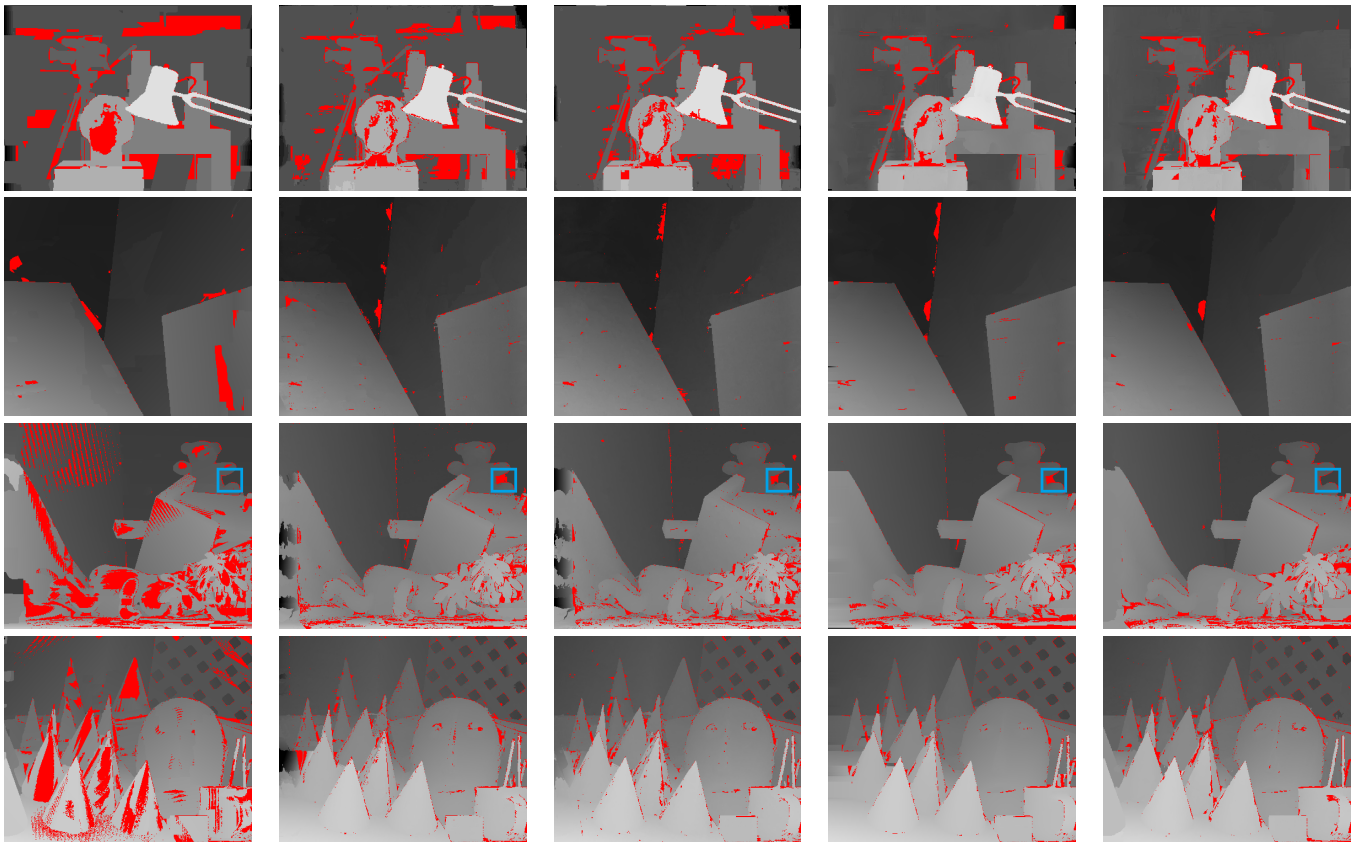


Fig. 4. Visual comparison with other methods for disparity results and sub-pixel error maps on the standard Middlebury benchmark stereo reference image (from top to bottom) 'Tsukuba', 'Venus', 'Teddy' and 'Cones'. Pixels with erroneous disparities are marked in red. (a) Disparity results and error maps computed with ObjectStereo [16]. (b) Disparity results and error maps computed with PatchMatch [15]. (c) Disparity results and error maps computed with PMBP [37]. (d) Disparity results and error maps computed with PM-Huber [17]. (e) Disparity results and error maps computed with our method.

level would act to complement the finer, pixel level disparity estimating and vice versa. In a textureless region, which has no disparity discontinuities, the region label set should dominate the resulting disparity estimation. Then, in a region of rich texture, the algorithm should rely more on the finer, pixel label set. In our algorithm, we use the combination label set to accurately estimate the disparity of pixels in Fig. 5(d). It is worth of noting that the proposed joint optimization enable us to apply an iterative process to gradually merge neighboring segments from over-segmentation to object-level segmentation, which makes the label propagation between separated segments of the same object possible and brings a coarse-to-fine disparity estimating with the combination label set effectively. As for pixel level algorithms, PMBP [37] and PM-Huber [17] produce competitive disparity estimation with special explicit smoothing model to the original PatchMatch algorithm [15], however, they still have difficulty with some large, textureless regions of uniform color, and fail to provide the correct disparity estimation in Fig. 5(e) - (f).

Apart from using standard images, it is important to test our method on real-world stereo pairs. Firstly, we evaluate our approach on the KITTI data set [53], which is the only real-world stereo data set with accurate ground truth. In contrast to the Middlebury benchmark [31] above, where the disparity search range is very small, the environment highly textured and the illumination conditions nearly constant, the KITTI data set [53] consists of 195 very challenging stereo images with 1226×370 resolution (4.5 Megapixel) captured from an autonomous driving platform driving around in outdoor scenarios, together with ground truth obtained by laser scanning. To alleviate sensor noise and reduce the illumination changes, we apply 3×3 median filtering for original stereo image pair before our disparity estimation. Our method achieves an average error rate of 4.15% on the KITTI stereo test set and is currently (January 2015) ranked in top two with 2-pixels precision on the online leaderboard. Table III compares the average error rates of the best performing stereo algorithms on this data set. Fig. 7 shows the sample results of disparity estimation on two stereo image pairs, where different objects are present in the scene in Fig. 7(b), note that 2-pixels errors are tagged in red, and in contrast to the Middlebury benchmark evaluation, which takes no account of the occlusion regions, all the areas (including occlusions) of both views are used to calculate these 2-pixels errors in Fig. 7. We compare the results of the proposed algorithm with the standalone disparity results with pixel label set in Fig. 7(c) and Fig. 7(d), which shows that our method achieves greater accuracy around the edge and large textureless regions (e.g., the surface of

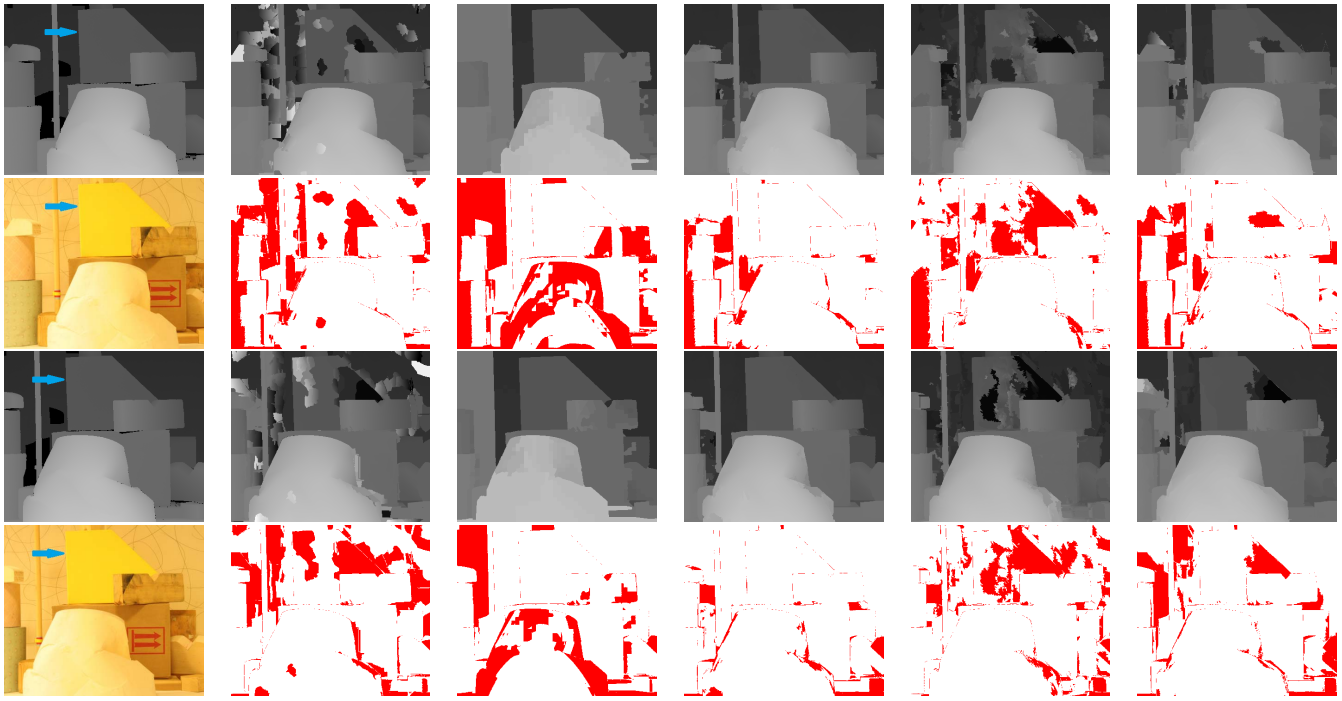


Fig. 5. Visual effects of each label subset, and visual comparison with PMBP [37] and PM-Huber [17] on challenging case for large textureless regions, sub-pixel errors are tagged in red. (a) Ground truth, (b) disparity estimation results with pixel label subset of our method, (c) disparity estimation results with region label subset of our method, (d) disparity estimation results with combination label set of our method, (e) disparity estimation results with PMBP [37], (f) disparity estimation results with PM-Huber [17].

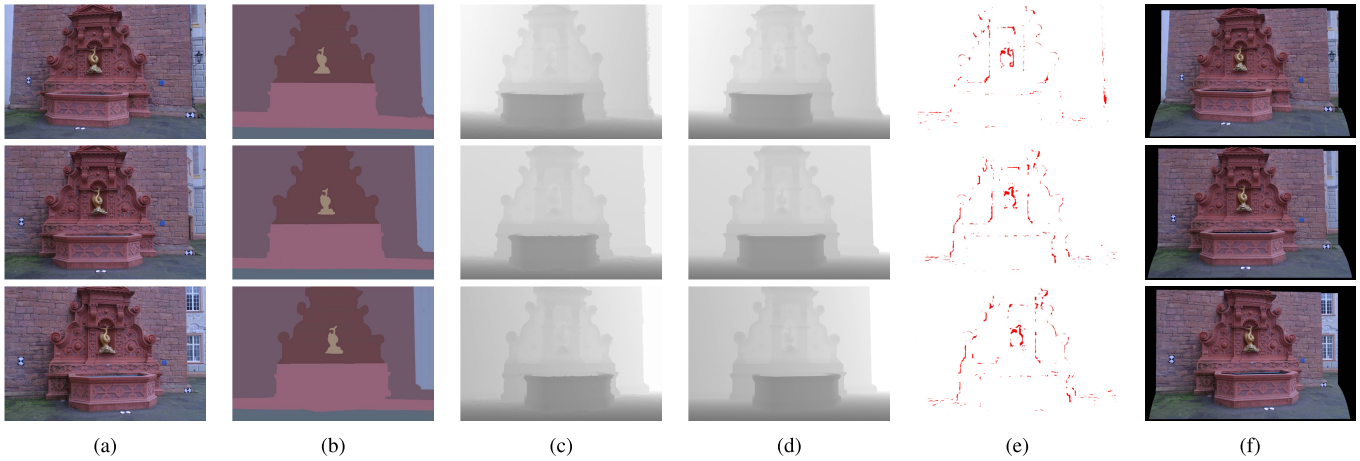
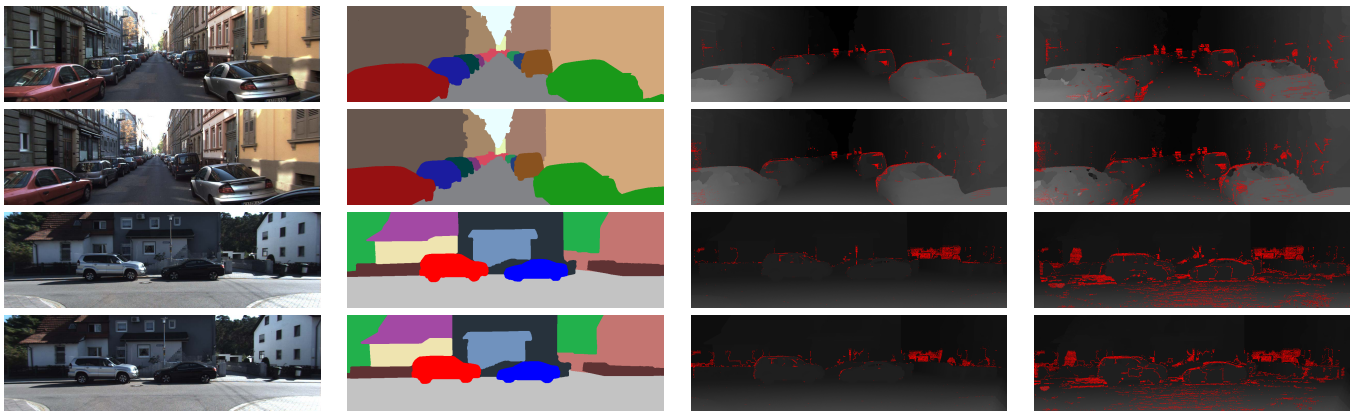


Fig. 6. Multi-view images and their final segmentation results and final depth results. All the sample images are from the multi-view benchmark data sets, Fountain-P11, provided by Strecha et al. [54]. Our depth-maps are well suited for the creation of point clouds without discretization or staircasing artifacts. (a) Multi-view images. (b) Segmentation results. (c) Depth results. (d) Ground truths. (e) Sub-pixel error maps. (f) Point cloud results.

car and road). Secondly, the multi-view benchmark data set, Fountain-P11, provided by Strecha et al. [54] is also used, Fountain-P11 has 11 images with 3072×2048 resolution (6 Megapixel). Fig. 6(a) shows some sample images in the data set. The raw ground truth is obtained by laser scanning, which is a single high resolution triangle mesh model. To make the comparison of depth-map feasible, we project the raw ground truth to each image to generate ground truth depth-maps in Fig. 6(d). Our algorithm can correctly reconstruct the object map, depth-map and point cloud without discretization or staircasing artifacts in Fig. 6(b), Fig. 6(c) and Fig. 6(f). Fig. 6(e) shows the sub-pixel depth error maps, which indicates that our method can generate accurate dense points with a small amount of errors. Note that the random initialization process of our method is very likely to have at least one good guess for each depth plane in the image, especially for high resolution images which contains more pixels, that means more guesses for depth plane than low resolution ones. That is why our algorithm is suitable for accurate large-scale scene reconstruction for high resolution images in object level.



(a) (b) (c) (d)

Fig. 7. Object level image segmentation and disparity estimating results on the KITTI data set [53], 2-pixels errors are tagged in red. (a) Original stereo image pair, (b) final object level segmentation results, (c) final joint disparity results, from top to bottom, the error rates are 3.52%, 4.93%, 4.09% and 4.55% with 2-pixels precision. (d) Standalone disparity results with pixel label set, from top to bottom, the error rates are 10.47%, 12.16%, 14.92% and 15.83% with 2-pixels precision.

C. Complexity and Performance Analysis

We show the complexity and performance analysis of the relevant methods using their CPU or GPU implementations. Firstly, the fusion approach QPBO-GC [57] used in ObjectStereo [16] are complicated and expensive, therefore, we only gave a CPU implementation for the fusion move problem without connectivity prior in integer-valued disparities precision. Let N be the number of the image pixels and L represent the label space size, the above minimization can be computed in $O(NL)$ time, which is extremely high complexity for continuous disparity space.

Secondly, the parallelization schemes of belief propagation [1], [38] usually propagate candidate labels to distant pixels, which is not applicable to PMBP [37] that must propagate messages to neighbors. And efficient parallelization implementations for PMBP [37] are not available in literature due to the overhead of data transfer, we also implemented the unary cost computation on CPU. According to the default settings of PMBP [37], they defined five candidate labels for each pixel, and computed the pixel's cost aggregation with perturbation for all the candidate labels (called "random search" in the original PatchMatch stereo algorithm [15]) using the plane parameters of its eight neighbor pixels for spatial propagation and one corresponding pixel in the other view for view propagation, which give the number of disparities to be tested for each pixel is $D_1 = (1 + (8 + 1) \times 5) \times 2 = 92$, consequently, the disparity computation complexity of PMBP [37] is $O(mD_1NlogL)$, where m is the size of the matching window.

Finally, our algorithm has been designed to be executed on massively parallel architectures. Our object segmentation subproblem can be solved very efficiently on parallel architectures. Given the segments number K, the object segmentation runs in linear complexity $O(MK) \approx O(M)$, where $M = O(N)$ is the number of the neighboring pixel pair operations. Also our PatchMatch sampling strategy is completely parallel in contrast to the original PatchMatch stereo algorithm. Therefore, we choose to implement our algorithm, PatchMatch [15] and PM-Huber [17] with CUDA on GPU, which search the best candidate label for each pixel and share similar local sampling strategy: the number of our sample sets, including eight neighbor pixels, four completely random pixels in object region and one corresponding pixel in the other view, is $D_2 = 1 + 8 + 4 + 1 = 14$; the number of PatchMatch [15] sample sets, including two neighbor pixels, one corresponding pixel in the other view and the perturbed pixels, is $D_3 = (1 + 2 + 1) \times 2 = 8$; the number of PM-Huber [17] sample sets, including eight neighbor pixels, two completely random pixels, one corresponding pixel in the other view and the perturbed pixels, is $D_4 = (1 + 8 + 2 + 1) \times 2 = 24$. Therefore, the disparity computation complexities of our algorithm, PatchMatch [15] and PM-Huber [17] are $O(mD_2NlogL)$, $O(mD_3NlogL)$ and $O(mD_4NlogL)$ respectively, where m is the size of the matching window. Furthermore, with the linear MST algorithm proposed by Karger et al. [51], the worst-case complexity of our occlusion handling is $O(N + N_{occ}L_{occ}K)$, where N_{occ} and L_{occ} are the number of invalid pixels and the number of candidate reliable labels in each segment region respectively, and these two variables are much lower than the number of the image pixels N.

We evaluate the computation performance of the five algorithms. For each method, we report its average processing time (in seconds) under different Megapixels settings shown in Table II. The runtime of the five algorithms highly varies with the parameter settings and number of iterations. There is no doubt that PatchMatch [15] is the fastest algorithm. Different settings for our algorithm allow the estimation of disparity maps in a few seconds, which runs as efficiently as PM-Huber [17], and it is 10 to 20 times faster than PMBP [37] and ObjectStereo [16] while attaining better accuracy. It is worth of noting that the computation performance in Table II for processing images grows nonlinearly, that is because our current GPU implementation is completely unoptimized and several obvious performance

TABLE II
PERFORMANCE EVALUATION OF THE FIVE METHODS ON DIFFERENT MEGAPIXELS

Megapixels	Average Processing Time (in seconds)				
	ObjectStereo [16]	PatchMatch [15] (GPU)	PMBP [37]	PM-Huber [17] (GPU)	Our method (GPU)
0.1	646.4	1.8	375.2	51.7	33.9
0.2	1415.5	2.4	647.2	125.6	78.9
0.3	2887.8	3.5	1157.7	263.4	198.3

TABLE III
OBJECTIVE EVALUATION FOR THE PROPOSED METHOD WITH THE KITTI DATA SET IN 2-PIXELS ACCURACY

Rank	Method		Error
1	CVPR 1186	Anonymous submission	3.90 %
2	PM-PM	Our method	4.15 %
3	SPS-StFl	Yamaguchi et al. [55]	4.30 %
4	MC-CNN	Anonymous submission	4.34 %
5	VC-SF	Vogel et al. [56]	4.56 %

enhancements have not been exploited yet, such as the parallel sampling strategy, which has not been optimized to evaluate the computation performance. Furthermore, the advanced PatchMatch Filter [36] may also come into help in reducing the complexity to $O(D_2 N log L)$, which removes the complexity dependency on the matching window size m.

VI. CONCLUSION

We presented a variational approach for joint stereo matching and object segmentation based on a continuous Potts like model with label costs. In our approach, our only assumption is that the scene is assembled of compact objects, which introduces a object segmentation into the depth estimation process, what is more, we propose a planar bias model to illustrate that objects may have a bias towards being planar in 3D space, which allows to extend depth in the same object region and to precisely capture depth discontinuities planar. Therefore, our approach can be used for multiple layers resulting in a quite dense depth-map at each image in object level, which takes both *accuracy* and *efficiency* into account. By comparing with the original PatchMatch algorithm [15] and its variants ObjectStereo [16], PMBP [37] and PM-Huber [17], our method shows an advantage in average accuracy and comparable or greater efficiency. What is more, the proposed algorithm could produce consistently good results for various data sets (Middlebury benchmark data sets [31], KITTI benchmark data sets [53] and multi-view benchmark data sets [54]).

In further research, we will incorporate more complex occlusion handling schemes into our framework, which may yield even greater accuracy; and we will enhance our algorithm implementation to improve the computation performance. Furthermore, we will concentrate on improving the algorithm for accurate multiple view reconstruction, including stereo pair selection, depth-map computation, depth-map refinement, and depth-map merging.

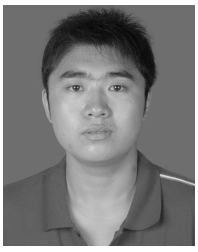
ACKNOWLEDGMENT

The authors would like to thank Dr. Shuhan Shen and Longquan Dai for their help with the implementations and also for fruitful discussions.

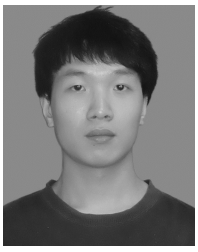
REFERENCES

[1] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *Proc. 18th Int. Conf. Pattern Recognit.*, 2006, pp. 15–18.
[2] C. L. Zitnick and S. B. Kang, "Stereo for image-based rendering using image over-segmentation," *Int. J. Comput. Vis.*, vol. 75, no. 1, pp. 49–65, 2007.
[3] Y. Taguchi, B. Wilburn, and C. L. Zitnick, "Stereo reconstruction with mixed pixels using adaptive over-segmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[4] L. Hong and G. Chen, "Segment-based stereo matching using graph cuts," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1. Jun. 2004, pp. I-74–I-81.
[5] H. Tao, H. S. Sawhney, and R. Kumar, "A global matching framework for stereo computation," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 1. Jul. 2001, pp. 532–539.
[6] Z.-F. Wang and Z.-G. Zheng, "A region based stereo matching algorithm using cooperative optimization," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[7] T. Pock, A. Chambolle, D. Cremers, and H. Bischof, "A convex relaxation approach for computing minimal partitions," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 810–817.
[8] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *J. Math. Imag. Vis.*, vol. 40, no. 1, pp. 120–145, 2011.
[9] R. B. Potts, "Some generalized order-disorder transformations," *Math. Proc. Cambridge Philos. Soc.*, vol. 48, no. 1, pp. 106–109, 1952.
[10] J. Yuan and Y. Boykov, "TV-based multi-label image segmentation with label cost prior," in *Proc. Brit. Mach. Vis. Conf.*, 2010, pp. 101.1–101.12.
[11] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann, "Local stereo matching using geodesic support weights," in *Proc. 16th IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 2093–2096.
[12] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 3017–3024.
[13] C. Richardt, D. Orr, I. Davies, A. Criminisi, and N. A. Dodgson, "Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid," in *Proc. 11th Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 510–523.
[14] K.-J. Yoon and I.-S. Kweon, "Locally adaptive support-weight approach for visual correspondence search," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2. Jun. 2005, pp. 924–931.
[15] M. Bleyer, C. Rhemann, and C. Rother, "PatchMatch stereo—Stereo matching with slanted support windows," in *Proc. Brit. Mach. Vis. Conf.*, 2011, pp. 1–11.
[16] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha, "Object stereo—Joint stereo matching and object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 3081–3088.
[17] P. Heise, S. Klose, B. Jensen, and A. Knoll, "PM-Huber: PatchMatch with Huber regularization for stereo matching," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 2360–2367.
[18] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, vol. 1. 1967, pp. 281–297.
[19] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
[20] J. Shi and J. Malik, "Normalized cuts and image segmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1997, pp. 731–737.
[21] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov. 2001.

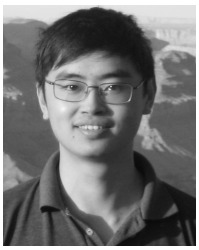
[22] D. M. Greig, B. T. Porteous, and A. H. Seheult, "Exact maximum *a posteriori* estimation for binary images," *J. Roy. Statist. Soc.*, vol. 51, no. 2, pp. 271–279, 1989.
[23] D. Cremers, M. Rousson, and R. Deriche, "A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape," *Int. J. Comput. Vis.*, vol. 72, no. 2, pp. 195–215, 2007.
[24] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.
[25] T. F. Chan, S. Esedoglu, and M. Nikolova, "Algorithms for finding global minimizers of image segmentation and denoising models," *SIAM J. Appl. Math.*, vol. 66, no. 5, pp. 1632–1648, 2004.
[26] C. Nieuwenhuis, E. Töppe, and D. Cremers, "A survey and comparison of discrete and continuous multi-label optimization approaches for the Potts model," *Int. J. Comput. Vis.*, vol. 104, no. 3, pp. 223–240, 2013.
[27] H. Ishikawa, "Exact optimization for Markov random fields with convex priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 10, pp. 1333–1336, Oct. 2003.
[28] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime TV-L^1 optical flow," in *Proc. 29th DAGM Conf. Pattern Recognit.*, 2007, pp. 214–223.
[29] J. Lellmann, J. Kappes, J. Yuan, F. Becker, and C. Schnörr, "Convex multi-class image labeling by simplex-constrained total variation," in *Proc. 2nd Int. Conf. Scale Space Variational Methods Comput. Vis.*, 2009, pp. 150–162.
[30] C. Zach, D. Gallup, J.-M. Frahm, and M. Niethammer, "Fast global labeling for real-time stereo using multiple plane sweeps," in *Proc. Vis., Modeling Vis. Workshop*, 2008, pp. 243–252.
[31] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vis.*, vol. 47, nos. 1–3, pp. 7–42, 2002.
[32] C. Lei, J. Selzer, and Y.-H. Yang, "Region-tree based stereo using dynamic programming optimization," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2006, pp. 2378–2385.
[33] Q. Yang, "A non-local cost aggregation method for stereo matching," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1402–1409.
[34] K.-J. Yoon and I. S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 650–656, Apr. 2006.
[35] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, "Stereo matching with color-weighted correlation, hierarchical belief propagation, and occlusion handling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 492–504, Mar. 2009.
[36] J. Lu, H. Yang, D. Min, and M. N. Do, "Patch match filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1854–1861.
[37] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz, "PMBP: PatchMatch belief propagation for correspondence field estimation," *Int. J. Comput. Vis.*, vol. 110, no. 1, pp. 2–13, Oct. 2014.
[38] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient belief propagation for early vision," *Int. J. Comput. Vis.*, vol. 70, no. 1, pp. 41–54, 2006.
[39] M. Bleyer, C. Rhemann, and C. Rother, "Extracting 3D scene-consistent object proposals and depth from stereo images," in *Computer Vision* (Lecture Notes in Computer Science), vol. 7576, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Germany: Springer-Verlag, 2012, pp. 467–481.
[40] L. Ladicky *et al.*, "Joint optimization for object class segmentation and dense stereo reconstruction," in *Int. J. Comput. Vis.*, vol. 100, no. 2, pp. 122–133, 2012.
[41] C. Häne, C. Zach, A. Cohen, R. Angst, and M. Pollefeys, "Joint 3D scene reconstruction and class segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 97–104.
[42] D. Sun, E. B. Sudderth, and M. J. Black, "Layered image motion with explicit occlusions, temporal consistency, and depth ordering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 2226–2234.
[43] G. Zhang, J. Jia, W. Hua, and H. Bao, "Robust bilayer segmentation and motion/depth estimation with a handheld camera," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 603–617, Mar. 2011.
[44] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.
[45] X. Bresson, S. Esedoglu, P. Vandergheynst, J.-P. Thiran, and S. Osher, "Fast global minimization of the active contour/snake model," *J. Math. Imag. Vis.*, vol. 28, no. 2, pp. 151–167, Jun. 2007.
[46] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
[47] D. Cremers, T. Pock, K. Kolev, and A. Chambolle, "Convex relaxation techniques for segmentation, stereo and multiview reconstruction," in *Advances in Markov Random Fields for Vision and Image Processing*. Cambridge, MA, USA: MIT Press, 2011.
[48] J. P. Boyle and R. L. Dykstra, "A method for finding projections onto the intersection of convex sets in Hilbert spaces," in *Advances in Order Restricted Statistical Inference* (Lecture Notes in Statistics), vol. 37, R. Dykstra, T. Robertson, and F. Wright, Eds. New York, NY, USA: Springer-Verlag, 1986, pp. 28–47.
[49] B. Khaleghi, S. Ahuja, and Q. M. J. Wu, "An improved real-time miniaturized embedded stereo vision system (MESVS-II)," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2008, pp. 1–8.
[50] Q. Yang, "A non-local cost aggregation method for stereo matching," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1402–1409.
[51] D. R. Karger, P. N. Klein, and R. E. Tarjan, "A randomized linear-time algorithm to find minimum spanning trees," *J. ACM*, vol. 42, no. 2, pp. 321–328, 1995.
[52] X. Mei, X. Sun, W. Dong, H. Wang, and X. Zhang, "Segment-tree based cost aggregation for stereo matching," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 313–320.
[53] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.
[54] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[55] K. Yamaguchi, D. McAllester, and R. Urtasun, "Efficient joint segmentation, occlusion labeling, stereo and flow estimation," in *Proc. 15th Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 756–771.
[56] C. Vogel, S. Roth, and K. Schindler, "View-consistent 3D scene flow estimation over multiple frames," in *Proc. 15th Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 263–278.
[57] V. Kolmogorov and C. Rother, "Minimizing nonsubmodular functions with graph cuts—A review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 7, pp. 1274–1279, Jul. 2007.



Shibiao Xu (M'15) received the B.S. degree in information engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2009, and the Ph.D. degree in computer science from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2014. He is currently a Post-Doctoral Researcher with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing. His current research interests include computer vision, and image-based 3D scene reconstruction and understanding.



Feihu Zhang (M'14) received the B.S. degree in computer science from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2014. He is currently an Intern with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His current research interests include computer vision and image processing.



Xiaofei He (M'05) received the B.S. degree in computer science from Zhejiang University, Hangzhou, China, in 2000, and the Ph.D. degree in computer science from the University of Chicago, Chicago, IL, USA, in 2005. He was a Research Scientist at Yahoo! Research Labs, Burbank, CA, USA. He is currently a Professor with the College of Computer Science, Zhejiang University. His research interests include machine learning, information retrieval, and computer vision.

Xukun Shen (M'10) received the B.S., M.S., and Ph.D. degrees in computer science from Beihang University, Beijing, China, in 1987, 1994, and 2007, respectively. From 1998 to 2003, he was an Associate Professor with the College of Computer Science, Beihang University. He is currently a Professor with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University. His main research interests are computer graphics and computer vision.

Xiaopeng Zhang (M'11) received the B.S. and M.S. degrees in mathematics from Northwest University, Xi'an, China, in 1984 and 1987, respectively, and the Ph.D. degree in computer science from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in 1999. He is currently a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His main research interests are computer graphics and computer vision.