MeshStereo: A Global Stereo Model with Mesh Alignment Regularization for View Interpolation

Chi Zhang[2,3*] Zhiwei Li[1] Yanhua Cheng[4] Rui Cai[1] Hongyang Chao[2,3] Yong Rui[1]

[1]Microsoft Research [2]Sun Yat-Sen University
[3]SYSU-CMU Shunde International Joint Research Institute, P.R. China
[4]Institute of Automation, Chinese Academy of Sciences

Abstract

We present a novel global stereo model designed for view interpolation. Unlike existing stereo models which only output a disparity map, our model is able to output a 3D triangular mesh, which can be directly used for view interpolation. To realize this feature, we partition the input stereo images into 2D triangles with shared vertices. Lifting the 2D triangulation to 3D naturally generates a corresponding mesh. A technical difficulty is to properly split vertices to multiple copies when they appear at depth discontinuous boundaries. To deal with this problem, we formulate our objective as a two-layer MRF, with the upper layer modeling the splitting properties of the vertices and the lower layer optimizing a region-based stereo matching. Experiments on the Middlebury and the Herodion datasets demonstrate that our model is able to synthesize visually coherent new view angles with high PSNR, as well as outputting high quality disparity maps which rank at the first place on the new challenging high resolution Middlebury 3.0 benchmark.

1. Introduction

Stereo model is a key component to generate 3D proxies (either point-clouds [18] or triangular meshes [15]) for view interpolation. Mesh-based approaches are becoming more and more popular [15][16][6][20] because they are faster in rendering speed, have more compact model size and produce better visual effects compared to the point-cloud-based approaches. A typical pipeline to obtain triangular meshes consists of two steps: 1) generate disparity/depth maps by a stereo model, and 2) extract meshes from the estimated disparity maps by heuristic methods, such as triangulating the polyline-simplified disparity regions [16].

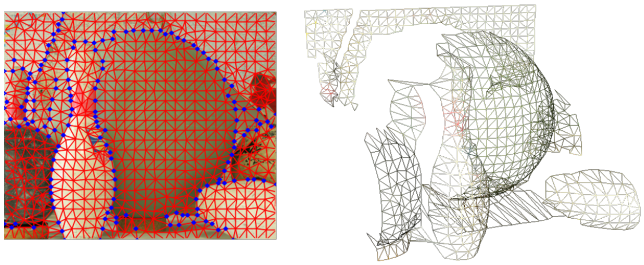


Figure 1. A 2D triangulation and a 3D mesh of the Bowling2 test case from Middlebury 2.0 [14] generated by our approach. Blue dots in the left images denotes 2D vertices with splitting probability greater than 0.5.

However, due to the separation of the two steps, the obtained meshes may not be optimal for the final goal of view interpolation. The stereo matching community mainly focuses on producing disparity maps with low pixel-wise error rates [5][9][25][23][17][12][24]. However, pixel-wise accurate disparity maps are not necessary nor sufficient to generate high quality meshes for the task of view interpolation. Moreover, the internal structures of a mesh, e.g., alignment of adjacent triangles, which would be helpful for rendering, are not exploited.

This observation motivates us to develop an integrated approach for view interpolation, which should have two desired properties: 1) the model can naturally produce a mesh structure, such that view interpolation reduces to a simple rendering problem; 2) the model can output a corresponding disparity map with high quality to make the rendering results look as natural as possible. In this paper, we propose a stereo model that works on a 2D triangulation of an input view with these two goals in mind.

In the proposed model, an input image is first partitioned into 2D triangular regions according to edge distribution and local visual consistency, as shown in Fig. 1, with common vertices shared by adjacent triangles. To generate the

*The first author was partially supported by the NSF of China under Grant 61173081, the Guangdong Natural Science Foundation, China, under Grant S2011020001215, and the Guangzhou Science and Technology Program, China, under Grant 201510010165.

corresponding 3D triangular mesh, a key step is to properly compute the disparities of the triangle vertices. However, due to the presence of discontinuities in depth, vertices on depth boundaries should be split to different triangle planes. To model such splitting properties, we assign a latent splitting variable to each vertex. Given the split or non-split properties of the vertices, a region-based stereo approach can be adopted to optimize a piecewise planar disparity map, guided by whether adjacent triangles should be aligned or split. Once the disparity map is computed, the split or non-split properties can be accordingly updated. By iteratively running the two steps, we obtain optimized disparities of the triangle vertices. The overall energy of the proposed model can be viewed as a two-layer Markov Random Field (MRF). The upper layer focuses on the goal of better view rendering results, and the lower layer focuses on better stereo quality.

1.1. Contribution

The contributions of this paper are two-fold

1. We proposed a stereo model for view interpolation whose output can be easily converted to a triangular mesh in 3D. The outputted mesh is able to synthesize novel views with both visual coherency and high PSNR values.

2. In terms of stereo quality, The proposed model ranks at the first place on the new challenging high-resolution Middlebury 3.0 benchmark.

2. Related Work

For a comprehensive survey on two-frame stereo matching, we refer readers to the survey by Scharstein and Szeliski [14]. And for an overview of point-based view interpolation approach, we refer readers to [19]. In this section, we mainly discuss works on stereo models for image-based rendering that make use of mesh structures.

To interpolate a new view, most of state-of-the-art view interpolation systems use multiple triangle meshes as rendering proxies [15]. A multi-pass rendering with proper blending parameters can synthesize new views in real-time [7]. Compared with the point-based rendering approaches [18], mesh-based approaches have lots of advantages, e.g., more compact model and without holes. Thus, mesh-based approaches have been a standard tool for general purpose view interpolation. Most of recent works focus on generating better meshes for existing views.

A typical pipeline to generate a triangular mesh consists of two consecutive steps: 1) run stereo matching to get a disparity map, and 2) generate a triangular mesh from the estimated map. The system presented by Zitnick et al. [26] is one of the seminal works. In the paper, the authors first generate a pixel-wise depth map using a local segment-based stereo method, then convert it to a mesh by a vertex shader program. In [16], Sinha et al. first extract a set of robust plane candidates based on the sparse point cloud and the 3D line segments of the scene. Then a piecewise planar depth map is recovered by solving a multi-labeling MRF where each label corresponds to a candidate plane. For rendering, the 3D meshes are obtained by triangulating the polygon-simplified version of each piecewise planar region. Separation of the depth map estimation and mesh generation process is a major problem of this pipeline. The obtained triangle meshes are not optimal for rendering.

For scenes with lots of planar structures, e.g., buildings, special rendering approaches have been developed. Plane fitting is a basic idea to generate rendering proxies. Xiao and Furukawa [20] proposed a reconstruction and visualization system for large indoor scenes such as museums. Provided with the laser-scanned 3D points, the authors recover a 3D Constructive Solid Geometry (CSG) model of the scene using cuboids primitives. For rendering, the union of the CSG models is converted to a mesh model using CGAL [1], and ground-level images are texture-mapped to the 3D model for navigation. Cabral et al. [6] proposed a piecewise model for floorplan reconstruction to visualize indoor scenes. The 3D mesh is obtained by triangulating the floorplan to generate a floor mesh, which is extruded to the ceiling to generate quad facades. Rendering is done by texture mapping those triangles and quads.

Despite of the pleasant visual effects, missing of necessary details is a common problem of these plane fitting-based systems. For example, we can easily see artifacts on regions of the furniture during the navigation in the video provided by [6]. Incorporating a region-based stereo matching when generating the rendering proxies, as proposed in this paper, is a solution for the problem. Our proposed model dose not have specific assumption on scene structures, can cope with non-planar shapes, therefore can serve as a powerful plugin for [16][6] to visualize finer structures.

To the best of our knowledge, the most related work of ours is Fickel et al. [8]. Starting from a 2D image triangulation, the method computes a fronto-parallel surface for each triangle, and then solve a linear system to adjust the disparity value on each private vertex. In mesh generation, a hand chosen threshold need to be set to classify each vertex as foreground or background. Due to the heavy fronto-parallel bias, the disparity maps's quality is low. In contrast, our model produces high quality disparity maps as well as good rendering results, and does not require a hand chosen threshold in mesh generation.

Another closely related work of us is Yamaguchi et al. [23][22]. In these papers, the authors optimize a slanted plane model over SLIC segments [2], and simultaneously

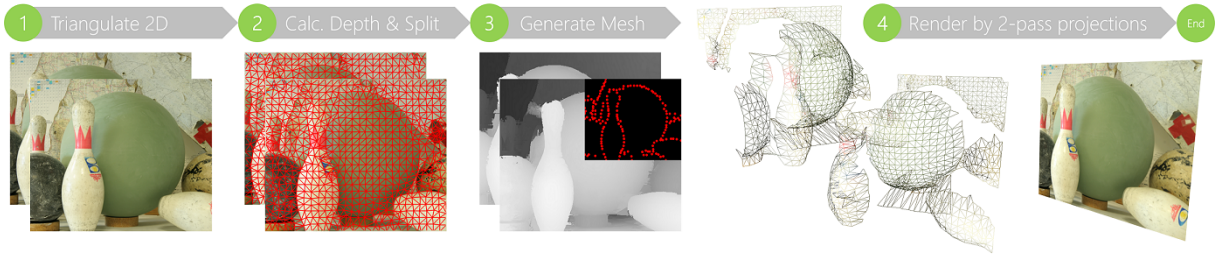


Figure 2. An overview of the proposed integrated approach. 1) The input stereo pair are triangulated in image domain. 2) Piecewise planar disparity maps and the corresponding splitting probabilities of the vertices are estimated by our model. 3) The 3D meshes are lifted from the 2D triangulations according to the disparity maps and splitting properties outputted by the previous step. 4) The generated meshes are texture mapped [7], an interpolated view is synthesized by blending the projections from the two meshes to the new view angle.

reason about the states of occlusion, hinge or coplanarity over adjacent segment pairs. Although both using a slanted plane model over super-pixels, there are noticeable differences between Yamaguchi et al. and ours. 1) To exploit the latent structures of a scene, instead of modeling the relationships over segment pairs, we model the splitting probabilities over shared triangle vertices, which is a natural choice for the task of view interpolation. 2) [23][22] performs a slanted plane smoothing on a pre-estimated disparity map, whose quality may limit the performance of the model. Our model avoids such limitation by computing matching cost based directly on photo consistency. 3) The above differences lead to completely different optimization approaches. We proposed an optimization scheme that makes use of PatchMatch [5] to achieve efficient inference in the slanted plane space.

3. Formulation

3.1. Overview

Fig. 2 illustrates the workflow of the proposed integrated approach. The basic idea is to first partition an input stereo pair to 2D triangles with shared vertices, then compute proper disparity/depth values for the vertices. With the known depth at each vertex, a 2D triangulation can be directly lifted to a 3D mesh, with shared vertices explicitly imposing the well-aligning properties on adjacent triangles. However, due to presence of depth discontinuities, vertices on depth boundaries should be split, that is, they should possess multiple depth values.

To tackle this difficulty, we model each triangle by a slanted plane, and assign a splitting probability to each vertex as a latent variable. Joint inference of the planes and the latent variables takes edge intensity, region connectivity and photo-consistency into consideration. The problem is formulated as a two layer continuous MRF with the following properties

1. Given the splitting variables, the lower layer is an MRF w.r.t. the set of plane equations, which is a region-based stereo model imposing photo consistency and normal smoothness.
2. The upper layer is an MRF w.r.t. the set of splitting probabilities, which provides a prior on the splitting structure of the triangulation given an image.
3. The two layers are glued together by an alignment energy which encourages alignment among neighboring triangles on continuous surface, but allows splitting at depth discontinuous boundaries.

3.2. The Lower Layer on Stereo

Notations. We parameterize the plane equation using the normal-depth representation: each triangle i is assigned a normal $\mathbf{n}_i = (n_{x,i}, n_{y,i}, 1)^\top$, and a disparity d_i of its barycenter $(\bar{x}_i, \bar{y}_i)^\top$. We use $\mathbf{N}, \mathbf{D}$ to denote the set of normals and disparities collected from all triangles. Note that we fix the n_z of a normal to 1, since this parameterization does not harm the plane's modeling capability, and enables a closed-form update in section 4.2.1. More details can be found in supplementary materials.

Matching Cost. The overall matching cost is defined as the sum of each individual slanted plane cost

$$E_{\text{MatchingCost}}(\mathbf{N}, \mathbf{D}) = \sum_i \rho_{\text{TRIANGLE}}(\mathbf{n}_i, d_i) \quad (1)$$
$$= \sum_i \sum_{p \in \text{Tri}_i} \rho\left(\begin{bmatrix} p_x \\ p_y \end{bmatrix}, \begin{bmatrix} p_x - \mathcal{D}_i(p) \\ p_y \end{bmatrix}\right)$$

where $\mathcal{D}_i(p) = a_i p_x + b_i p_y + c_i$ is the disparity value at pixel p induced by triangle i's plane equation. The plane coefficients a_i, b_i, c_i are converted from their normal-depth representation by $a = -\frac{n_x}{n_z}, b = -\frac{n_y}{n_z}, c = \frac{n_x \bar{x}_i + n_y \bar{y}_i + n_z d_i}{n_z}$ and $\rho(p, q)$ is the pixel-wise matching cost defined by

$$\rho(p,q) = \begin{array}{l} \text{hamdist}(\text{Census}_1(p) - \text{Census}_2(q)) \\ \quad + \omega \cdot \min(||\nabla I_1(p) - \nabla I_2(q)||, \tau_{grad}) \end{array} \quad (2)$$

where $\text{Census}_1(p)$ represents a 5×5 census feature around p on image I_1 and $\text{hamdist}(\cdot, \cdot)$ is the hamming distance. $\nabla \cdot$ denotes the gradient operator. The values of τ_{grad}, ω are determined by maximizing the survival rate in the left-right consistency check of the winner-takes-all results on the training images.

Normal Smoothness. We impose a smoothness constraint on the normal pairs of neighboring triangles with similar appearances

$$E_{\text{NormalSmooth}}(\mathbf{N}) = \sum_{i,j\in\mathcal{N}} w_{ij}(\mathbf{n}_i - \mathbf{n}_j)^\top(\mathbf{n}_i - \mathbf{n}_j) \quad (3)$$

where $w_{ij} = \exp(-||\mathbf{c}_i - \mathbf{c}_j||_1/\gamma_1)$ measures the similarity of the triangles' mean colors $\mathbf{c}_i$, $\mathbf{c}_j$, and γ_1 controls the influence of the color difference to the weight.

3.3. The Upper Layer on Splitting

Notations. Each shared vertex s is assigned a splitting probability α_s. Splitting probabilities of all 2D vertices are denoted by $\boldsymbol{\alpha}$.

Prior on the Splitting Probabilities. We model the prior by a data term and a smoothness term

$$E_{\text{SplitPenalty}}(\boldsymbol{\alpha}) = \sum_s \alpha_s \cdot \tau_s \quad (4)$$

$$E_{\text{SplitSmooth}}(\boldsymbol{\alpha}) = \sum_{s,t\in\mathcal{N}} w_{st}(\alpha_s - \alpha_t)^2 \quad (5)$$

where the penalty τ_s is set adaptively based on the evidence of whether s is located at a strong edge. $s, t \in \mathcal{N}$ means s and t are neighboring vertices, and w_{st} is a weight determined by the distance of the visual complexity levels of location s and t. The data term encourages splitting if s is on a strong edge. The smoothness term imposes smoothness if s and t are of similar local visual complexities. Specifically

$$\tau_s = \exp\left(-||\nabla I^3(x_s, y_s)||_1/\gamma_2\right) \quad (6)$$

$$w_{st} = \exp(-||k(x_s, y_s) - k(x_t, y_t)||/\gamma_3) \quad (7)$$

where $\{I^l, l = 1, 3, 5, 7, 9\}$ is a sequence of progressively gaussian-blurred images using kernel size $l \times l$. $k(x, y)$ is largest scale that pixel (x, y)'s color stay roughly constant

$$k(x, y) = \underset{j}{\text{argmax}}\{||I^l(x, y) - I(x, y)||_2 < 10,\ \forall l \le j\} \quad (8)$$

We discuss how γ_2, γ_3 are set in the experiment section.

3.4. Gluing the Two Layers

So far the two layers have defined two global models for the two sets of variables (i.e., the plane equations and the splitting probabilities), we further combine them together through an alignment term. The new term demands a strong alignment on shared vertices located at continuous surfaces, while allows a shared vertex to posses multiple depth values if its splitting probability is high

$$E_{\text{Alignment}}(\mathbf{N}, \mathbf{D}, \boldsymbol{\alpha}) = \sum_s (1 - \alpha_s) \cdot \sum_{i,j\in G_s} \frac{1}{2} w_{ij} \cdot \left(\mathcal{D}_i(x_s, y_s) - \mathcal{D}_j(x_s, y_s)\right)^2 \quad (9)$$

Algorithm 1 Optimize E_{ALL}

RandomInit($\mathbf{N}, \mathbf{D}$); $\boldsymbol{\alpha} = \mathbf{1}$;
repeat
  M-step: fix $\boldsymbol{\alpha}$, minimize E_{LOWER} with respect to $\mathbf{N}, \mathbf{D}$ by Algorithm 2.
  E-step: fix $\mathbf{N}, \mathbf{D}$, minimize E_{UPPER} with respect to $\boldsymbol{\alpha}$ by solving (13) using Cholesky decomposition.
until converged

where G_s denotes the set of triangles sharing the vertex s, and recall that $\mathcal{D}_i(x_s, y_s)$ defined in (1) denotes disparity of s evaluated by triangle i's plane equation. Therefore, at regions with continuous depth (α_s close to zero), the alignment term will encourage triangles to have their disparities agreed on s. At depth discontinuous boundaries (α_s close to 1), it allows the triangles in G_s to stick to their own disparities with little penalty.

Note that $E_{\text{NormalSmooth}}$ and $E_{\text{Alignment}}$ together impose a complete second-order smoothness on the disparity field. That is, the energy terms do not penalize slanted disparity planes.

3.5. Overall Energy

Combining the two layers and the alignment energy, we obtain the overall objective function

$$E_{\text{ALL}}(\mathbf{N}, \mathbf{D}, \boldsymbol{\alpha}) = E_{\text{MatchingCost}} + \lambda_{\text{NS}} \cdot E_{\text{NormalSmooth}} + \lambda_{\text{AL}} \cdot E_{\text{Alignment}} + \lambda_{\text{SP}} \cdot E_{\text{SplitPenalty}} + \lambda_{\text{SS}} \cdot E_{\text{SplitSmooth}} \quad (10)$$

where $\lambda_\times$ are scaling factors. For the ease of presentation in the next section, we introduce the following notations

$$E_{\text{UPPER}} = \lambda_{\text{SP}} E_{\text{SplitPenalty}} + \lambda_{\text{SS}} E_{\text{SplitSmooth}} + \lambda_{\text{AL}} E_{\text{Alignment}} \quad (11)$$

$$E_{\text{LOWER}} = E_{\text{MatchingCost}} + \lambda_{\text{NS}} E_{\text{NormalSmooth}} + \lambda_{\text{AL}} E_{\text{Alignment}} \quad (12)$$

4. Optimization

The overall energy E_{ALL} in (10) poses a challenging optimization problem since the variables are all continuous and tightly coupled. We propose an EM-like approach to optimize the energy, in which the splitting probabilities $\boldsymbol{\alpha}$ are treated as latent variables, and the plane equations $\mathbf{N}, \mathbf{D}$ are treated as model parameters. Optimization proceeds by iteratively updating $\boldsymbol{\alpha}$ (E-step), which can be solved in closed-form, and $\mathbf{N}, \mathbf{D}$ (M-step), which is optimized by a region-based stereo model through quadratic relaxation. The method is summarized in Algorithm 1.

4.1. E-Step: Optimize E_{UPPER}

By fixing $\mathbf{N}, \mathbf{D}$, Optimizing E_{ALL} is equivalent to optimize E_{UPPER}, which is listed as follow with some algebraic

rearrangement

$$\min_{\boldsymbol{\alpha}} \sum_s \big(\tau_s - c_{\text{AL}}(s)\big)\alpha_s + \sum_{s,t\in\mathcal{N}} w_{st}(\alpha_s-\alpha_t)^2 + \sum_s c_{\text{AL}}(s) \tag{13}$$

where $c_{\text{AL}}(s) = \sum_{i,j\in G_s} \frac{1}{2} w_{ij} \cdot \big(\mathcal{D}_i(x_s,y_s) - \mathcal{D}_j(x_s,y_s)\big)^2$ is the current alignment cost at vertex s. And $\sum_s c_{\text{AL}}(s)$ is now a constant independent of $\boldsymbol{\alpha}$. Since (13) is quadratic in $\boldsymbol{\alpha}$, we can update $\boldsymbol{\alpha}$ in closed-form by solving the positive semi-definite linear system obtained by setting the derivatives of (13) to zeros. We solve the linear system using Cholesky decomposition.

4.2. M-step: Optimize E_{LOWER}

By fixing $\boldsymbol{\alpha}$, Optimizing E_{ALL} is equivalent to optimize E_{LOWER}

$$\min_{\mathbf{N},\mathbf{D}} \Big\{ E_{\text{MatchingCost}}(\mathbf{N},\mathbf{D}) + \lambda_{\text{NS}} E_{\text{NormalSmooth}}(\mathbf{N},\mathbf{D}) + \lambda_{\text{AL}} E_{\text{Alignment}}(\mathbf{N},\mathbf{D}) \Big\} \tag{14}$$

However, this subproblem is not yet directly solvable due to the non-convexity of $\rho(\mathbf{n}_i, d_i)$. Inspired by [25][21], we adopt the quadratic splitting technique to relax the energy

$$\min_{\mathbf{N},\mathbf{D},\widetilde{\mathbf{N}},\widetilde{\mathbf{D}}} E_{\text{RELAXED}}(\mathbf{N},\mathbf{D},\widetilde{\mathbf{N}},\widetilde{\mathbf{D}}) \tag{15}$$

$$\equiv \min_{\mathbf{N},\mathbf{D},\widetilde{\mathbf{N}},\widetilde{\mathbf{D}}} \Big\{ E_{\text{MatchingCost}}(\widetilde{\mathbf{N}},\widetilde{\mathbf{D}}) + \frac{\theta}{2} E_{\text{Couple}}(\mathbf{N},\mathbf{D},\widetilde{\mathbf{N}},\widetilde{\mathbf{D}}) + \lambda_{\text{NS}} E_{\text{NormalSmooth}}(\mathbf{N},\mathbf{D}) + \lambda_{\text{AL}} E_{\text{Alignment}}(\mathbf{N},\mathbf{D}) \Big\}$$

in which

$$E_{\text{Couple}}(\mathbf{N},\mathbf{D},\widetilde{\mathbf{N}},\widetilde{\mathbf{D}}) = \sum_i (\boldsymbol{\Pi}_i - \widetilde{\boldsymbol{\Pi}}_i)^\top \Sigma (\boldsymbol{\Pi}_i - \widetilde{\boldsymbol{\Pi}}_i) \tag{16}$$

where $\boldsymbol{\Pi}_i = [\mathbf{n}_i^\top, d_i]^\top$, and $\Sigma = \text{diag}(\sigma_n, \sigma_n, \sigma_n, \sigma_d)$ controls the coupling strength between $\boldsymbol{\Pi}_i$ and $\widetilde{\boldsymbol{\Pi}}_i$. As the θ in (15) goes to infinity, minimizing E_{RELAXED} is equivalent to minimizing E_{LOWER}. We found that increasing θ from 0 to 100 in ten levels using the smooth step function offers good numerical stability. The values of σ_d, σ_n are set to the inverse variances of the disparities and normals respectively from the training images.

4.2.1 Optimize E_{RELAXED}

We propose an algorithm similar to that in [25] to optimize the relaxed energy. The algorithm operates by alternatively updating $\widetilde{\mathbf{N}}, \widetilde{\mathbf{D}}$ and $\mathbf{N}, \mathbf{D}$, which is summarized in Algorithm 2.

Update $\widetilde{\mathbf{N}}, \widetilde{\mathbf{D}}$. By fixing $\mathbf{N}, \mathbf{D}$, Problem (15) reduces to

$$\min_{\widetilde{\mathbf{N}},\widetilde{\mathbf{D}}} \Big\{ E_{\text{MatchingCost}}(\widetilde{\mathbf{N}},\widetilde{\mathbf{D}}) + \frac{\theta}{2} E_{\text{Couple}}(\mathbf{N},\mathbf{D},\widetilde{\mathbf{N}},\widetilde{\mathbf{D}}) \Big\} \tag{17}$$

Algorithm 2 Optimize E_{LOWER}

```
set θ to zero
repeat
  repeat
    Minimize (17) by PatchMatch [4][5].
    Minimize (18) by Cholesky decomposition.
  until converged
  Increase θ
until converged
```

This is a Nearest Neighbor Field (NNF) searching problem in disparity plane space, which can be efficiently optimized by PatchMatch [5][4].

Update $\mathbf{N}, \mathbf{D}$. By fixing $\widetilde{\mathbf{N}}, \widetilde{\mathbf{D}}$, problem (15) reduces to

$$\min_{\mathbf{N},\mathbf{D}} \Big\{ \frac{\theta}{2} E_{\text{Couple}}(\mathbf{N},\mathbf{D},\widetilde{\mathbf{N}},\widetilde{\mathbf{D}}) + \lambda_{\text{NS}} E_{\text{NormalSmooth}}(\mathbf{N},\mathbf{D}) + \lambda_{\text{AL}} E_{\text{Alignment}}(\mathbf{N},\mathbf{D}) \Big\} \tag{18}$$

It is easy to verify that all the three terms in (18) are quadratic in $\mathbf{N}, \mathbf{D}$ (details in supplementary materials). Therefore the update can be done in closed-form by Cholesky decomposition. Note that different from [25], we are working on triangles instead of image pixels. Therefore the update is much faster and scales to large size images.

5. Rendering

5.1. Triangulation and Mesh Generation

We propose a simple but effective method for image domain triangulation. First we partition the input image into SLIC [2] segments and perform a joint polyline simplification on each segment boundary. Second the output is obtained by delaunay triangulating each polygonized SLIC segment. We also try other triangulation approach, such as the one presented in Fickel et al. [8]. We refer interested readers to [8] for details.

Given the estimated piecewise planar disparity map and the splitting variables, a 3D mesh is lifted from the triangulation, where vertices with splitting probabilities > 0.5 are split to multiple instances, or merged to their median value otherwise. This results in a tightly aligned mesh and there is no hand-chosen thresholds involved in contrast to Fickel et al. [8]. Fig. 3 shows example 3D meshes generated by our model.

5.2. View Interpolation

Given a stereo pair I_L, I_R and their corresponding meshes, we now want to synthesize a new view at μ between the pair, where $0 \le \mu \le 1$ denotes the normalized distance from the virtual view I_M to I_L (with $\mu = 0$ the position of I_L, and $\mu = 1$ the position of I_R). First we

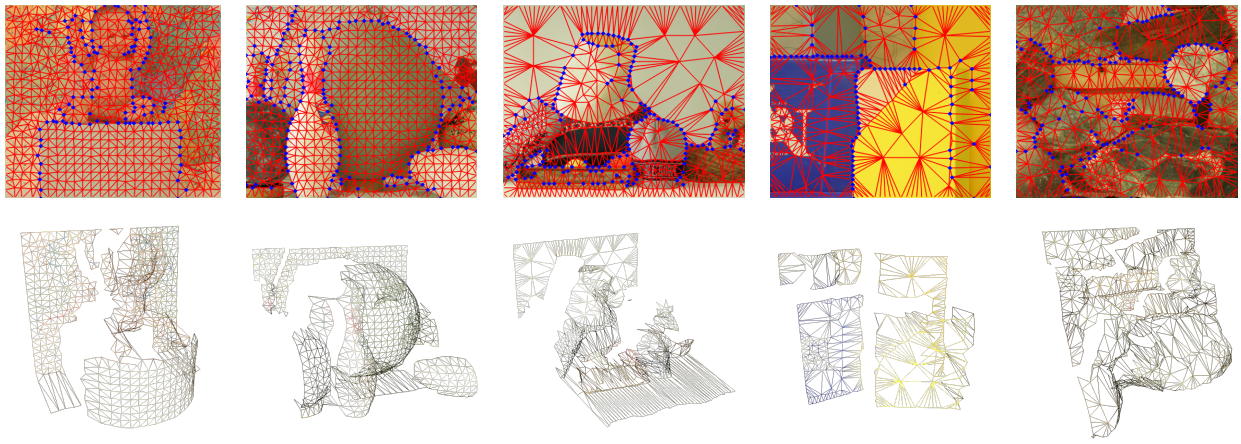
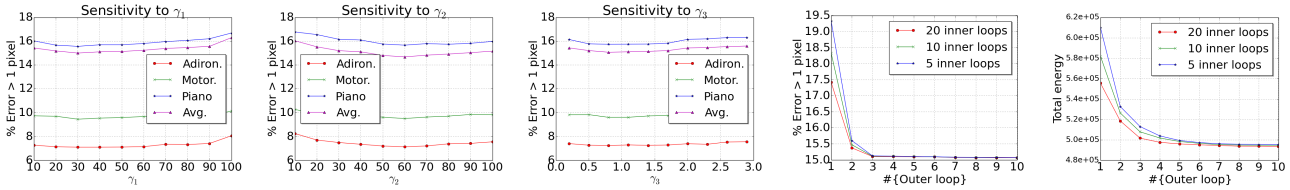


Figure 3. 2D image triangulations and 3D meshes from Middlebury 2.0 generated by our approach. Blue dots in the first row represent triangle vertices with splitting probabilities greater than 0.5. Test cases from left to right are (*Baby1, Bowling2, Midd1, Plastic, Rocks1*) respectively. Results on *Baby1,Bowling2* uses our proposed triangulation method, results on *Midd1, Plastic, Rocks1* use the one in [8].



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

Figure 4. Sensitivity and convergence analysis. (a)-(c) show that the results are not sensitive to $\gamma_1, \gamma_2, \gamma_3$. (d) and (e) depict the error rate and energy against the number of iterations, with different numbers of inner loops when solving E_{LOWER}.

perform projective texture mapping on the proxy meshes, then the interpolated view is rendered by blending the projections of two meshes on the new angle I_M

$$I_M(x,y) = (1-\mu)I_L^{\mu}(x,y)\delta_L(x,y) + \mu I_R^{\mu}(x,y)\delta_R(x,y) \tag{19}$$

where I_L^{μ} is the projection of the mesh of I_L at position μ, and $\delta_L(x,y) = 1$ if $I_L^{\mu}(x,y)$ is valid, and is 0 otherwise. I_L^{μ} and δ_R are defined analogly.

6. Experiments

To balance the subenergies, we employ the structured SVM [11] to the learn the weights $\lambda_{\text{AL}}, \lambda_{\text{DS}}, \lambda_{\text{NS}}, \lambda_{\text{SP}}$, λ_{SS}, where these parameters are optimized using the subgradient descent approach in [11] during training. We use five test cases *Art, Motorcycle, Pipes, Shelves, Teddy* from Middlebury 3.0 as training images. For the three parameters $\gamma_1, \gamma_2, \gamma_3$ which may need hand tuning, we found the results are not sensitive to their values. As shown in figure 4 (a)-(c), when $\gamma_1, \gamma_2, \gamma_3$ vary in the active range of their corresponding exponents, the average error rates vary in less than 1.0 percentage, and we set $\gamma_1 = 30, \gamma_2 = 60, \gamma_3 = 0.8$ according to the sensitivity figures. Figure 4 (d)-(e) provide a convergence analysis of the optimization framework, with different numbers of iterations when increasing θ in the inner loop of Alg. 2. The energy decreases steadily as iteration goes by. As expected, larger number of inner iterations shows faster convergence. The most dominant drops of both the error rate and the energy happen during the first three iterations. The error rate stays relatively stable after the first three iterations while the energy continues to decrease gently. This phenomenon is expected as at later stages the error rate is not able to reveal subtle changes less than 1.0 disparity.

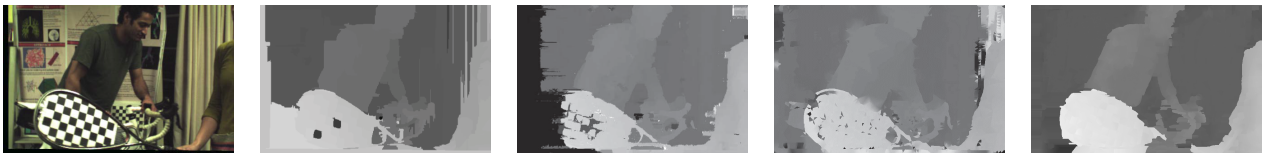
6.1. Rendering Results

Fig. 6 presents a comparisons of rendering results on the Middlebury 2.0 and the Herodion datasets[1] with ground truth intermediate views. Following Fickel et al. [8], we use VSRS3.5 [18] to generate interpolated view using disparity maps estimated by [24][12] as baselines, which are referred as VSRS-Nonlocal and VSRS-Volume. Compared to the point-based approaches (VSRS), our integrated approaches produces better visual effects, for example the surrounding of the reeindeer's neck; the text "Pattern Recognition" on the book, and the calibration pattern on the wall of the Herodion test case. Fig. 5 shows a comparisons of the stereo

[1]The Herodion array generates a video stream from 22 synchronized uncompressed images operating at 30 Hz on a single PC [3], courtesy from Fickel et al. [8].

	Aloe	Baby1	Books	Bowling2	Cloth3	Lamp.2	Laundry	Reindeer	Rocks1	Wood2
VSRS-Volume	26.13	27.45	24.20	26.02	25.75	28.88	21.67	25.03	28.46	29.76
VSRS-Nonlocal	25.86	28.39	25.83	27.47	25.97	30.99	24.03	26.03	29.63	31.47
Triangle-Nonlocal	27.01	31.61	28.35	30.14	31.39	31.50	27.04	27.34	34.35	31.50
Fickel et al.	27.69	32.15	28.41	30.10	31.31	31.76	27.82	27.70	34.45	33.28
Ours	**29.34**	**35.84**	**30.07**	**33.16**	**35.05**	**36.28**	**28.66**	**31.97**	**35.95**	**37.72**

Table 1. Comparison of PSNR values of the rendering results from sampled test cases on Middlebury 2.0.



(a) Left image (b) VSRS-Nonlocal (c) VSRS-Volume (d) Fickel et al. (e) Ours

Figure 5. Comparisons of estimated disparity maps on the Herodion datasets.

quality on the Herodion datasets. Compared to Fickel et al. [8], our approach generates cleaner disparity maps and sharper depth edges. Both [8] and ours are using the same rendering pipeline, however, our approach produces higher PSNR in all cases because of the better stereo quality.

Table 1 provides a comparisons on ten sampled test cases from Middlebury 2.0 in terms of PSNR. Our model outperforms other approaches for all test cases. Also notice in all the cases, the mesh-based approaches (including Fickel et al), produce higher PSNR than the point-based approaches, which demonstrates the advantages of using mesh structure. Moreover, with the reconstructed 3D meshes, rendering can be done much faster (0.1s) than VSRS (1s). Among the mesh-based approaches, we compare our results to *Triangle-Nonlocal*, which constructs the 3D meshes using the pre-estimated disparity maps from [24] by plane fitting. In all the cases, we obtain higher PSNR results, which demonstrates the advantages of the "integrated" approach compared to the "separated" one.

6.2. Stereo Results

We evaluated our model on the new Middlebury 3.0 [13] benchmark and currently ranks at the first place. Middlebury 3.0 is a new challenging benchmark suited with 30 high resolution test cases divided into training and test sets, including stereo pairs with different exposures, lighting conditions and slight rectification errors. Fig. 7 shows a comparison of the estimated disparity maps between SGM [10], Yamaguchi et al. [23] and our method, with corresponding error rates listed on Table 2. Our approach outperforms all the other approaches in all sampled test cases, including Yamaguchi et al. [23], which also uses a slanted plane model over super-pixels[2]. Our model also produces depth discontinuities that are more respectful to the ground truth scene structure, such as the mug handle in *Adirondack*, and the left boundary of the dustbin in *Recycle*. In the experiment we use 8000 triangles, and each test case takes about 60s on CPU. The source code is available for readers to reproduce our results.

[2]We use the publicly available code of [23] and set the preferred number of super-pixels to 8000 for fairness.

	Adiron.	Motor.	Piano	Playt.	Recyc.
Ours	**7.1**	**9.4**	**15.4**	**13.3**	**13.5**
Yamaguchi [23]	8.1	10.6	19.0	15.3	13.6
PatchMatch [5]	8.1	11.0	17.4	16.2	14.3
SGM [10]	15.3	10.9	16.4	15.8	14.6

Table 2. Bad pixel rates of sampled test cases evaluated at 1-pixel error thresholds on Middebury 3.0.

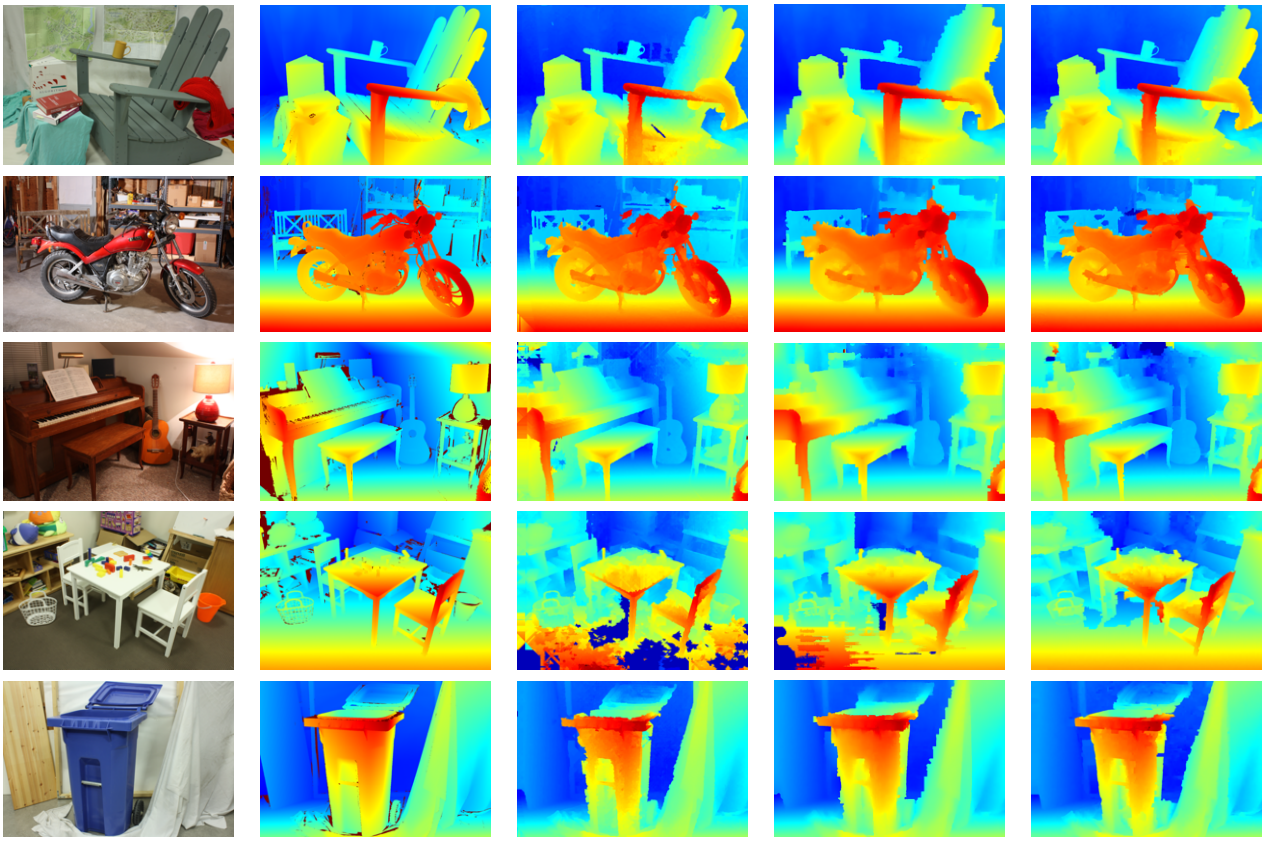
Contribution of the Latent Variables. To verify that the latent splitting variables do help improve the stereo quality, we compare our model with PatchMatch [5], which is essentially Eq. (1) in our formulation. To ensure fairness, both methods uses the same triangulation. As shown in Table 2, with the help of the splitting variables, our model improve the error rates of PatchMatch by 2.0 on average.

7. Conclusion

We have presented a global stereo model for view interpolation. Compared with existing stereo-based view interpolation approaches, where disparity map computation and mesh generation are separated into two steps, our approach integrates the two processes in a unified model. The proposed model is able to generate high quality disparity maps as well as synthesize new view angles with high visual coherency. Evaluations on the challenging Middlebury 3.0 benchmark demonstrated its excellent performance.

(a) GT (b) VSRS-Nonlocal (c) VSRS-Volume (d) Fickel et al. (e) Ours

Figure 6. Comparisons of view interpolation results on *Reindeer, Books, Herodion* datasets. Red digits at top right corners denote PSNR values. Blue boxes in the images denotes selected failed regions (best viewed zooming in).



(a) Left (b) GT (c) SGM (d) Yamaguchi et al. (e) Ours

Figure 7. Results on the high resolution Middlebury 3.0 [13] benchmark with ground truth (*Adirondack, Motorcycle, Piano, Playtable, Recycle*). Test cases are of averaged size 1400×1000.

References

[1] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2274–2282, 2012.

[3] H. H. Baker and D. Tanguay. A multi-imager camera for variable-definition video (xdtv). In *Multimedia Content Representation, Classification and Security*, pages 594–601. Springer, 2006.

[4] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics-TOG*, 28(3):24, 2009.

[5] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo-stereo matching with slanted support windows. In *BMVC*, volume 11, pages 1–11, 2011.

[6] R. Cabral and Y. Furukawa. Piecewise planar and compact floorplan reconstruction from images. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 628–635. IEEE, 2014.

[7] P. Debevec, Y. Yu, and G. Borshukov. *Efficient view-dependent image-based rendering with projective texture-mapping*. Springer, 1998.

[8] G. P. Fickel, C. R. Jung, T. Malzbender, R. Samadani, and B. Culbertson. Stereo matching and view interpolation based on image domain triangulation. *Image Processing, IEEE Transactions on*, 22(9):3353–3365, 2013.

[9] P. Heise, S. Klose, B. Jensen, and A. Knoll. Pm-huber: Patchmatch with huber regularization for stereo matching. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2360–2367. IEEE, 2013.

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

[11] S. Nowozin and C. H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends® in Computer Graphics and Vision*, 6(3–4):185–365, 2011.

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

[13] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *Pattern Recognition: 36th German Conference, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings*, pages 31–42. Springer, 2014.

[14] 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.

[15] S. N. Sinha, J. Kopf, M. Goesele, D. Scharstein, and R. Szeliski. Image-based rendering for scenes with reflections. *ACM Trans. Graph.*, 31(4):100, 2012.

[16] S. N. Sinha, D. Steedly, and R. Szeliski. Piecewise planar stereo for image-based rendering. In *ICCV*, pages 1881–1888. Citeseer, 2009.

[17] T. Taniai, Y. Matsushita, and T. Naemura. Graph cut based continuous stereo matching using locally shared labels.

[18] M. Tanimoto, T. Fujii, and K. Suzuki. View synthesis algorithm in view synthesis reference software 2.0 (vsrs2. 0). *ISO/IEC JTC1/SC29/WG11 M*, 16090, 2009.

[19] D. Tian, P.-L. Lai, P. Lopez, and C. Gomila. View synthesis techniques for 3d video. *Applications of Digital Image Processing XXXII, Proceedings of the SPIE*, 7443:74430T–74430T, 2009.

[20] J. Xiao and Y. Furukawa. Reconstructing the worlds museums. In *Computer Vision–ECCV 2012*, pages 668–681. Springer, 2012.

[21] L. Xu, C. Lu, Y. Xu, and J. Jia. Image smoothing via l0 gradient minimization. *ACM Transactions on Graphics (TOG)*, 30(6):174, 2011.

[22] K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun. Continuous markov random fields for robust stereo estimation. In *Computer Vision–ECCV 2012*, pages 45–58. Springer, 2012.

[23] K. Yamaguchi, D. McAllester, and R. Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *Computer Vision–ECCV 2014*, pages 756–771. Springer, 2014.

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

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

[26] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 600–608. ACM, 2004.