Journal of Electronic Imaging

JElectronicImaging.org

Local-linear-fitting-based matting for joint hole filling and depth upsampling of RGB-D images

Yanfu Zhang
Li Ding
Gaurav Sharma

Yanfu Zhang, Li Ding, Gaurav Sharma, “Local-linear-fitting-based matting for joint hole filling and depth upsampling of RGB-D images,” *J. Electron. Imaging* **28**(3), 033019 (2019), doi: 10.1117/1.JEI.28.3.033019.

Local-linear-fitting-based matting for joint hole filling and depth upsampling of RGB-D images

Yanfu Zhang,[a,†] **Li Ding,**[b,†] **and Gaurav Sharma**[b,c,*]
[a]University of Pittsburgh, Department of Electrical and Computer Engineering, Pittsburgh, Pennsylvania, United States
[b]University of Rochester, Department of Electrical and Computer Engineering, Rochester, New York, United States
[c]University of Rochester, Department of Computer Science, Rochester, New York, United States

Abstract. We propose an approach for jointly filling holes and upsampling depth information for RGB-D images captured with common acquisition systems, where RGB color information is available at all pixel locations whereas depth information is only available at lower resolution and entirely missing in small regions referred to as "holes." Depth information completion is formulated as a minimization of an objective function composed of two additive terms. The first data fidelity term penalizes disagreement with the observed low-resolution data. The second regularization term penalizes weighted depth deviations from a local linear model in spatial coordinates, where the weights are experimentally determined to ensure consistency between the RGB color image and the estimated depth image. Analogous to techniques used for optimization formulations of image matting, the completed depth image is then obtained by solving a large sparse linear system of equations. We also propose a memory-efficient implementation of the proposed method based on the conjugate gradient method. Visual evaluation of results obtained with the proposed algorithm demonstrates that the method provides high-resolution depth maps that are consistent with the color images. Furthermore, the memory-efficient implementation significantly reduces memory requirements, allowing for computation of the upsampled, hole-filled depth maps for typical RGB-D images on normal workstation hardware. Quantitative comparisons demonstrate that the method offers an improvement in accuracy over the current state-of-the-art techniques for depth information completion. Importantly, statistical analysis, which we present in this paper, also reveals that prior evaluations of depth upsampling accuracy are potentially biased because the evaluations inappropriately used preprocessed hole-filled data as "ground truth." An implementation of the proposed algorithm can be accessed and executed through Code Ocean: https://codeocean.com/capsule/5103691/tree/v1. *© 2019 SPIE and IS&T* [DOI: 10.1117/1.JEI.28.3.033019]

Keywords: depth map upsampling; hole filling; Laplacian matrix; RGB-D image.

Paper 190015 received Jan. 4, 2019; accepted for publication May 7, 2019; published online Jun. 4, 2019.

1 Introduction

RGB-D images are widely used for multiple purposes, for example, segmentation, tracking, image dehazing, and three-dimensional (3-D) scene reconstruction. A key challenge in using RGB-D images is that the depth data are often incomplete; compared to the RGB images, depth images are often in lower resolution and contain missing regions. Time of flight (ToF)- and structured light-based systems are the two prominent methods for capturing depth data. While ToF-based systems provide highly accurate depth information, they are relatively tedious to use and even after sophisticated alignment with images,[1] they usually offer a lower resolution than typical high-resolution color cameras. For structured light-based RGB-D images, a significant fraction of the pixels (up to 10%) is not assigned depth values due to the challenges of these systems. Thus, for both ToF- and structured light-based RGB-D image capture systems, joint depth upsampling and hole filling are required to generate complete RGB-D images.

The depth map upsampling problem has attracted considerable research. Traditionally this task is accomplished by bilinear or bicubic interpolation methods, which have difficulty in preserving the sharp edges in depth maps. "Manhattan world" assumption,[2] which states that images involving urban scenes are characterized by edge gradient statistics and are further verified on more general scenes,[3] has inspired several successful methods toward this problem. Several methods have been developed to overcome these problems, aiming at improving accuracy. One class of techniques relies on proposing *a prior* and optimizing an objective function that combines prior and data fidelity terms.[4–10] Diebel and Thrun[4] proposed a Markov random field (MRF)-based depth upsampling algorithm. This MRF framework has been further improved by other researchers.[11,12] Lo et al.[13] proposed a learning-based depth upsampling framework to handle the texture-copying artifacts, which are introduced by the inconsistency between the color edges and the depth discontinuities. Yang et al.[5] made use of a bilateral filter in an iterative refinement framework. The refinement is iteratively applied based on the current depth map and the RGB image. This algorithm can also work on two view depth map refinement with appropriate modification. In another work,[6] a guided filter was designed for edge-preserving filter, which can be viewed as an extension of the bilateral filter. Kopf et al.[7] proposed a joint bilateral filter, which is also similar in principle. Both filters can be used to upsample the depth map with a high-resolution RGB image. Park et al.[8] proposed an algorithm based on a nonlocal mean filter. The low-resolution depth map is preprocessed to detect outliers.

[*] Address all correspondence to Gaurav Sharma, E-mail: gaurav.sharma@rochester.edu

[†] Yanfu Zhang and Li Ding contributed equally in this paper.

1017-9909/2019/$25.00 © 2019 SPIE and IS&T

These points are removed, and to obtain the high-resolution depth map an objective function consisting of a smooth term, nonlocal structure term, and data term is optimized. This algorithm is also suitable for filling large holes in the depth data. Liu et al.[14] proposed a joint filtering algorithm for depth upsampling based on geodesic distance, which combines both color and spatial changes to recover sharp depth discontinuities. Ferstl et al.[9] gave an algorithm based on total generalization variance (TGV). A TGV regularization weighted according to intensity image texture is used in the objective function and the optimization is solved as a primal-dual problem. Yang et al.[10] built a color-guided adaptive regression model for depth map upsampling. Different edge-preserving terms, including nonlocal mean and bilateral filters are tested and an analysis is given on the parameter selection and the system stability. Another category of depth map upsampling utilizes segmentation techniques to extract depth information. Uruma et al.[15] started from an upsampled depth map using standard interpolation methods and refined the result by image segmentation techniques. The segmentation process serves a similar function in preserving edges as the aforementioned filters. Dong et al.[16] proposed a joint edge-guided convolutional neural network (CNN) to recover high-resolution depth map based on synthesized view quality. In addition, robust methods have also been developed for handling noisy depth information and color-depth inconsistency.[17–19]

Indeed, the Manhattan world assumption indicates that breaking apart hole filling and depth upsampling may suffer from biased local statistics compared to joint processing. However, performance evaluation for hole filling is challenging because ground truth data are rarely available without holes. As a result, the hole filling is seldom treated as an independent problem. Some methods[8,10] address hole filling at the same time as upsampling, whereas others[20] treat this as a separate problem. For instance, Feng et al.[21] first filled the depth holes caused by abnormal reflection using color information, and then filled the remaining holes according to the background; Wang et al.[22] preprocessed the depth maps using the deepest depth images, and then enhanced the results using geometry and color information. All these methods suffer from a lack of validation due to unavailability of ground truth data.

A common theme of prior algorithms, also adopted in our work, is to “fix” edges of the upsampled depth map for better consistency with the color image. Our work is inspired by Levin et al.’s optimization formulation of matting,[23] in which the alpha value for the matting mask is modeled as a linear combination of neighboring color values. Analogous to the matting problem, we formulate depth completion as an optimization problem. Specifically, the upsampled image is estimated by minimizing an objective function comprising two additive terms. The first term ensures that the estimated depth map is locally smooth, consistent with the color image, and the second term ensures consistency of the estimated upsampled data with the low-resolution observed data at the corresponding locations. Depth map completion is then achieved by solving a large sparse linear system following an approach similar to that adopted for the matting problem.[23] A key difference between the matting problem and our approach is that we model the depth as a linear function of the local spatial coordinates and not as a linear function of the image intensity values. The experiments also support our hypothesis that the performance for complicated scenes degrades with separate processing for hole filling and upsampling.

The main contributions of this paper are as follows:

- We propose an approach that jointly solves the closely related problems of depth map upsampling and hole filling for RGB-D images.
- We present a memory-efficient conjugate-gradient-based implementation for the proposed approach that significantly reduces the required memory by avoiding explicit storage of the image Laplacian matrix, which is extremely large for high-resolution images. This memory-efficient improvement allows the method to be used on typical resolution RGB-D images on normal workstation hardware.
- We demonstrate that prior evaluations of depth map upsampling are potentially biased because the evaluations inappropriately used preprocessed hole-filled data as “ground truth.” Specifically, statistical tests conducted with a number of alternative upsampling techniques demonstrate significant differences between error statistics for hole-filled and adjacent nonhole-filled regions, highlighting the fact that the use of such data as ground truth potentially biased evaluations of alternative techniques.

This paper is organized as follows: Sec. 2 describes the proposed algorithm and the memory-efficient improvement. Quantitative and the qualitative results are presented in Sec. 3. Section 4 concludes the paper.

2 Proposed Joint Hole Filling and Upsampling Algorithm

Our proposed method is motivated by the fact that regions of the image that correspond to a smooth 3-D surface can be locally approximated by a plane (e.g., via a Taylor series expansion). Thus, over small patches in the image, corresponding to regions with smooth surfaces, a local linear fit (in spatial coordinates) provides a good approximation to the depth. To account for edges, where the assumption breaks down, adaptive nonnegative weights are introduced for the linear fitting. The weighting seeks to concentrate the linear fit at each point on the neighboring pixel locations that are hypothesized, based on their color similarity to the pixel of interest, to be on the same side of the edge. The weights can be obtained from one of several edge-preserving techniques, for example, nonlocal means or bilateral filter. The completed depth map is obtained by minimizing an overall objective function that combines a term corresponding to the weighted deviation from the local linear fitting with a data fidelity term that penalizes deviations from observations at the locations where the low-resolution depth map is available. Figure 1 illustrates the intuition for the proposed scheme, which is described in detail next.

2.1 *Local-Linear-Fitting-Based Problem Formulation*

To formally describe our algorithm we use the simplified one-dimensional (1-D) representation in Fig. 2 that illustrates the contribution of one pixel to the objective function. The axis $\boldsymbol{g}$; (a two-dimensional vector for actual images)

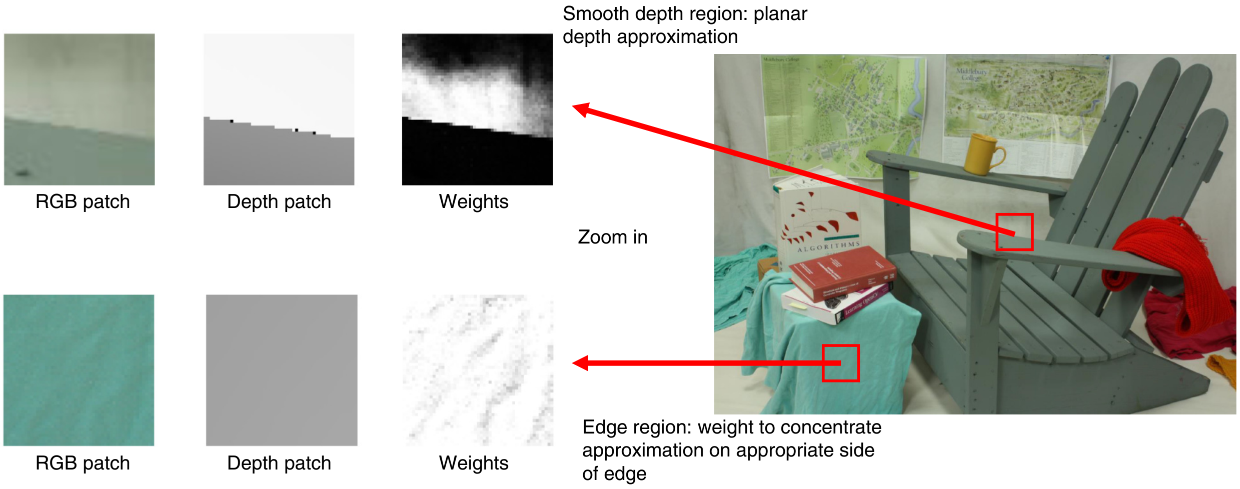
Smooth depth region: planar depth approximation
RGB patch
Depth patch
Weights
Zoom in
RGB patch
Depth patch
Weights
Edge region: weight to concentrate approximation on appropriate side of edge

Fig. 1 Motivation and intuitive explanation of the proposed depth completion algorithm. Two patches, respectively cropped from the handle area and the box area, display different depth characteristics. By using color-similarity-based adaptive weights for the locally linear fitting, the proposed method aims to construct a linear interpolation using only pixels within the same object, which are likely to be similar in both depth and color.

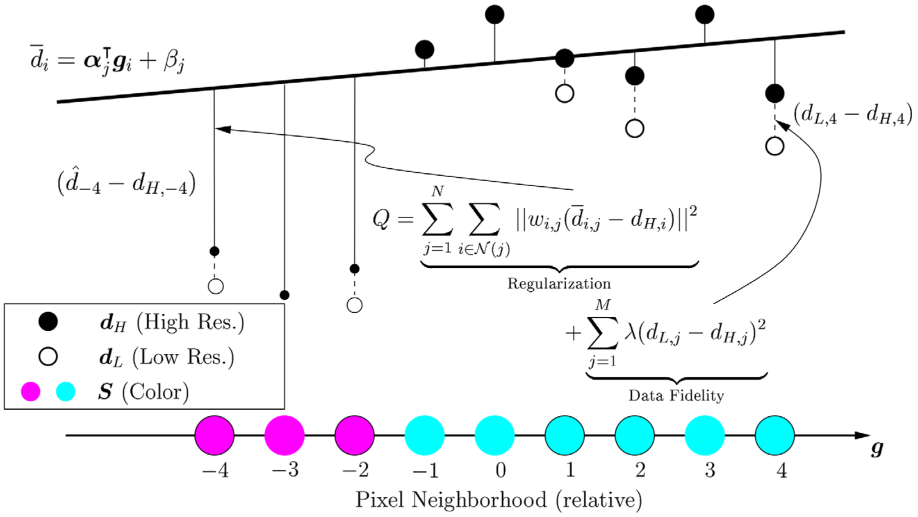
$\bar{d}_i = \boldsymbol{\alpha}_j^\intercal \boldsymbol{g}_i + \beta_j$
$(d_{L,4} - d_{H,4})$
$(\hat{d}_{-4} - d_{H,-4})$
$Q = \sum_{j=1}^{N} \sum_{i\in\mathcal{N}(j)} \|w_{i,j}(\bar{d}_{i,j} - d_{H,i})\|^2$
Regularization
$+\sum_{j=1}^{M} \lambda(d_{L,j} - d_{H,j})^2$
Data Fidelity
● $\boldsymbol{d}_H$ (High Res.)
○ $\boldsymbol{d}_L$ (Low Res.)
● ● $\boldsymbol{S}$ (Color)
$\boldsymbol{g}$
−4
−3
−2
−1
0
1
2
3
4
Pixel Neighborhood (relative)

Fig. 2 The problem formulation illustrated in 1-D. The magenta and cyan points show different color pixels in a color image patch, and the circles around the data points indicate the available low-resolution depth values. The unfilled and filled circles indicate, respectively, input and desired depth map values, and the black line shows the weighted linear fit over the example area. The sizes of filled circles represent the weights $w_{i,j}$.

represents the relative pixel positions of points in local pixel neighborhood of the target pixel, which is located at $\boldsymbol{g} = 0$. The low-resolution depth map, denoted by $\boldsymbol{d}_L$, is available at a subset of the pixel locations in the neighborhood as indicated in the figure by hollow circles. Color values, denoted by $\boldsymbol{S}$;, form the high-resolution RGB image. The goal is to estimate a high-resolution depth map $\boldsymbol{d}_H$, for which tentative values are shown by the solid circles. Our objective function is formulated as

$$Q = \sum_{j=1}^{N} \sum_{i\in\mathcal{N}(j)} \|w_{i,j}(\bar{d}_{i,j} - d_{H,i})\|^2 + \sum_{j=1}^{M} \lambda(d_{L,j} - d_{H,j})^2, \quad (1)$$

where j indices the pixel locations in the completed image, N is the number of pixels in the completed image, M is the number of pixels in the low-resolution depth map, $\bar{d}_{i,j}$ is the value of linear fitting of pixel i in the neighborhood $\mathcal{N}(j)$ of pixel j, $d_{H,i}$ is the estimated depth at pixel $i \in \mathcal{N}(j)$, $d_{L,j}$ is the depth value at the pixel j of the low-resolution depth map, $w_{i,j}$ is the similarity metric of pixel i and j, and λ is the free parameter to control the relation of fidelity and smoothness. The local linear fit is defined as

$$\bar{d}_{i,j} = \boldsymbol{\alpha}_j^\intercal \boldsymbol{g}_{i,j} + \beta_j, \quad (2)$$

where $\boldsymbol{\alpha}_j$ and β_j are the parameters for linear fitting at pixel $d_{H,j}$, α_j is a 2×1 vector and β_j is a scalar, and $\boldsymbol{g}_{i,j}$ is a 2×1 vector denoting the relative coordinates of pixel i in the neighborhood of the pixel j. Specifically, for a pixel i with coordinates (x_i, y_i) lying in the neighborhood $\mathcal{N}(j)$ of a target pixel j at with coordinates (x_j, y_j), we define $\boldsymbol{g}_{i,j} = (x_i - x_j, y_i - y_j)$. The first term in Eq. (1) is the regularization term and the second term is the data fidelity. The formulation is readily extended to include hole filling by adding a multiplicative factor in the data fidelity penalty term that corresponds to the indicator function of pixels that are not missing depth data, which will be discussed in Sec. 2.2.

Our problem formulation and the algorithmic approach we use for the solution (described in the next section) are inspired by Levin et al.'s formulation of matting as an optimization problem,[23] where the alpha channel is formulated as a weighted linear combination of neighboring color values. A key difference in our formulation is that our weighted local linear fit is formulated in terms of the local relative spatial position for the neighborhood, whereas in Levin et al.'s formulation,[23] the weighted linear fit is performed on the color values for the neighborhood pixels. Various alternative schemes can be used for determining the weights, which are considered subsequently, after we discuss the solution approach.

2.2 *Solution Approach*

A direct solution to the problem of Eq. (1) is challenging because we need to simultaneously determine both the upsampled depth map ($d_{H,i}$) and the fitting parameters (α_j and β_j), each of which depends on the other. To address this problem, we denote the vector of upsampled depth values by $\boldsymbol{d}_H \in \mathbb{R}^N$ and rewrite Eq. (1) in matrix form to obtain

$$Q = \sum_{j=1}^{N} \|\boldsymbol{W}_j[\boldsymbol{d}_{H,\mathcal{N}(j)} - \boldsymbol{G}_j\boldsymbol{p}_j]\|^2 + \lambda f_j, \tag{3}$$

where $\boldsymbol{G}_j = [\tilde{\boldsymbol{G}}_j, 1]$ and $\boldsymbol{p}_j = [\boldsymbol{\alpha}_j^{\intercal}, \beta_j]^{\intercal}$, with $\tilde{\boldsymbol{G}}_j$ as the $|\mathcal{N}(j)| \times 2$ matrix with $\boldsymbol{g}_{i,j}^{\intercal}$ as its i^{th} row, f_j is the value of the fidelity term from Eq. (1), $\boldsymbol{W}_j$ is a diagonal matrix with $w_{i,j}$ as the diagonal entries, and $\boldsymbol{d}_{H,\mathcal{N}(j)}$ is the vector formed by depth values selected from $\boldsymbol{d}_H$ over the neighborhood $\mathcal{N}(j)$. The vector $\boldsymbol{p}_j$ can be eliminated by replacing it in Eq. (3) by its optimal value

$$\begin{aligned}\boldsymbol{p}_j &= \arg\ \min_{**\boldsymbol{p}_j} \|\boldsymbol{W}_j(\boldsymbol{d}_{H,\mathcal{N}(j)} - \boldsymbol{G}_j\boldsymbol{p}_j^{\intercal})\|^2 \\ &= (\boldsymbol{G}_j^{\intercal}\boldsymbol{W}_{0,j}^{\intercal}\boldsymbol{G}_j)^{-1}\boldsymbol{G}_j^{\intercal}\boldsymbol{W}_{0,j}^{\intercal}\boldsymbol{d}_{H,\mathcal{N}(j)},\end{aligned} \tag{4}$$

where $\boldsymbol{W}_{0,j}$ is the diagonal matrix

$$\boldsymbol{W}_{0,j} = \boldsymbol{W}_j^{\intercal}\boldsymbol{W}_j. \tag{5}$$

Replacing $\boldsymbol{p}_j$ in Eq. (3) by Eq. (4), we obtain

$$Q = \sum_{j=1}^{N} \boldsymbol{d}_{H,\mathcal{N}(j)}^{\intercal}(\overline{\boldsymbol{G}}_j^{\intercal}\boldsymbol{W}_{0,j}\overline{\boldsymbol{G}}_j)\boldsymbol{d}_{H,\mathcal{N}(j)} + \lambda f_j, \tag{6}$$

$$\overline{\boldsymbol{G}}_j = \boldsymbol{E} - \boldsymbol{G}_j(\boldsymbol{G}_j^{\intercal}\boldsymbol{W}_{0,j}^{\intercal}\boldsymbol{G}_j)^{-1}\boldsymbol{G}_j^{\intercal}\boldsymbol{W}_{0,j}, \tag{7}$$

where $\boldsymbol{E}$; denotes the identity matrix. Details of the derivation are in Sec. 5.

The minimizer for the quadratic objective function Q is readily seen to be the solution to the linear equation

$$\boldsymbol{L}\boldsymbol{d}_H + \lambda\boldsymbol{A}_{\text{aff}}(\boldsymbol{d}_H - \boldsymbol{d}_L) = \boldsymbol{0}, \tag{8}$$

where L is the Laplacian matrix[24]

$$\boldsymbol{L} = \sum_{j=1}^{N} \overline{\boldsymbol{G}}_j^{\intercal}\boldsymbol{W}_{0,j}\overline{\boldsymbol{G}}_j, \tag{9}$$

where $\overline{\boldsymbol{G}}_j$ is padded with zero to full pixel size, and $\boldsymbol{A}_{\text{aff}}$ is the affinity matrix indicating the correspondence of pixels in low-resolution map to the desired high-resolution (and hole-filled) depth map. Specifically, $\boldsymbol{A}_{\text{aff}}$ is a diagonal matrix whose i'th diagonal entry is 1 if the pixel at location i in the high-resolution depth map is observed (in the low-resolution data with holes) and 0 otherwise.

2.3 *Memory-Efficient Implementation*

Equation (8) is a large sparse linear system of the form $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{A} = \boldsymbol{L} + \lambda\boldsymbol{A}_{\text{aff}}$, $\boldsymbol{x} = \boldsymbol{d}_H$, and $\boldsymbol{b} = \lambda\boldsymbol{A}_{\text{aff}}\boldsymbol{d}_L$, which can be efficiently solved with iterative methods such as a conjugate gradient solver. Algorithm 1 describes the conjugate gradient algorithm for solving sparse symmetric and positive-definite linear systems, which is suitable for the optimization in our problem where the Laplacian matrix automatically satisfies the required constraints. One bottleneck for the proposed algorithm is that, for high-resolution images, the Laplacian matrix is quite large, for example, a 10^6 (mega)-pixel image will require constructing a matrix with 10^{12} (tera) entries. Although the memory required for storing the matrix itself can be significantly reduced by using sparse matrix representations that exploit the structure of the Laplacian matrix, naive use of such representations does not directly reduce overall memory requirements when the matrices are used in subsequent computations. The problem can, however, be effectively addressed by a modification of the naive conjugate gradient method that calculates $\mathbf{A}\mathbf{q}_j$ in Algorithm 1 without explicitly constructing the matrix $\mathbf{A}$.[25,26]

Algorithm 1 Iterative for solving the sparse linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$; using the conjugate gradient algorithm.

```
Input: Initial guess $\mathbf{x}_0$, convergence threshold $\tau$, positive semidefinite
       matrix $\mathbf{A}$ and vector $\mathbf{b}$

Output: $\tilde{\boldsymbol{x}}$: estimate for $\boldsymbol{x}$, such that $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$;

Procedure initialize: $\tilde{\boldsymbol{x}} \leftarrow \boldsymbol{x}_0$, $\boldsymbol{r}_0 \leftarrow \boldsymbol{b} - \boldsymbol{A}\tilde{\boldsymbol{x}}$, $\boldsymbol{q}_0 \leftarrow \boldsymbol{r}_0$, $j \leftarrow 0$

while $\boldsymbol{r}_j^{\intercal}\boldsymbol{r}_j > \tau|\boldsymbol{x}|$ do
    $\alpha_j \leftarrow \frac{\boldsymbol{r}_j^{\intercal}\boldsymbol{r}_j}{\boldsymbol{q}_j^{\intercal}\boldsymbol{A}\boldsymbol{q}_j}$
    $\tilde{\boldsymbol{x}} \leftarrow \tilde{\boldsymbol{x}} + \alpha_j\boldsymbol{A}\boldsymbol{q}_j$
    $\boldsymbol{r}_{j+1} \leftarrow \boldsymbol{r}_j - \alpha_j\boldsymbol{A}\boldsymbol{q}_j$
    $\beta_j \leftarrow \frac{\boldsymbol{r}_{j+1}^{\intercal}\boldsymbol{r}_{j+1}}{\boldsymbol{r}_j^{\intercal}\boldsymbol{r}_j}$
    $\boldsymbol{q}_{j+1} \leftarrow \boldsymbol{r}_{j+1} + \beta_j\boldsymbol{q}_j$
end
```

From Eq. (9) we can obtain the entry $l_{i,j}$ at the i'th row and the j'th column of the Laplacian matrix $\boldsymbol{L}$ as

$$l_{i,j} = \sum_{k|(i,j)\in\mathcal{N}(k)} [\delta_{ij}w_{ki} - w_{ki}w_{kj}(\boldsymbol{g}_i - \boldsymbol{g}_k)^{\intercal}\boldsymbol{C}_k(\boldsymbol{g}_j - \boldsymbol{g}_k)], \tag{10}$$

where w_{ki} is the weight between pixel k and pixel i, δ_{ij} is the Kronecker delta, $\boldsymbol{C}_k$ is the inverse of $\overline{\boldsymbol{G}}_k^{\intercal}\boldsymbol{W}_{0,k}\overline{\boldsymbol{G}}_k$, and $\boldsymbol{g}_k$ is the global coordinate of the pixel k. Then, we break the summation stepwise, first computing

$$\boldsymbol{a}_k = \boldsymbol{C}_k\left(\sum_{j\in\mathcal{N}(k)} w_{kj}\boldsymbol{g}_j q_j - \boldsymbol{g}_k\overline{q}_k\right), \tag{11}$$

where q_j is the j'th entry of vector $\boldsymbol{q}$, and $\overline{q}_k$ is the average of q_j in $\mathcal{N}(k)$. Then,

$$b_k = \boldsymbol{g}_k^{\intercal}\boldsymbol{a}_k. \tag{12}$$

At the last step, we combine a_k and b_k to obtain $(\boldsymbol{Aq})_i$, the entry in the i'th column of the vector $\boldsymbol{Aq}$. For our problem setting, $(\boldsymbol{Aq})_i = (\boldsymbol{Lq})_i + \lambda(\boldsymbol{A}_{\text{aff}}\boldsymbol{q})_i$, where from the preceding discussion, we can obtain the Laplacian term as

$$(\boldsymbol{Lq})_i = \sum \delta_{ij}w_{ki}q_i - \boldsymbol{g}_i^{\intercal}\sum_{k\in\mathcal{N}(i)} \boldsymbol{a}_k w_{ki} + \sum_{k\in\mathcal{N}(i)} b_k w_{ki}. \tag{13}$$

Computation of the term $(\boldsymbol{A}_{\text{aff}}\boldsymbol{p})_i$ is straightforward. Using Eq. (13) in the update step for $\boldsymbol{r}_{j+1}$ in Algorithm 1, we can get the desired memory-efficient realization for the proposed algorithm. Details of the derivation of Eq. (13) are included in Sec. 6.

We note that for the colorization problem,[26] in additional to memory efficiency, computational efficiency can also be obtained. For the colorization problem, the summations in Eqs. (11)–(13) can be efficiently calculated by integral image techniques and dynamic programming. This step is possible because the formulation in colorization utilizes a fixed summation table. However, in our case, the $\boldsymbol{g}_j w_{kj}$ is a summed table of localized filters w_k, which means that the values reused in summed table are no longer applicable here. This key difference prevents us from obtaining a computational acceleration for our depth completion problem using the same approach but still allows for the memory-efficient implementation. The space complexity of the improved implementation is $O(N)$, which is a very significant improvement over the $O(N^2)$ space complexity of the naive conjugate gradient approach (where N is the number of pixels of an image).

2.4 *Weighting Functions*

We explored different choices for the weighting functions, in the proposed algorithm:

- Gaussian profile weights, which are defined as

$$w_{i,j} = \exp\left(-\frac{\|\boldsymbol{s}_i - \boldsymbol{s}_j\|^2}{2\sigma^2}\right), \tag{14}$$

where $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ are the RGB pixel values in the image $\boldsymbol{S}$ at corresponding positions, and σ controls the relative emphasis of pixel similarity in the allocation of weights. The weights are analogous to the commonly used bilateral filter;[14] however, unlike the typical bilateral filter, we do not use a distance decay term in Eq. (14) because the window we use is quite small in relation to size of the high-resolution images.

- Laplacian profile weights, defined as

$$w_{i,j} = \exp\left(-\frac{|\boldsymbol{s}_i - \boldsymbol{s}_j|}{\sigma}\right), \tag{15}$$

where the symbols are as defined previously. Compared to the Gaussian weights, the Laplacian weights are more localized on neighborhood pixel pairs with smaller differences in colors.

- Max channel weights, defined as

$$w_{i,j} = \exp\left(-\frac{\max_{C\in\{R,G,B\}}|s_{C,i} - s_{C,j}|}{\sigma}\right), \tag{16}$$

where $s_{C,i}$ and $s_{C,j}$ are the pixel values of the RGB image $\boldsymbol{S}$ at corresponding position in channel C. These weights are more sensitive to color difference than the Gaussian/Laplacian weights.

- Gaussian weights combined with depth information, defined as

$$w_{i,j} = \exp\left(-\left[\frac{\|\boldsymbol{s}_i - \boldsymbol{s}_j\|^2}{2\sigma_1^2} + \frac{\|\hat{d}_i - \hat{d}_j\|^2}{2\sigma_2^2}\right]\right), \tag{17}$$

where $\hat{d}_i$ and $\hat{d}_j$ are the depth information estimated by a median filter based on the low-resolution depth maps. We use this combination of color similarity and depth similarity to avoid inappropriate weighting in situations where there is a complex color pattern with a planar depth spatial variation.

3 Experimental Results

We test our algorithm on the Middlebury (stereo) dataset,[27–30] which provides high-resolution RGB images of multiple views and corresponding disparity maps, which are used as the ground truth in our experiment. A 7 × 7 square patch is used as the neighborhood $\mathcal{N}(j)$ [the neighborhood size $|\mathcal{N}(j)| = 49$] and the parameter $\lambda = 10^5$. The RGB-D images are zero-padded for consistent use of Eq. (3), and the padded area is cropped out in the final results. The parameter σ^2 in Eq. (14) for computation of the weights $w_{i,j}$ is set to one-third of the local variance in each window. In each patch, the weight of the center pixel is set to 10^{-5}. For the naive, nonmemory-efficient implementation, we use the built-in MATLAB™ conjugate gradient solver (*cgs*) for solving Eq. (8) (a tolerance of 10^{-10} and the maximum number of iteration was set to 10^4).

3.1 *Qualitative and Quantitative Results*

The proposed algorithm is suitable for both hole filling and depth map upsampling, as indicated earlier. In this section, we first visually examine the performance for filling holes in the depth map, as shown in Fig. 3. From the images in the last row, we see that the holes, which correspond to

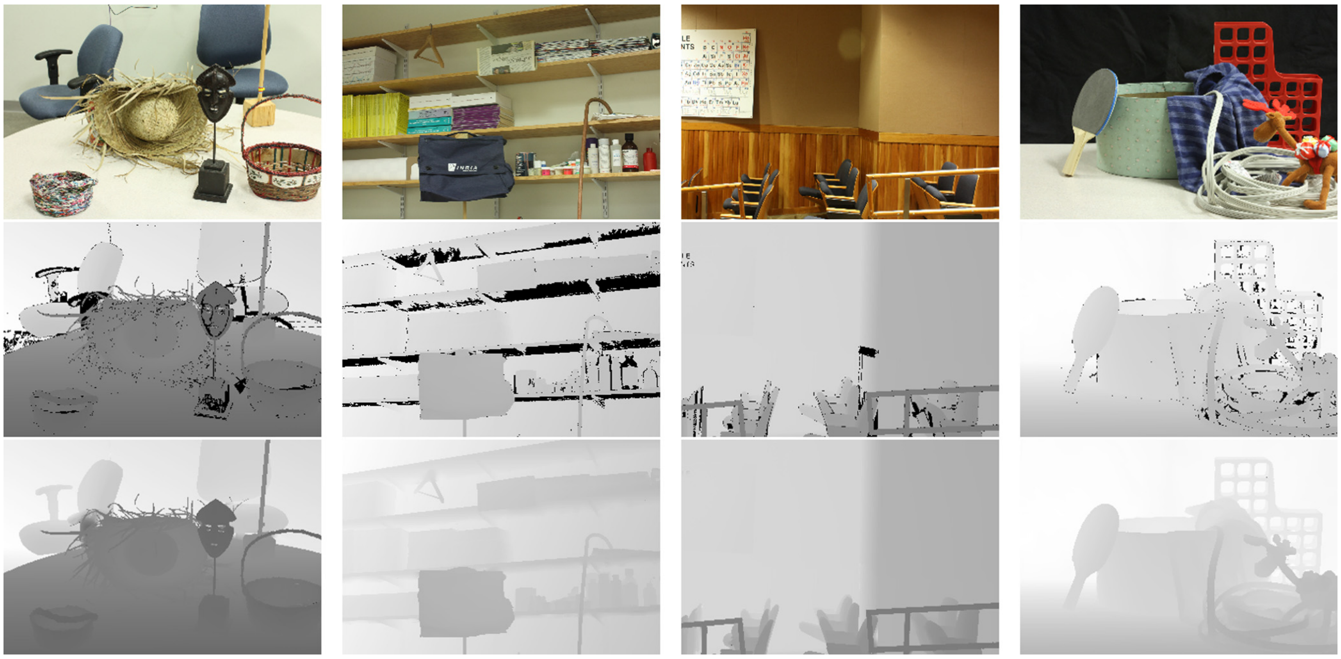

Fig. 3 Visual results demonstrating the hole-filling ability of the proposed algorithm on a subset of images from the Middlebury stereo dataset 2014.[30] The first row shows the input high-resolution color images and the second row shows the corresponding depth maps. The results are shown in the last row. The illustration in this figure has used input images with approximately 60 to 80 k pixels.

the occluded area in the disparity map, are filled well. Unlike traditional interpolation methods, our algorithm is able to fix the holes in the depth images, so as to keep the consistency of depth map edges with those in the RGB images and avoid smoothing in such areas. For example, see the third row of Fig. 3. The computed depth map values for the missing points in the original depth map along the wall are consistent on the two sides of the edge and not blurred across the edge.

Quantitative results for depth upsampling are obtained on the Middlebury stereodataset 2005,[27] which has been used in prior evaluations. We compare the proposed method against the following prior methods: IBL,[5] TGV,[9] AD,[31] DGDE,[20] and RCG.[17] Unlike prior evaluations reported in the literature, we use the original dataset without hole filling as the ground truth, because we hypothesize that prior evaluations of depth upsampling are biased by the inappropriate use of preprocessed hole-filled data as "ground truth." We justify this hypothesis in Sec. 3.3.

To quantitatively evaluate the performance on original Middlebury dataset, we first downsample the input depth map to obtain the low-resolution version, and then run different algorithms on these images to obtain the upsampled versions. We use mean absolute error (MAE) as the metric to evaluate the performance of different algorithms. Table 1 summarizes the results (Code for IBL implementation is provided by Chunhua Shen[33]). Figure 4 shows the corresponding visual results for 4× upsampling. The proposed method outperforms the state-of-the-art methods in most cases. Compared to the other Laplacian-based RCG method,[17] the proposed method significantly improves the performance. In addition, the proposed method with the memory-efficient implementation is capable of processing typical high-resolution images, unlike the RCG method,[17] for which the memory requirements are inordinately large. The results in Table 1 used the Gaussian weights combined with depth information defined in Eq. (17). We also compared the performance of the proposed approach with alternative weights described in Sec. 2.4. Results for these comparisons are presented in Sec. 7.

The results for the DGDE[20] method reported here have been computed directly using the corresponding upsampling model for each scale. We have also tested the procedure of first hole filling then upsampling, which resulted in worse results. Compared to DGDE[20] in which iterative enhancements on the guided weight function for dependency modeling are used, the proposed method has the advantage of a clearer intuition and simplicity in usage. The proposed method does not require training of a dedicated model for each setting with different upsampling scale and image resolutions. Moreover, it is capable of solving the hole filling and upsampling simultaneously, which is of particular importance because both problems occur for typical depth image-capturing methods. Again, we can observe that the proposed method clearly outperforms DGDE.[20]

The memory-efficient implementation and naive implementation give very similar results, when using the same threshold for conjugate gradient solver. Figure 5 shows an example comparison of the results for the two implementations. By using the memory-efficient implementation, the required memory for a two mega-pixel image is reduced from 60 to 2 GB memory (for computations performed in standard double-precision float-point format).

3.2 *Color and Depth Inconsistency Handling*

Regions with discrepancies between color and depth edges commonly pose a challenge for depth upsampling and hole-filling algorithms. Often the reliance on color information to upsample and fill holes in such regions results in spurious variations in depth that mirror the color texture variations.

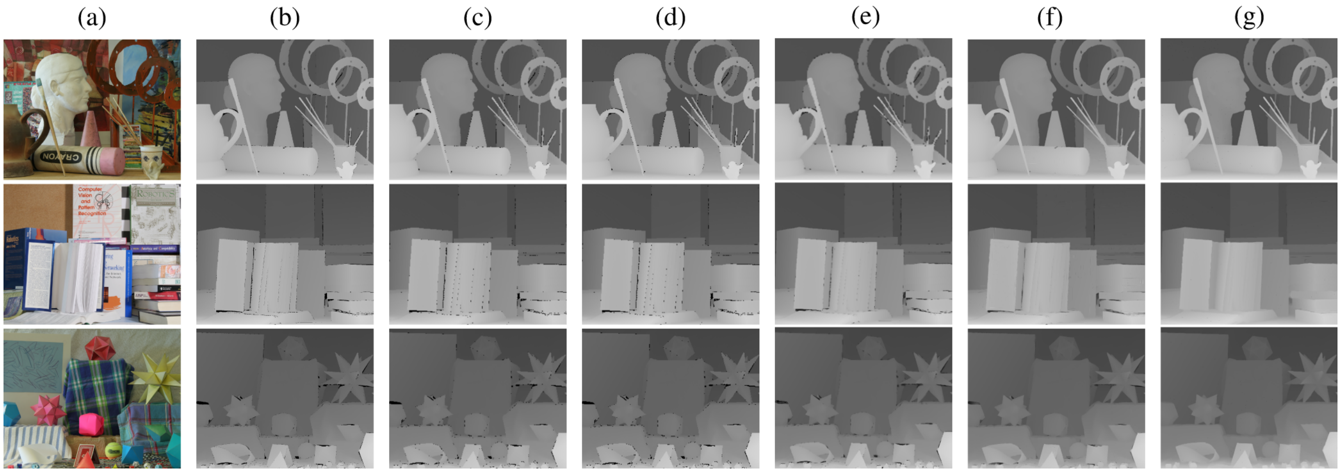

Fig. 4 Visual comparison of results obtained for 4× upsampling with different algorithms for images from the Middlebury dataset.[27] Rows from top to bottom correspond to the images Art, Books, and Moebius. (a) RGB color image, (b) GT high resolution depth ground-truth, unsampled and hole-filled versions obtained with (c) bilinear interpolation, (d) bicubic interpolation, (e) IBL,[10] (f) TGV,[9] and (g) proposed methods.

Table 1 Quantitative comparison of the performance of the different algorithms. MAE depth-disparity values are reported for the images in the Middlebury dataset[27] for four different upsampling ratios, as listed in the second row. The best result for each case is highlighted in bold font.

	Images											
	Art				Books				Moebius			
	Sample rate											
Methods	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×
Bicubic	0.8965	1.4298	2.4363	4.3456	0.7911	1.0842	1.7031	2.5419	0.6855	1.0287	1.5821	2.5527
Bilinear	0.7642	1.2300	2.1495	3.9500	0.6620	0.8993	1.4183	2.1174	0.5685	0.8578	1.3347	2.1942
IBL[5]	0.5016	0.8934	1.7028	4.2324	0.2790	0.7361	1.4056	2.4561	0.3987	0.7071	1.1289	2.5885
TGV[9]	0.6457	0.8926	3.2633	7.6490	0.5980	0.7507	2.3091	6.3240	0.4722	0.5627	2.0375	6.6210
AD[31]	0.3571	0.8334	1.7943	3.8267	**0.1316**	**0.3006**	0.5387	1.0706	**0.1497**	0.3461	0.7164	1.5300
DGDE[20]	0.5076	0.8867	1.5465	2.7042	0.6365	1.0936	1.4957	1.9623	0.5435	0.8804	1.2256	1.6751
RCG[17]	1.0324	1.6941	2.1856	4.1398	1.1922	1.8922	2.3520	3.2686	1.0169	1.6053	2.0135	2.9229
Color[32]	0.4423	0.8765	1.7616	3.6033	0.1986	0.3594	0.6655	1.1888	0.1864	0.3426	0.6478	1.2393
Color + depth	**0.2744**	**0.6612**	**1.3049**	**2.6243**	0.1671	0.3067	**0.5207**	**0.9030**	0.1714	**0.3396**	**0.5328**	**1.0317**

We illustrate the performance of the proposed algorithm in such regions by selecting two suitable regions from the results presented in Sec. 3.1 and present zoomed-in views for these regions for 4× and 8× upsampling, respectively, in Figs. 6 and 7. By comparing the ground truth to the results obtained with the proposed method, we can see that the method works fairly well and does not introduce spurious depth variations correlated with the color texture, whereas spurious texture is seen for TGV with 8× upsampling.

3.3 *Evaluation Datasets: Ground Truth Considerations*

Prior evaluations of upsampling have used, as "ground truth," a hole-filled version of the original Middlebury dataset[27] obtained using the method of Park et al.[8] In this section, we highlight the fact that such assessment is potentially biased by the fact that the hole-filled data does not indeed represent ground truth. The assessments are therefore potentially biased: instead of assessing the accuracy of upsampling, the prior evaluations are instead (partly) assessing conformance of the upsampling with the hole-filling technique used for generating the "ground truth." To test our hypothesis, we formulate and conduct a statistical test shown in Fig. 8. We run the algorithms on downsampled versions of the "ground truth" hole-filled depth map and compute the error between the upsampled depth map and the "ground truth." We then compare the error statistics of pixels that correspond to holes in the original Middlebury dataset to

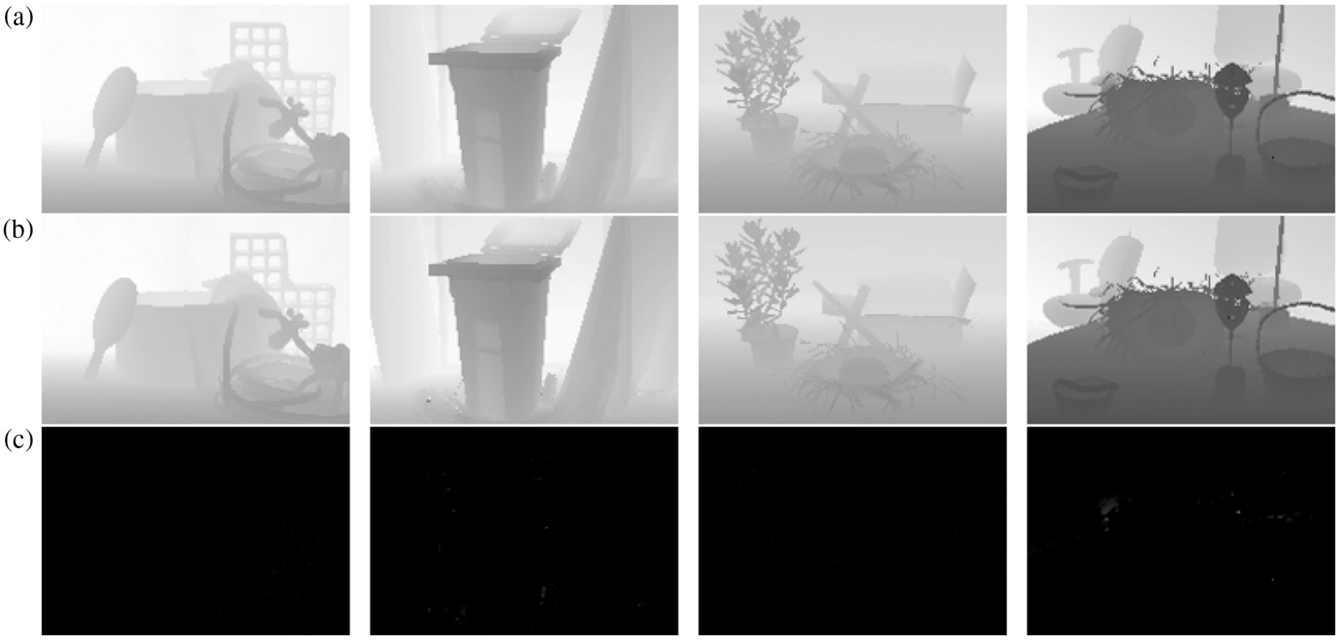

Fig. 5 Visual comparison of results obtained with the proposed approach using either the naive conjugate gradient or the memory-efficient solver for a subset of images from the Middlebury dataset.[27] The three rows of images from top to bottom represent (a) the depth maps obtained by using the MATLAB™ inbuilt cgs, (b) depth maps obtained with the proposed memory-efficient implementation, and (c) the corresponding difference depth maps.

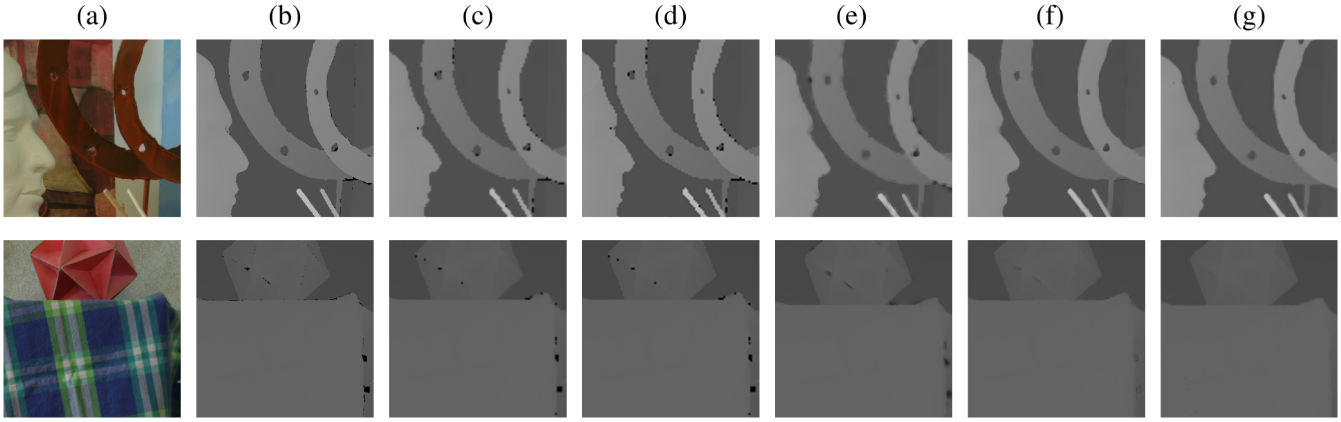

Fig. 6 Enlarged views of square regions selected from Fig. 4 that visual illustrate the performance of the proposed method and alternatives for 4× depth upsampling in regions with inconsistency between color and depth information. (a) RGB color image, (b) GT high resolution depth ground-truth, unsampled and hole-filled versions obtained with (c) bilinear interpolation, (d) bicubic interpolation, (e) IBL,[10] (f) TGV,[9] and (g) proposed methods.

nonhole-filled pixels adjacent to the holes. Specifically, we run Welch's t-test[34] on two data samples X and Y where the null hypothesis is that the error in two samples comes from the distributions with equal means but unequal variances. The p-values, which indicate the probability of accepting the null hypothesis, are listed in Table 2. The results show that, for most algorithms, statistically, there exists a performance discrepancy between the regions of holes and the adjacent nonhole-filled pixels. This indicates that the assessed accuracy for these algorithms does not necessarily characterize their ability to fill holes and is likely, instead, assessing conformance with the original hole-filling algorithm used for generating the "ground truth." For completeness, we also provide, as Table 5 in Sec. 8, the potentially biased version of the Table 4 obtained by treating the preprocessed hole-filled data[8] as ground truth.

3.4 *Discussion*

Table 1 illustrates that the depth map obtained with the proposed algorithm is accurate and achieves the state-of-the-art results on the common benchmarking dataset, providing a better performance compared with other algorithms. The proposed algorithm, however, still suffers from some limitations. First, there are a few outlier points where the method yields a large error. Second, the edges are not sharply defined, especially under high upsampling rate (this limitation is also typical for other depth completion algorithms).

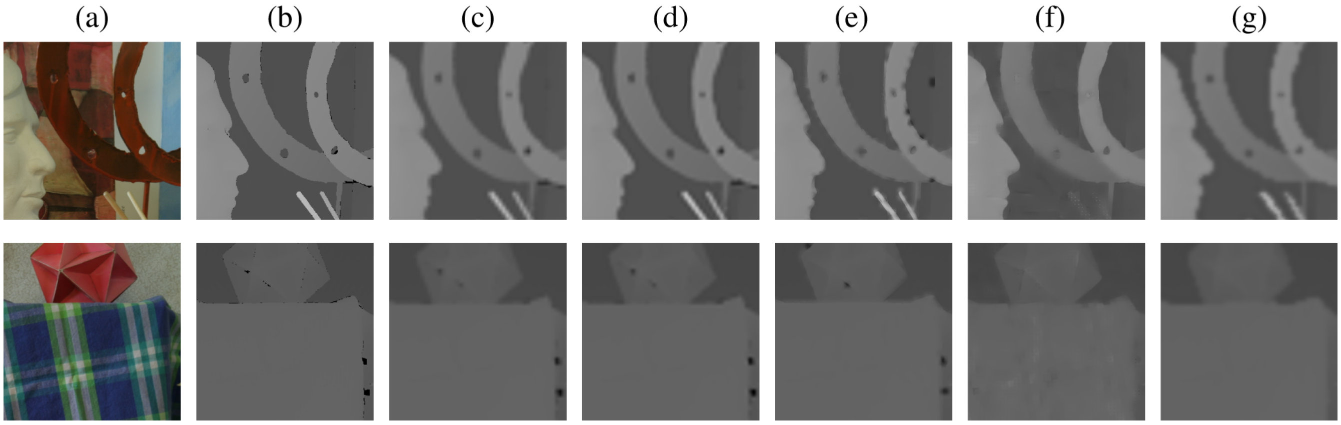

Fig. 7 Enlarged views of square regions selected from Fig 4 that visual illustrate the performance of the proposed method and alternatives for 8× depth upsampling in regions with inconsistency between color and depth information. (a) RGB color image, (b) GT high resolution depth ground-truth, unsampled and hole-filled versions obtained with (c) bilinear interpolation, (d) bicubic interpolation, (e) IBL,[10] (f) TGV,[9] and (g) proposed methods.

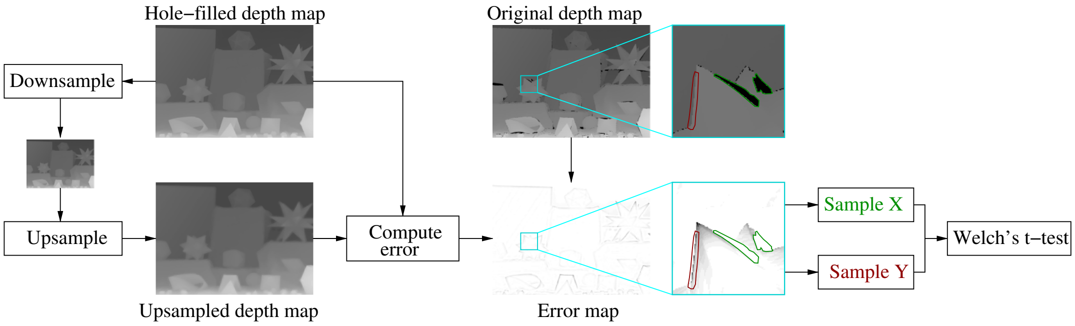

Fig. 8 Statistical test for bias in assessments of upsampling algorithms caused by computation of accuracy using preprocessed data (hole-filled Middlebury dataset[8]) instead of actual ground-truth data. Using Welch's t-test, statistics of computed errors (MAE) are compared between two samples: sample X corresponding to hole-filled regions and sample Y corresponding to nonhole-filled regions adjacent to the holes. The regions corresponding to the two samples are highlighted in the region indicated by the cyan rectangle.

The computational requirements are an additional challenge: to process a 1088×1296 pixel image, our algorithm takes about 40 min. While the time requirement is comparable for several other completion algorithms, a speed-up is desirable for many applications. In future work, parallel techniques may be promising for accelerating the computations; hierarchical upsampling also has the potential to offer speed-up, although our preliminary experiments show that directly adapting the proposed algorithm to a hierarchical structure yields worse results. An additional limitation of the proposed algorithm is that it is not directly designed to handle the high levels of noise in the input low-resolution

Table 2 The p-values for a statistical test of consistency of error statistics over hole-filled and nonhole-filled regions for different upsampling methods (see Fig. 8 and text in Sec. 3.3, for details).

	Images											
	Art				Books				Moebius			
	Sample rate											
Methods	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×
IBL[5]	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$
TGV[9]	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$
AD[31]	0.1429	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$
DGDE[20]	0.6959	0.9951	0.8028	0.9369	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$
Color + depth	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$	$\leq 10^{-4}$

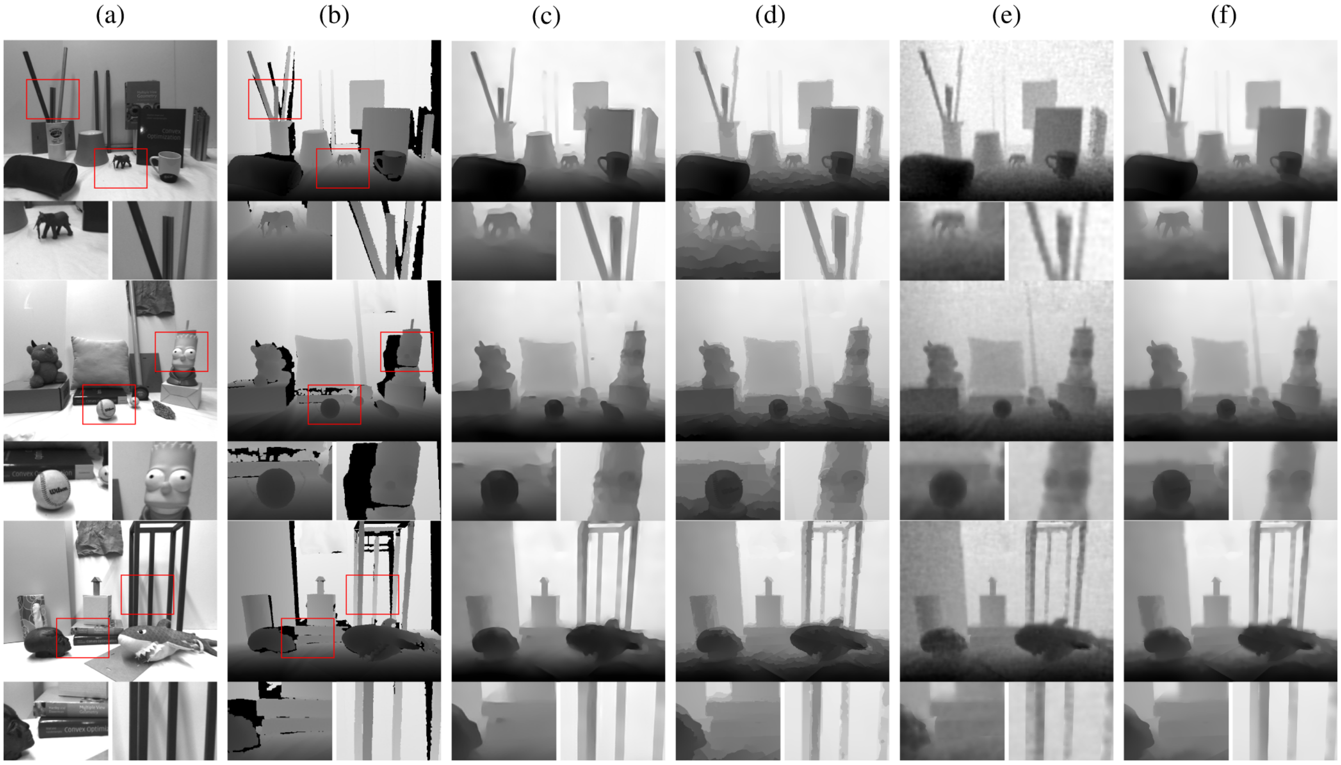

Fig. 9 Visual comparison of results obtained with different algorithms for images from the ToF dataset.[9] To allow the differences to be viewed more clearly, for each full image, zoomed-in views of select regions (shown by red rectangles) are also included. The column labeled "proposed" shows the results for the proposed algorithm directly applied to the observed noisy low-resolution depth map and the column labeled "proposed*" shows the results obtained by using the proposed algorithm with a denoised low-resolution depth map as the input (corresponding to "denoised init. 2" in Table 3). (a) Intensity image, (b) GT high resolution depth ground-truth, unsampled and hole-filled versions obtained with (c) DGDE,[20] (d) RCG,[17] (e) proposed, and (f) proposed*.

depth maps that is encountered with some capture technologies. This is the case, for instance, with the dataset of Ferstl et al.,[9] where the low-resolution depth data has been captured with an actual ToF camera and not obtained synthetically by downsampling high-resolution depth data. The numerical performance of the alternative techniques on this dataset is summarized in Table 3. Because the proposed technique is not particularly designed to handle noise, using the proposed method directly to upsample the noisy low-resolution depth maps does not perform very well [the results for this case are indicated in the rows labeled "proposed (noisy init)" in Table 3]. The performance for the proposed technique can, however, be improved by denoising the low-resolution depth map data prior to upsampling—an approach that has also been used previously (and in the results obtained in this section) with the DGDE[20] method. Results obtained by using denoised low-resolution depth maps as the input are also included in Table 3 and are indicated by the qualifying label "denoised init." These results indicate that the method can provide results comparable with the state-of-the-art methods when the input data are denoised prior to upsampling. Visual comparisons corresponding to the better performing methods in Table 3 are presented in Fig. 9. From examining these, one can see that even the best performing methods have relatively large errors compared with the results in Fig. 4. An enhancement of the proposed method to comprehensively handle upsampling for noisy low-resolution depth data is therefore a direction worthy of further investigation but beyond the scope of the current paper. It is worth noting here that the technology of ToF cameras is undergoing rapid technological advance, which should help reduce noise.

Table 3 Quantitative comparison of the performance of different methods on the ToF dataset.[9] The table lists MAE depth values (in millimeter).

	Images		
	Books	Devil	Shark
Bilinear	17.21	17.49	19.01
IBL[5]	15.41	16.48	17.14
TGV[9]	13.51	14.60	15.11
AD[31]	15.35	16.17	17.09
DGDE[20] (denoised init)	13.38	15.58	15.65
RCG[17]	13.57	14.62	15.74
Proposed (noisy init)	15.82	16.32	17.18
Proposed (denoised init 1)	14.40	15.71	16.54
Proposed (denoised init 2)	13.46	14.68	15.56

We also have a couple of additional observations regarding the relation of the proposed algorithm to Levin et al.'s matting method[23] and He et al.'s fast implementation.[25] Levin et al.'s method formulates the matting map (alpha channel) as a weighted linear combination of neighboring pixel values. He et al.'s method is based on Levin et al.'s formulation and also proposes the acceleration using the conjugate gradient method. In our problem, we model the depth as a linear function of local spatial coordinates instead of pixel values directly, and we use a local filtered conjugate gradient formulation instead of using an adjacency matrix.

The analysis presented in Sec. 3.3 of this paper brought to light how using preprocessed hole-filled data as "ground truth" may introduce potential bias in the evaluation of alternative methods for upsampling. Evaluation of only nonhole-filled regions, as has been done in the results reported in Sec. 3.1, eliminates the potential for such bias. Another alternative would be to consider ground truth from simulations where all the data can be intrinsically obtained and there are no holes to be filled. Physically based photorealistic renders, such as Mitsuba,[35] may offer one option for the generation of simulated ground-truth RGB-D images from 3-D models. Evaluation of ground-truth datasets can be constructed from multiple meaningful perspectives that address one or more of the concerns regarding available data, for instance, the dynamic range represented in the depth maps, the diversity of scene content, and perhaps, to include hyperspectral images beyond RGB channels. A potential issue introduced by simulated photorealistic images is whether synthesized images are statistically representative of natural scenes. Particularly, synthesized images may use polygon-based mesh representations of surfaces, which may introduce their own artifacts. The methodology therefore requires careful consideration and is also worthy of further independent study.

4 Conclusion

The algorithm proposed in this paper provides an effective method for joint depth map upsampling and hole filling on large images. Experiments demonstrate that the proposed method offers an improvement over the current state-of-the-art methods. The proposed memory-efficient implementation significantly reduces the memory requirement making the approach feasible on typical workstation hardware. In addition to presenting a novel joint hole filling and depth map upsampling approach, the paper also provides valuable statistical analysis that highlights the fact that prior assessments of depth upsampling using preprocessed data as "ground truth" suffer from potential bias: the assessments are likely evaluating conformance with the hole-filling method used in the preprocessing rather than accuracy of the upsampling. An implementation of the proposed algorithm can be accessed and executed through Code Ocean: https://codeocean.com/capsule/5103691/tree/v1.

5 Appendix A: Detailed Derivation of the Solution Approach

Applying the first-order optimality conditions to Eq. (4), we see that the optimum solution satisfies

$$\frac{d\{\boldsymbol{W}_j[\boldsymbol{d}_{H,\mathcal{N}(j)}-\boldsymbol{G}_j\boldsymbol{p}_j^\intercal]\}^2}{d\boldsymbol{p}_j}=\boldsymbol{W}_j[\boldsymbol{d}_{H,\mathcal{N}(j)}-\boldsymbol{G}_j\boldsymbol{p}_j^\intercal]\boldsymbol{G}_j^\intercal=\boldsymbol{0}. \tag{18}$$

The solution to this linear equation is obtained as

$$\boldsymbol{p}_j=(\boldsymbol{G}_j^\intercal\boldsymbol{W}_{0,j}^\intercal\boldsymbol{G}_j)^{-1}\boldsymbol{G}_j^\intercal\boldsymbol{W}_{0,j}^\intercal\boldsymbol{d}_{H,\mathcal{N}(j)}. \tag{19}$$

The expression for the objective function Q in Eq. (6) is obtained as

$$\begin{aligned}
Q&=\sum_{j=1}^{N}\{\boldsymbol{W}_j[\boldsymbol{d}_{H,\mathcal{N}(j)}-\boldsymbol{G}_j(\boldsymbol{G}_j^\intercal\boldsymbol{W}_{0,j}^\intercal\boldsymbol{G}_j)^{-1}\boldsymbol{G}_j^\intercal\boldsymbol{W}_{0,j}^\intercal\boldsymbol{d}_{H,\mathcal{N}(j)}^\intercal]\}^2\\
&\quad+\lambda F_j\\
&=\sum_{j=1}^{N}([\boldsymbol{W}_j\boldsymbol{d}_{H,\mathcal{N}(j)}]\{\boldsymbol{E}-\boldsymbol{G}_j[(\boldsymbol{G}_j^\intercal\boldsymbol{W}_{0,j}^\intercal\boldsymbol{G}_j)^{-1}\boldsymbol{G}_j^\intercal\boldsymbol{W}_{0,j}^\intercal]^\intercal\})^2\\
&\quad+\lambda F_j\\
&=\sum_{j=1}^{N}\boldsymbol{d}_{H,\mathcal{N}(j)}^\intercal(\overline{\boldsymbol{G}}_j^\intercal\boldsymbol{W}_{0,j}\overline{\boldsymbol{G}}_j)\boldsymbol{d}_{H,\mathcal{N}(j)}+\lambda F_j\\
&=\sum_{j=1}^{N}\boldsymbol{d}_{H,\mathcal{N}(j)}^\intercal(\overline{\boldsymbol{G}}_j^\intercal\boldsymbol{W}_{0,j}\overline{\boldsymbol{G}}_j)\boldsymbol{d}_{H,\mathcal{N}(j)}+\sum_{j=1}^{M}\lambda(d_{L,j}-d_{H,j})^2.
\end{aligned} \tag{20}$$

6 Appendix B: Details of Memory Efficiency Implementation

The expression in Eq. (13) can be obtained from Eq. (10) as

$$\begin{aligned}
(\boldsymbol{Lq})_i&=\sum\delta_{ij}w_{ki}q_i-\boldsymbol{g}_i^\intercal\sum_{k\in\mathcal{N}(i)}\boldsymbol{a}_k w_{ki}+\sum_{k\in\mathcal{N}(i)}b_k w_{ki}\\
&=\sum\delta_{ij}w_{ki}q_i-\sum_{k\in\mathcal{N}(i)}(\boldsymbol{g}_i^\intercal-\boldsymbol{g}_k^\intercal)\boldsymbol{a}_k w_{ki}\\
&=\sum\delta_{ij}w_{ki}q_i\\
&\quad-\sum_{k\in\mathcal{N}(i)}(\boldsymbol{g}_i^\intercal-\boldsymbol{g}_k^\intercal)\left\{\boldsymbol{C}_k\left[\sum_{j\in\mathcal{N}(k)}w_{kj}\boldsymbol{g}_j^\intercal q_j-k\bar{q}_k\right]\right\}w_{ki}\\
&=\sum\delta_{ij}w_{ki}q_i\\
&\quad-\sum_{k\in\mathcal{N}(i)}(\boldsymbol{g}_i^\intercal-\boldsymbol{g}_k^\intercal)\left\{\boldsymbol{C}_k\left[\sum_{j\in\mathcal{N}(k)}w_{kj}(\boldsymbol{g}_j^\intercal-\boldsymbol{g}_k^\intercal)q_j\right]\right\}w_{ki}\\
&=\sum_j\left\{\sum_{k|(i,j)\in\mathcal{N}(k)}[\delta_{ij}w_{ki}\right.\\
&\quad\left.-w_{ki}w_{kj}(\boldsymbol{g}_i-\boldsymbol{g}_k)^\intercal\boldsymbol{C}_k(\boldsymbol{g}_j-\boldsymbol{g}_k)]\right\}q_j.
\end{aligned} \tag{21}$$

7 Appendix C: Performance for Alternative Weighting Functions

Table 4 compares the results obtained for the proposed scheme with the alternative weighting functions that are defined in Sec. 2.4. The results highlight the importance of using depth similarity. Weights considering only color similarity, which include Gaussian profile, Laplacian profile, and max channel, are close to each other in performance. However, the joint color-depth similarity weighting provides significantly better results for all the test cases.

Table 4 Quantitative comparison of different weighting functions for the proposed scheme. MAE depth-disparity values are reported for the images in the Middlebury dataset[27] for four different upsampling ratios, as listed in the second row. The best result for each case is highlighted in bold font.

	Images											
	Art				Books				Moebius			
	Sample rate											
Methods	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×
Bicubic	0.8965	1.4298	2.4363	4.3456	0.7911	1.0842	1.7031	2.5419	0.6855	1.0287	1.5821	2.5527
Bilinear	0.7642	1.2300	2.1495	3.9500	0.6620	0.8993	1.4183	2.1174	0.5685	0.8578	1.3347	2.1942
Gaussian	0.5203	0.6552	1.4717	3.6534	0.2723	0.4235	0.6080	1.4223	0.3770	0.4609	0.6656	1.1064
Laplacian	0.4796	0.7499	2.4235	7.2141	0.1859	0.3175	1.1089	3.771	0.1766	0.2980	1.1473	3.5967
MaxChannel	0.5339	1.0258	1.7853	3.3673	0.2070	0.3816	0.6485	1.1029	0.2069	0.3743	0.6202	1.1433
Color + depth	**0.2744**	**0.6612**	**1.3049**	**2.6243**	**0.1671**	**0.3067**	**0.5207**	**0.9030**	**0.1714**	**0.3396**	**0.5328**	**1.0317**

8 Appendix D: Evaluation of Methods on Potentially Biased Ground Truth

Table 5 summarizes the potentially biased numerical performance metrics for the different methods obtained by regarding preprocessed, hole-filled data as ground truth. Other than the ground-truth data, the evaluation procedure is identical to the one used in Table 1. While the reported (potentially biased) MAE values in Table 5 for the proposed method are only slightly higher than the corresponding values in Table 4 and comparable with the best results, the relative performance of the different methods shows differences against what is reported in Table 4 in Sec. 3.3.

Acknowledgments

We thank the Center for Integrated Research Computing, University of Rochester, for providing access to computational resources. We are also grateful to the anonymous reviewers for several constructive comments that have significantly improved the paper. A workshop paper[32] featured results from preliminary work leading to this paper. Compared to the prior publication, the paper here is more comprehensive in exposition and includes improvements and enhancements. Specifically, new elements in this paper include: the joint use of depth and color information for upsampling in the proposed formulation, the memory-efficient

Table 5 Quantitative comparison of the potentially biased performance for the different algorithms using the hole-filled data as ground truth. MAE depth-disparity values are reported for the images in the Middlebury dataset[27] for four different upsampling ratios, as listed in the second row. The best result for each case is highlighted in bold font.

	Images											
	Art				Books				Moebius			
	Sample rate											
Methods	2×	4×	8×	16×	2×	4×	8×	16×	2×	4×	8×	16×
Bicubic	0.89	1.43	2.44	4.35	0.79	1.08	1.70	2.54	0.69	1.03	1.58	2.55
Bilinear	0.76	1.23	2.15	3.95	0.66	0.90	1.42	2.12	0.57	0.86	1.33	2.19
IBL[5]	0.57	0.70	1.50	3.69	0.30	0.45	0.64	1.45	0.39	0.48	0.69	1.14
TGV[9]	0.51	0.78	2.46	7.27	0.60	0.75	2.31	6.32	0.19	0.32	1.19	3.64
AD[31]	**0.01**	**0.63**	1.57	3.61	**0.00**	**0.24**	**0.51**	1.01	**0.00**	**0.27**	0.66	1.50
DGDE[20]	0.43	0.96	1.85	3.62	0.38	0.50	0.71	1.21	0.39	0.50	0.74	1.24
RCG[17]	0.45	0.73	**1.30**	**2.48**	0.19	0.31	**0.51**	**0.93**	0.19	0.33	**0.50**	**1.00**
Color[32]	0.47	0.91	1.82	3.68	0.21	0.37	0.67	1.24	0.20	0.36	0.67	1.23
Color + depth	0.31	0.73	1.42	2.80	0.19	0.34	0.53	1.02	0.19	0.37	0.59	1.10

implementation, and the statistical analysis for using preprocessed data as ground truth.

References

1. L. Ding and G. Sharma, "Fusing structure from motion and lidar for dense accurate depth map estimation," in *IEEE Int. Conf. Acoust., Speech, and Signal Process.*, pp. 1283–1287 (2017).
2. J. M. Coughlan and A. L. Yuille, "The Manhattan world assumption: regularities in scene statistics which enable Bayesian inference," in *Adv. Neural Inf. Process. Syst.*, pp. 845–851 (2001).
3. J. M. Coughlan and A. L. Yuille, "Manhattan world: orientation and outlier detection by Bayesian inference," *Neural Comput.* **15**(5), 1063–1088 (2003).
4. J. Diebel and S. Thrun, "An application of Markov random fields to range sensing," in *Adv. Neural Inf. Process. Syst.*, pp. 291–298 (2005).
5. Q. Yang et al., "Spatial-depth super resolution for range images," in *IEEE Int. Conf. Comput. Vision and Pattern Recognit.*, pp. 1–8 (2007).
6. K. He, J. Sun, and X. Tang, "Guided image filtering," *Lect. Notes Comput. Sci.* **6311**, 1–14 (2010).
7. J. Kopf et al., "Joint bilateral upsampling," *ACM Trans. Graphics* **26**(3), 96 (2007).
8. J. Park et al., "High quality depth map upsampling for 3D-TOF cameras," in *IEEE Int. Conf. Comput. Vision*, pp. 1623–1630 (2011).
9. D. Ferstl et al., "Image guided depth upsampling using anisotropic total generalized variation," in *IEEE Int. Conf. Comput. Vision*, pp. 993–1000 (2013).
10. J. Yang et al., "Color-guided depth recovery from RGB-D data using an adaptive autoregressive model," *IEEE Trans. Image Process.* **23**(8), 3443–3458 (2014).
11. J. Lu et al., "A revisit to MRF-based depth map super-resolution and enhancement," in *IEEE Int. Conf. Acoust., Speech, and Signal Process.*, pp. 985–988 (2011).
12. A. Harrison and P. Newman, "Image and sparse laser fusion for dense scene reconstruction," in *Field and Service Robotics*, A. Howard, K. Iagnemma, and A. Kelly, Eds., pp. 219–228, Springer, Berlin, Heidelberg (2010).
13. K.-H. Lo, K.-L. Hua, and Y.-C. F. Wang, "Depth map super-resolution via Markov random fields without texture-copying artifacts," in *IEEE Int. Conf. Acoust., Speech, and Signal Process.*, pp. 1414–1418 (2013).
14. M.-Y. Liu, O. Tuzel, and Y. Taguchi, "Joint geodesic upsampling of depth images," in *IEEE Int. Conf. Comput. Vision and Pattern Recognit.*, pp. 169–176 (2013).
15. K. Uruma et al., "High resolution depth image recovery algorithm based on the modeling of the sum of an average distance image and a surface image," in *IEEE Int. Conf. Image Process.*, pp. 2836–2840 (2016).
16. Y. Dong et al., "Depth map upsampling using joint edge-guided convolutional neural network for virtual view synthesizing," *J. Electron. Imaging* **26**(4), 043004 (2017).
17. W. Liu et al., "Robust color guided depth map restoration," *IEEE Trans. Image Process.* **26**(1), 315–327 (2017).
18. W. Liu et al., "Variable bandwidth weighting for texture copy artifact suppression in guided depth upsampling," *IEEE Trans. Circuits Syst. Video Technol.* **27**(10), 2072–2085 (2017).
19. W. Liu et al., "Semi-global weighted least squares in image filtering," in *IEEE Int. Conf. Comput. Vision*, pp. 5861–5869 (2017).
20. S. Gu et al., "Learning dynamic guidance for depth image enhancement," in *IEEE Int. Conf. Comput. Vision and Pattern Recognit.*, pp. 712–721 (2017).
21. L. Feng et al., "An adaptive background biased depth map hole-filling method for Kinect," in *Annu. Conf. IEEE Ind. Electron. Soc.*, pp. 2366–2371 (2013).
22. J. Wang et al., "High accuracy hole filling for Kinect depth maps," *Proc. SPIE* **9273**, 92732L (2014).
23. A. Levin, D. Lischinski, and Y. Weiss, "A closed-form solution to natural image matting," *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(2), 228–242 (2008).
24. R. Merris, "Laplacian matrices of graphs: a survey," *Linear Algebra Appl.* **197**, 143–176 (1994).
25. K. He, J. Sun, and X. Tang, "Fast matting using large kernel matting Laplacian matrices," in *IEEE Int. Conf. Comput. Vision and Pattern Recognit.*, pp. 2165–2172 (2010).
26. C. Yu, G. Sharma, and H. Aly, "Computational efficiency improvements for image colorization," *Proc. SPIE* **9020**, 902004 (2014).
27. D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *IEEE Int. Conf. Comput. Vision and Pattern Recognit.*, pp. 195–202 (2003).
28. D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *IEEE Int. Conf. Comput. Vision and Pattern Recognit.*, pp. 1–8 (2007).
29. H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *IEEE Int. Conf. Comput. Vision and Pattern Recognit.*, pp. 1–8 (2007).
30. D. Scharstein et al., "High-resolution stereo datasets with subpixel-accurate ground truth," *Lect. Notes Comput. Sci.* **8753**, 31–42 (2014).
31. J. Liu and X. Gong, "Guided depth enhancement via anisotropic diffusion," *Lect. Notes Comput. Sci.* **8294**, 408–417 (2013).
32. Y. Zhang, L. Ding, and G. Sharma, "A local-linear-fitting-based matting approach for accurate depth upsampling," in *IEEE Western New York Image and Signal Process. Workshop (WNYISPW)*, pp. 1–5 (2016).
33. C. Shen, "Matlab toolbox for depth enhancement using color information," https://bitbucket.org/chhshen/depth-enhancement (August 2017).
34. B. L. Welch, "The significance of the difference between two means when the population variances are unequal," *Biometrika* **29**, 350–362 (1938).
35. W. Jakob, "Mitsuba renderer," 2010, http://www.mitsuba-renderer.org (August 2019).

Yanfu Zhang received his MS degree in electrical and computer engineering from the University of Rochester, New York, in 2017, his MS degree in optical engineering from Chinese Academy of Sciences, Changchun, China, in 2015, and his BS degree in electronics information from the University of Science and Technology of China, Hefei, China, in 2012, respectively. Currently, he is pursuing his PhD in electrical and computer engineering at the University of Pittsburgh, Pennsylvania. His research interests include machine learning and the applications on computational bioinformatics and image processing.

Li Ding received his MS degree in electrical and computer engineering from the University of Rochester in 2017 and his BS degree in electrical engineering from Harbin Institute of Technology, Harbin, China, in 2015. Currently, he is pursuing his PhD in the electrical and computer engineering at the University of Rochester. His research interests span the field of computer vision, digital image processing, and machine learning, with a focus on three-dimensional (3-D) range data processing and 3-D reconstruction. He is the recipient of the best student paper award at the 2016 Western New York Image and Signal Processing Workshop.

Gaurav Sharma is a professor at the University of Rochester in the Department of Electrical and Computer Engineering, in the Department of Computer Science, and in the Department of Biostatistics and Computational Biology. His research interests include computer vision, image processing, color science and imaging, and multimedia security and watermarking, and bioinformatics. He is the editor of the *Color Imaging Handbook*, published by CRC Press in 2003. He is a fellow of SPIE, of the Society of Imaging Science and Technology (IS&T), and of the IEEE, and a member of Sigma Xi. From 2011 to 2015, he served as the editor-in-chief for the *Journal of Electronic Imaging* and is currently serving as the editor-in-chief for the *IEEE Transactions on Image Processing*.