Joint Geodesic Upsampling of Depth Images

Ming-Yu Liu Oncel Tuzel Yuichi Taguchi
Mitsubishi Electric Research Labs (MERL), Cambridge, MA, USA
{mliu,oncel,taguchi}@merl.com

Abstract

We propose an algorithm utilizing geodesic distances to upsample a low resolution depth image using a registered high resolution color image. Specifically, it computes depth for each pixel in the high resolution image using geodesic paths to the pixels whose depths are known from the low resolution one. Though this is closely related to the all-pair-shortest-path problem which has $O(n^2 \log n)$ complexity, we develop a novel approximation algorithm whose complexity grows linearly with the image size and achieve real-time performance. We compare our algorithm with the state of the art on the benchmark dataset and show that our approach provides more accurate depth upsampling with fewer artifacts. In addition, we show that the proposed algorithm is well suited for upsampling depth images using binary edge maps, an important sensor fusion application.

1. Introduction

Depth sensors have emerged as an important tool for 3D scene understanding. They have been applied to various applications and gradually shaped the way people interact with machines. However, unlike the conventional optical camera, the resolution of depth sensors advances at a much slower pace. This is mainly due to the manufacturing cost of main sensor elements. While the resolution of mainstream optical cameras is in the order of 10 megapixels, the resolution of mainstream time-of-flight depth sensors is still lower than 0.02 megapixels, which greatly limits their applications.

One way to improve the resolution of depth images is to use a high resolution optical camera in tandem with the depth sensor. In general, geometric and color boundaries of a scene are correlated: abrupt depth transition often leads to abrupt color transition. Therefore, it is possible to leverage this correlation to *upsample* the depth images. In this paper, we study depth image upsampling problem using registered color images and propose a new algorithm based on geodesic curves in joint color and depth images.

Our algorithm is inspired by the joint bilateral upsampling algorithm [11] (current state of the art in terms of both accuracy and efficiency), which interpolates low resolution depths on the high resolution grid based on a set of weights computed as multiplications of spatial and color kernels. These kernels utilize Euclidean distance to quantify the dissimilarities of the pixels and the word "joint" is due to utilization of two channels: optical image for computing color distance and depth image for computing spatial distance. We argue that using two separate kernels causes blurry depth boundaries and depth bleeding artifacts particularly when the colors of the surfaces across the depth boundaries are similar. In addition, fine scene structures are usually not preserved during upsampling.

Geodesic curves are shortest paths on the image grid. We compute geodesic distances—lengths of the geodesic curves—from each pixel in the target high resolution depth image to all the pixels whose depths are known from the low resolution depth image. These distances are used to propagate the known depths to the high resolution grid in a smooth and depth discontinuity preserving manner. Since the geodesic distance integrates joint color and spatial changes along the curve, it is sensitive to thin contours around the surfaces, providing sharp depth boundaries even when the color difference between two sides of a contour is subtle. In addition, the geodesic path can follow thin segments with uniform colors and therefore produce high quality depth images with fine details. An example motivating the use of geodesic distances for depth upsampling is illustrated in Figure 1 (see Section 2 for further details).

Computing geodesic distances is closely related to the all-pair-shortest-path problem, which is computationally expensive. For real-time processing, we propose a new approximation algorithm for simultaneously finding K nearest (in geodesic sense) nodes from each source node and show that its complexity grows linearly with the image size and K. We conduct extensive experimental validation on the Middlebury dataset and compared the proposed algorithm with the state of the art. The result shows that the proposed algorithm produces more accurate high resolution depth images for both smooth surfaces and boundary re-

1063-6919/13 $26.00 © 2013 IEEE
DOI 10.1109/CVPR.2013.29

gions. We also show that our method is well suited for upsampling depth images using binary edge maps (e.g. provided by a multi-flash camera [15]), which is a difficult task for most of the existing methods.

1.1. Related Work

Depth upsampling methods can be categorized as global or local. Global methods [6, 14, 19] formulate depth upsampling as an optimization problem where a large cost is induced if two neighboring pixels having a similar color are assigned two very different depths. These methods can produce accurate depth images but are generally too slow for real-time applications.

Local methods [11, 22, 7] are based on filtering and can often obtain real-time performance. Among them, joint bilateral upsampling is particularly popular, which uses bilateral filtering in a joint color-spatial space. The proposed algorithm is also based on filtering where the upsampling filter weights are determined by geodesic distances.

Geodesic distances were previously applied to the colorization [24], image matting [2], and image de-nosing and editing [5] problems. Here, we define joint geodesic filtering in a color-spatial space and show its application for depth upsampling. The fast marching algorithm [23] and the geodesic distance transform [18] are two common implementations for geodesic distance computation, both of which have a linear time complexity. Still, they are not fast enough to compute all pair shortest paths needed for the proposed algorithm. We derive a novel approximation algorithm for simultaneously finding K nearest nodes from each source node based on geodesic distance transforms and achieve real-time performance. Recently, [8] proposed using geodesic distances to compute Voronoi cells for image tessellation, which are used for fitting planes to sparse depth measurements for depth interpolation. This algorithm has a quadratic time complexity and is a magnitude slower than the proposed algorithm.

1.2. Contributions

The contributions of the paper are listed below.

- We propose a new joint filtering algorithm using geodesic distances for upsampling a depth image using a registered high resolution color image.
- We develop a fast optimization technique for finding approximate K nearest nodes based on geodesic distance and achieving real-time upsampling performance.
- We compare the proposed algorithm with the state of the art and show that it has superior performance, especially in depth discontinuity regions.

The rest of the paper is organized as follows. In Section 2, we present the depth upsampling formulation. Section 3 discusses the optimization technique. Experiment results are given in Section 4. Section 5 concludes the paper.

2. Joint Geodesic Upsampling

Let $D_{\downarrow}$ and I be the low resolution depth and high resolution optical images, respectively, where the resolution of I is r times larger than that of $D_{\downarrow}$. The two images are registered so that each grid point in $D_{\downarrow}$ can be mapped to a unique grid point in I. Without loss of generality, we can assume that a simple mapping exists where the pixel at location (i, j) in $D_{\downarrow}$ is mapped to the pixel at location (ir, jr) in I. Our goal is to construct a high resolution depth image D whose resolution is equal to that of I.

Depths of pixels in a sparse grid in D are known (which we refer to as *seed pixels*) from the corresponding low resolution depth image $D_{\downarrow}$. They are used to fill in the depths of the other pixels in D. We adopt a filtering-based approach such that the depth for a pixel $\mathbf{x}$ in D is given by

$$D(\mathbf{x}) = \sum_{\mathbf{y}_{\downarrow}} g(\mathbf{x}, \mathbf{y}) D_{\downarrow}(\mathbf{y}_{\downarrow}) \tag{1}$$

where $\mathbf{y}$ and $\mathbf{y}_{\downarrow}$ are the corresponding coordinates of the seed pixels in the high and low resolution images respectively, and g is a kernel function measuring the affinity between the two coordinates in the argument. The joint bilateral filtering algorithm [11] uses a similar formulation where its kernel function is a product of two Gaussian kernels given by

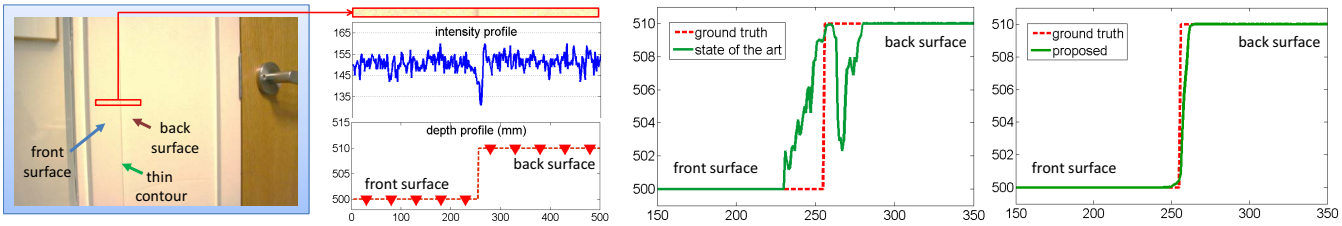
$$\exp(\frac{-||\mathbf{x}-\mathbf{y}||_2^2}{2\sigma_D^2}) \exp(\frac{-||I(\mathbf{x})-I(\mathbf{y})||_2^2}{2\sigma_R^2}) \tag{2}$$

where σ_D and σ_R are the kernel bandwidth parameters for spatial distance $||\mathbf{x}-\mathbf{y}||_2$ and color distance $||I(\mathbf{x})-I(\mathbf{y})||_2$ respectively.

We propose computing the affinity measure between two pixels using geodesic distances defined on the image grid[1], which can be considered as a two dimensional embedding in a joint color-spatial space. Let p be a path joining $\mathbf{x}$ and $\mathbf{y}$. Note that such path can be uniquely represented by the sequence of pixels it traverses, $\mathbf{x} = \mathbf{x}_p^{(1)}, \mathbf{x}_p^{(2)}, \mathbf{x}_p^{(3)}, \ldots, \mathbf{x}_p^{(|p|)} = \mathbf{y}$, where $|p|$ denotes the number of pixels in the sequence. Let $\mathcal{P}$ be the set of all the paths joining $\mathbf{x}$ and $\mathbf{y}$; $p \in \mathcal{P}$. The geodesic distance between $\mathbf{x}$ and $\mathbf{y}$ is given by the length of the shortest path;

$$d_{\mathrm{G}}(\mathbf{x}, \mathbf{y}) = \min_{p \in \mathcal{P}} \sum_{i=2}^{|p|} \left(\frac{1}{r} ||\mathbf{x}_p^{(i)} - \mathbf{x}_p^{(i-1)}||_2 + \lambda ||I(\mathbf{x}_p^{(i)}) - I(\mathbf{x}_p^{(i-1)})||_2 \right) \tag{3}$$

[1] We assume the image grid is 8-connected, i.e., each pixel is only connected to its 8 neighbors via graph edges.



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

Figure 1. **Comparison between joint bilateral upsampling and joint geodesic upsampling.** (a) An image of an indoor scene consisting of a wall, a door, and a window. The door and window frames have the same color as the wall. (b) A horizontal slice of the image and its intensity and depth profiles. The triangles indicate the locations where depth measurements are collected using a low resolution depth sensor. Note that there is a thin dark band at the location of the occluding boundary of the frame and the wall, due to shading of the rounded surface edge. (c) The upsampled depth profile using joint bilateral upsampling. (d) The upsampled depth profile using the proposed joint geodesic upsampling algorithm. The proposed method integrates color changes along the geodesic path and accurately recovers high resolution depth profile, whereas joint bilateral upsampling smooths depths across occlusion boundary resulting in blurring effect.

where λ is a weighting parameter. We use the Gaussian kernel to convert the geodesic distance into the affinity measure:

$$g_{\mathrm{G}}(\mathbf{x},\mathbf{y}) = \exp\left(\frac{-d_{\mathrm{G}}^2(\mathbf{x},\mathbf{y})}{2\sigma^2}\right) \tag{4}$$

where σ is the kernel bandwidth parameter.

Figure 1 shows a 1D illustration comparing joint bilateral upsampling and joint geodesic upsampling. Although the front and back surfaces have similar colors (Figure 1a), there is a very thin dark band at the location of the occlusion boundary[2] (Figure 1b). Joint geodesic upsampling integrates color changes along the geodesic curves; therefore it is sensitive to thin structures and fine scale changes, producing smooth surfaces with sharp occlusion boundaries (Figure 1d). In contrast, the joint bilateral upsampling algorithm incorrectly propagates depths across the depth boundary due to Euclidean color distance computation (Figure 1c).

3. Fast Geodesic Upsampling Computation

The upsampling formulation in (1) requires computation of shortest paths from each pixel to all the seed pixels, which is equivalent to the all-pair-shortest-path problem. The best known exact algorithm for this problem has $O(n^2 \log n)$ complexity where n is the image size: $O(n \log n)$ time complexity for computing single source shortest paths to all the pixels and it is repeated n times for each pixel. Even the best known approximation algorithm with a constant bound [17] has $O(n^{\frac{3}{2}})$ complexity, which is still prohibitive for real-time applications.

Here we derive an approximate formulation of the upsampling operation and present an $O(Kn)$ algorithm which achieves real-time performance. There are two assumptions involved in our approximation. First, we make the assumption that to compute the depth of a pixel it is sufficient to propagate information from its K "nearest" depth pixels. Note that the "nearest" is defined in the geodesic sense. Let $\mathcal{K}(\mathbf{x})$ be the set of the K nearest seed pixels for a pixel $\mathbf{x}$. The K-nearest approximation to geodesic upsampling (4) is given by

$$D(\mathbf{x}) = \sum_{\mathbf{y}_\downarrow \in \mathcal{K}(\mathbf{x})} g_G(\mathbf{x},\mathbf{y}) D_\downarrow(\mathbf{y}_\downarrow). \tag{5}$$

Our second assumption is that if the two seed pixels are spatially far away, they are unlikely to be simultaneously in the set of K nearest depth pixels of a given pixel. Empirically this assumption largely holds which we analyze in the experiments section.

Our algorithm consists of three major processing steps, as summarized in Figure 2: 1) we demultiplex the pixels into K channels, 2) for each channel we compute geodesic distance transform, and 3) we interpolate the depths according to computed geodesic distances. Below we detail each step.

1) **Demultiplexing**: For an input low resolution depth image, we first partition its pixels into K separate channels which is called demultiplexing. Based on our second assumption, the best demultiplexing strategy is uniform partitioning of the image grid. Let δ be the interval of this demultiplexing, which corresponds to the distance between two consecutive pixels in a given channel on the low resolution grid. Note that these two pixels map to the seed pixels on the high resolution grid, which are $L = \delta r$ pixel away from each other. Then, the first channel consists of grid points (iL, jL) in the high resolution grid where i and j are nonnegative integers, the second channel consists of grid points $(iL, r + jL)$, and so forth. Note that $K = \delta^2$.

2) **Geodesic Distance Transform (GDT)**: For each channel, we compute the geodesic distance transform [18], which provides the shortest distance from each pixel to the nearest seed point. Let S_k be the set of seed pixels in the

[2]Most of the man-made and natural surfaces do not have sharp edges, but exhibit rounded edges with very high curvature. A thin contour around the occlusion boundaries is almost always visible in high resolution images due to changing illumination around these high curvature boundaries and cast shadows.

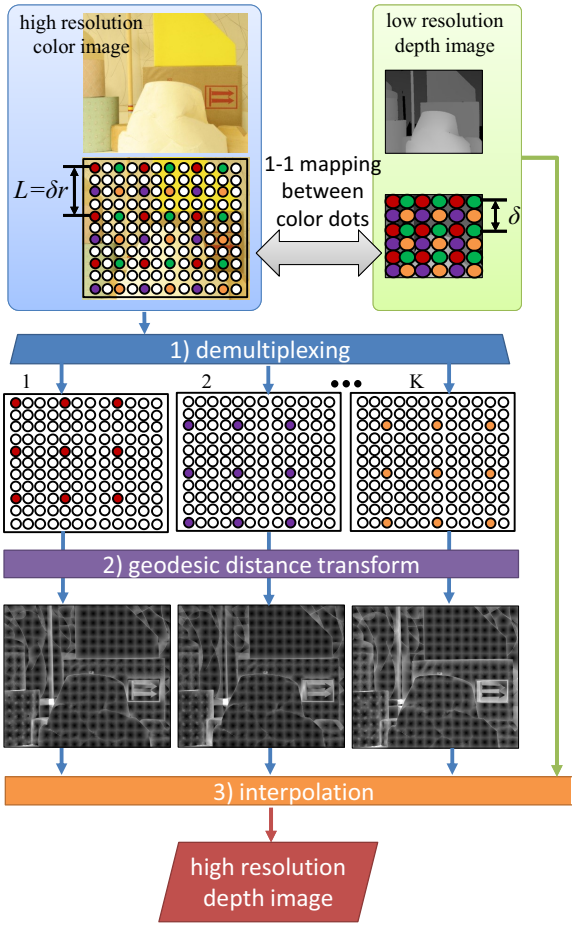


Figure 2. **Illustration of the fast geodesic upsampling computation.** 1) We demultiplex the seed pixels into K disjoint channels shown by color coding. 2) For each channel, we compute the geodesic distance transform, which provides the shortest distance from each pixel to the nearest seed pixel. 3) The depths of the seed pixels are propagated to the high resolution grid using the computed distances.

kth channel. The geodesic distance transform solves the following optimization problem

$$M_k(\mathbf{x}) = \min_{\mathbf{y} \in S_k} d_G(\mathbf{x}, \mathbf{y}) \tag{6}$$

where $d_G(\mathbf{x}, \mathbf{y})$ is defined in (3). Equation (6) can be solved efficiently with a dynamic program consisting of a sequence of forward and backward passes over the image.

The forward pass traverses the image from top-left to bottom-right where each pixel is updated according to

$$M_k(\mathbf{x}) \leftarrow \min_{\mathbf{v} \in \mathbf{V}_f} M_k(\mathbf{x}+\mathbf{v}) + d_G(\mathbf{x}, \mathbf{x}+\mathbf{v}). \tag{7}$$

The shift set of the forward pass consists of original pixel and upper-left, upper, upper-right, and left neighboring pixels, $\mathbf{V}_f = \{(0,0), (-1,-1), (0,-1), (1,-1), (-1,0)\}$.

Similarly, the backward pass traverses the image in the reverse order according to

$$M_k(\mathbf{x}) \leftarrow \min_{\mathbf{v} \in \mathbf{V}_b} M_k(\mathbf{x}+\mathbf{v}) + d_G(\mathbf{x}, \mathbf{x}+\mathbf{v}), \tag{8}$$

where the backward pass shift set is given by $\mathbf{V}_b = \{(0,0), (1,1), (0,1), (-1,1), (1,0)\}$.

Exact computation of geodesic distance transform requires multiple iterations of the forward and backward passes until convergence. It can be shown that, in pathological cases such as spiral shaped geodesic paths, the number of iterations grows by the image size, leading to super linear complexity. Yet such patterns are very rare in natural images and the average case performance can be shown to be linear. In all our experiments, convergence was achieved within ten iterations. To further speed up the algorithm, one can terminate the algorithm after a few iterations and obtain a close approximation in practice. In the experiment section, we analyze empirical performance of these cases.

3) **Interpolation**: After computing geodesic distance transform for each channel, we propagate the sparse depths given by the low resolution depth image to the high resolution grid using the computed distances. The geodesic distance transform not only provides the geodesic distance but also the nearest seed coordinate and hence its depth. Let M_k be the geodesic distance transform for channel k, where the distance from a pixel $\mathbf{x}$ to its nearest seed point in channel k is given by $M_k(\mathbf{x})$ and its coordinate is given by $\mathbf{y}^k_\downarrow(\mathbf{x})$. The approximate geodesic upsampling is then given by

$$D(\mathbf{x}) = \sum_{k=1}^{K} \exp\left(\frac{-\left(M_k(\mathbf{x})\right)^2}{2\sigma^2}\right) D_\downarrow(\mathbf{y}^k_\downarrow(\mathbf{x})). \tag{9}$$

Note that even when $K = 1$ neighbor is used for upsampling, the nearest seed pixel to a pixel is not necessarily one of its immediate neighbors in the low resolution grid. The shortest paths are defined on the high resolution grid which is sparsely covered by the seed pixels, and a path can reach a spatially distant seed pixel if the pixels along the path have similar colors. This property is important for recovering thin structures that are only apparent in the high resolution image.

4. Experiments

We conduct extensive evaluation on a benchmark dataset and compare our results with the state of the art. Parameter sensitivity, approximation quality, and computational performance of the algorithm are analyzed. We also show a new upsampling application using binary edge maps.

4.1. Quantitative and Visual Evaluation

We benchmark the performance of the proposed algorithm using the Middlebury 2005 dataset [10]. It contains

Table 1. **Quantitative comparison.** We compare the proposed algorithm with the state of the art on the Middlebury dataset using 3 performance metrics: DISC, RMS and SRMS. The DISC metric measures the error rate in the depth discontinuity region, while SRMS measures the root mean square error in the depth smooth region. RMS measures the root mean square error over the whole image.

Alg.	DISC /Rank				RMS /Rank				SRMS /Rank			
	2x	4x	8x	16x	2x	4x	8x	16x	2x	4x	8x	16x
BL	0.20 /6.1	0.39 /6.3	0.54 /6.3	0.67 /6.3	2.19 /6.1	3.37 /6.7	4.41 /6.8	5.57 /7.0	0.19 /**1.6**	0.84 /6.3	2.41 /6.8	4.24 /7.0
JBU	0.08 /3.3	0.13 /2.1	0.22 /2.1	0.38 /2.4	1.06 /**2.1**	1.52 /2.3	2.44 /3.3	3.55 /3.6	0.37 /6.0	0.49 /3.4	1.41 /3.6	2.48 /3.8
NLM	0.15 /4.9	0.20 /4.8	0.31 /4.5	0.45 /4.3	1.08 /3.4	1.56 /3.8	2.16 /3.3	3.27 /3.4	0.34 /5.0	0.46 /3.3	0.97 /3.3	2.16 /3.5
GIF	0.27 /6.9	0.40 /6.7	0.54 /6.8	0.67 /6.7	1.20 /5.0	1.60 /4.7	2.34 /4.6	3.66 /4.8	0.32 /4.2	0.70 /5.3	1.50 /5.3	2.92 /5.6
QUAD	0.08 /3.0	0.17 /3.7	0.30 /4.0	0.48 /4.6	**1.05** /2.2	**1.41 /1.3**	**1.92 /1.3**	**2.97 /1.5**	**0.18** /1.7	0.40 /2.0	0.88 /2.4	2.07 /3.0
MST	0.08 /2.5	0.16 /3.4	0.27 /3.4	0.40 /2.8	2.51 /6.9	3.13 /6.3	3.61 /6.0	3.92 /5.0	0.46 /6.2	0.91 /6.3	1.65 /5.6	2.40 /3.8
Prop.	**0.07 /1.3**	**0.11 /1.0**	**0.19 /1.0**	**0.33 /1.0**	1.07 /2.3	1.52 /3.1	2.05 /2.7	3.05 /2.7	0.30 /3.4	**0.36 /1.5**	**0.67 /1.0**	**1.68 /1.3**

6 scenes, namely art, books, dolls, laundry, moebius, and reindeer. Each scene contains two views, and each view contains a registered pair of color and depth[3] images. We generate the low resolution depth images by downsampling the original ones with the downsampling rate varying from 2x to 16x. The task is to upsample the low resolution depth image to the original resolution using the registered high resolution color image.

Metrics: We use three performance metrics for evaluation: 1) the error rate on the depth discontinuity regions (DISC), 2) the root-mean-square error (RMS), and 3) the root-mean-square error in the smooth region (SRMS).

DISC measures the reconstruction error in the depth discontinuity regions, a standard metric for benchmarking stereo reconstruction algorithms [16]. We first extract depth edges in the ground truth depth image. The extracted depth edges are 1-pixel wide. We dilate them to cover both sides of the depth boundary. We compare the upsampled depth map to the ground truth *only* in the extracted discontinuity regions. A depth is denoted as erroneous if deviating from the ground truth value by more than one disparity. The final metric is given by the ratio of the number of erroneous pixels to the total number of pixels in the extracted region.

RMS is commonly used for comparing depth upsampling algorithms [11, 14]. However, we argue that RMS favors blurry depth boundaries, which produce major artifacts for depth upsampling. Square error magnifies large few pixel errors, which are commonly generated by algorithms producing sharp depth boundaries but minimized by blurry edges. Therefore, this metric is less suited for evaluation of upsampling performances.

SRMS is a modified version of RMS error where the error is computed only in the smooth region given by the complement of the extracted depth discontinuity region. This metric ensures that error in smooth regions is not dominated by few boundary pixels having large errors and is better suited for measuring upsampling accuracy in the smooth region.

Algorithms: We compare the proposed algorithm with several filtering-based depth upsampling algorithms, bilinear interpolation (BL), joint bilateral upsampling (JBU) [11], non-local means filtering (NLM) [3], minimal-spanning-tree based cost aggregation (MST) [20], and guided image filtering (GIF) [9]. In addition, a global approach based on a quadratic cost function on the image graph (QUAD) [12] is included in our comparison. Below, we briefly review these algorithms.

JBU (discussed in Section 2) is a popular choice for depth upsampling [4, 7]. We note that real-time upsampling performance can be achieved by using the fast implementations described in [21, 1].

NLM was proposed for image de-noising [3]. It assumes that small patches are repeated throughout the image. Denoising is achieved by averaging intensities of similar pixels with the similarity measured by patch matching scores. NLM was used as a system component for a recent depth upsampling work [14], which transfers depth information from similarly-colored patches.

GIF filters an image using information from a guidance image [9]. It assumes that a linear mapping exists from the guidance image to the filtered image for each small patch. For depth upsampling, a linear regression function from the color image to the depth image is estimated for each low resolution patch, which is then used to transfer higher resolution color image to the output depth image.

MST was introduced for cost aggregation in stereo matching [20]. It first constructs a minimal spanning tree whose edge weights are given by the color differences. The tree is then used to smooth the disparity cost volume. We modify the algorithm for depth upsampling by prorogating depths based on the distances on the spanning tree.

QUAD was proposed for colorization of gray-scale images [12] and recently modified for depth upsampling [19]. Upsampling is formulated as a quadratic optimization problem where the cost function enforces color and depth correlation subject to the linear constraints given by the low resolution depth image.

For the implementations that were not publicly available, we used our own implementations. These algorithms have several preference parameters. We either used the recommended values from the original papers or ran a grid search

[3]Depths are given by disparities in the dataset.

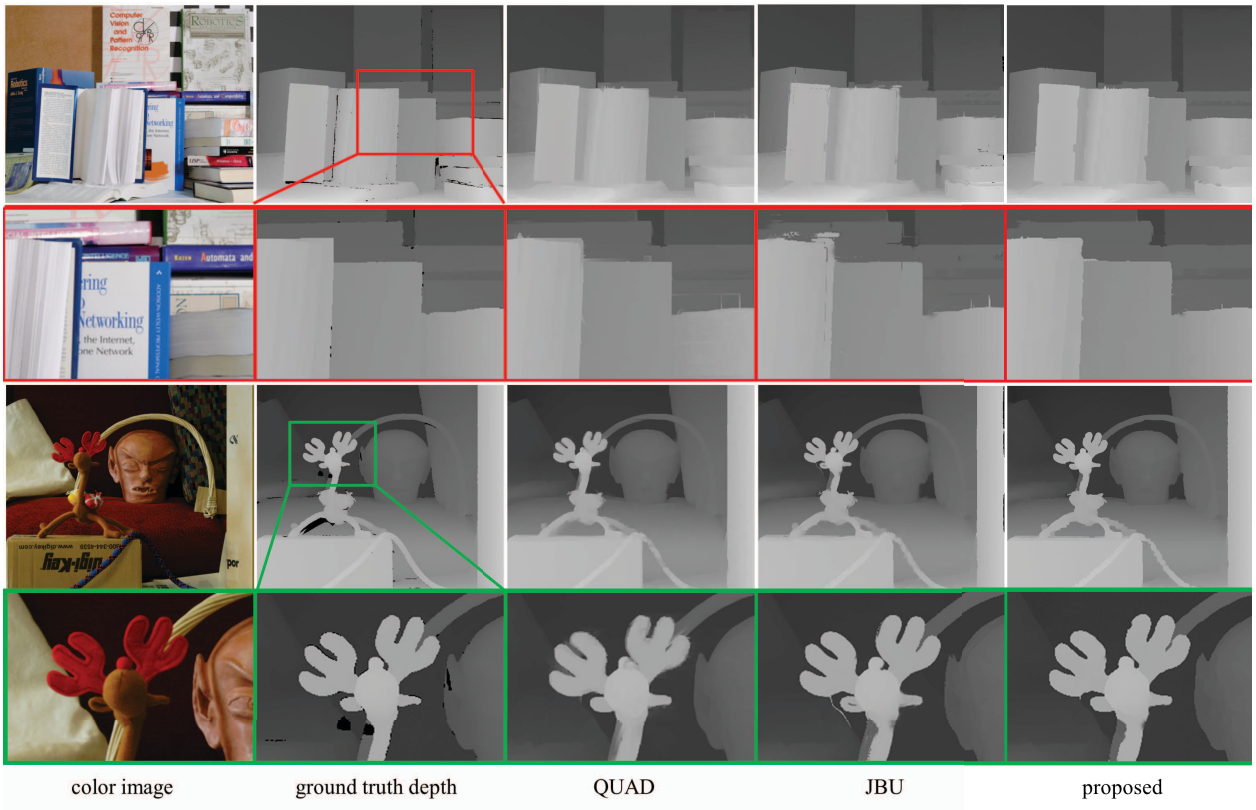


Figure 3. **Visual comparison.** Visual comparison of the depth upsampling results at 8x upsampling rate. From left to right, the columns correspond to input color image, ground truth depth images, output of the QUAD algorithm, output of the JBU algorithm, and output of the proposed algorithm. The proposed algorithm produces the sharpest depth boundaries with the fewest artifacts.

for the best parameter. For the proposed algorithm, we found that $\sigma = 0.5$ and $\lambda = 10$ provide the best performance. We set $\delta = 2$ and hence the number of nearest nodes used is $K = 4$. We analyze the parameter sensitivity of our algorithm in Section 4.2.

Results: The first column of Table 1 shows the comparison using the DISC metric. The results were averaged over all the images in the dataset at different upsampling rates. The proposed algorithm consistently yields the smallest DISC error followed by the JBU algorithm. At 4x and 8x, it reduces the error rate by 15% and 14%, respectively, as compared to the second best one. The table also shows the average ranks of the competing algorithms where the proposed algorithm has the best rank except for the dolls image in the second view.

The second column of Table 1 shows the comparison results using the RMS metric. We found that QUAD produces the best result in this metric, which is closely followed by our algorithm. As discussed above, RMS metric is misleading and is minimized by the blurry depth boundaries produced by the QUAD algorithm. This artifact is clearly visible in Figure 3, which confirms our argument. We also note that the processing time per image for the QUAD algorithm is about 25 seconds, while the proposed one only utilizes a small fraction of a second.

The third column of Table 1 shows the results using the SRMS metric. It can be seen that the proposed algorithm provides better depth recovery in the smooth regions at 4x, 8x, and 16x upsampling rates. It has 24% less SRMS error at 8x with respect to the second best one.

Figure 3 provides a visual comparison. It can be observed that the proposed algorithm produces the sharpest depth image. It also has much fewer artifacts. The results obtained by the JBU algorithm tends to include depth bleeding artifacts especially when the upsampling rate is large. This is because it fails to consider intermediate color transition between the interpolated and interpolating pixels and merely utilizes their color difference for similarity computation. Existence of a thin edge in-between has no effect on the upsampling process. Our algorithm uses geodesic paths for interpolation and is free from this drawback. We also note that the QUAD algorithm produces relatively blurred outputs as compared to the proposed one since its optimization formulation favors smooth depth transitions.

4.2. Parameter Sensitivity

Our algorithm has three parameters: two for interpolation, σ and λ, and one for approximation, K. We report

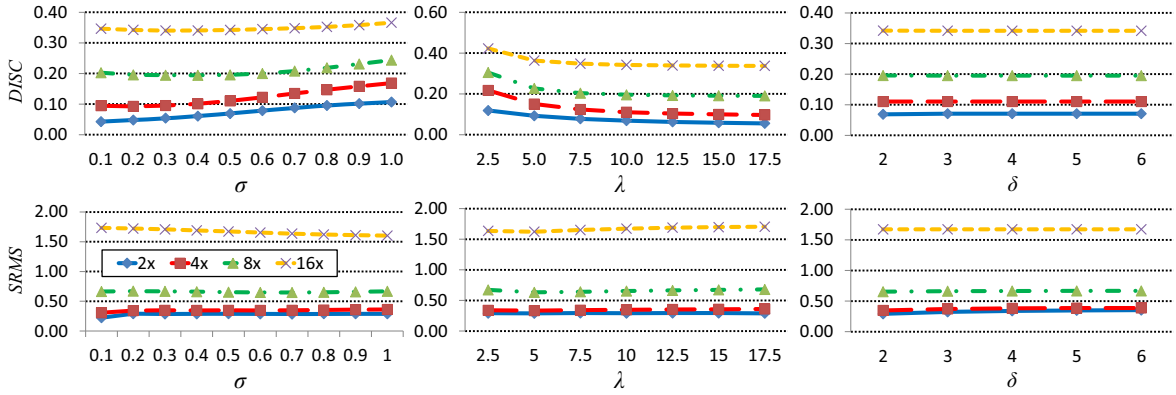


Figure 4. **Parameter sensitivity.** Accuracy is shown as a function of parameter values. The proposed algorithm renders good performance for a wide range of values. The parameters are fixed at $\sigma = 0.5$, $\lambda = 10$, $\delta = 2$ and $K = 4$ if they are not varied.

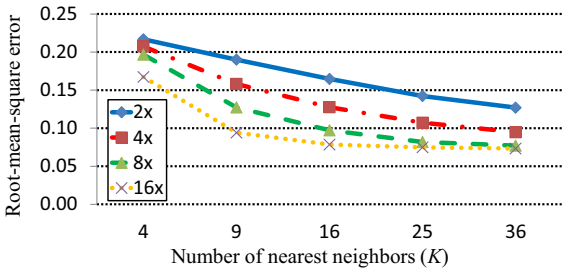


Figure 5. **Approximation analysis.** The approximation algorithm produces close results to the exact algorithm particularly for larger K. Our algorithm is exact when $K = 1$.

the performance sensitivity to the parameters in Figure 4. It shows that our algorithm does not require fine tuning of the parameters and produces good performance for a wide range of values, although extreme values could still lead to poor performance. The figure also shows that the DISC error increases as the kernel bandwidth σ increases. This is because the color difference is downplayed when σ is large, which leads to blurred boundaries. Similar effect occurs when the color difference weighting λ is decreased.

4.3. Approximation and Computation Analysis

Our optimization scheme produces an approximation to the geodesic upsampling operation as explained in Section 3. We analyzed the approximation quality by comparing it with the the exact algorithm, which find the K nearest neighbors with $O(n^2 \log n)$ complexity. The approximation error, measured using the RMS error metric, is plotted as a function of number of nearest neighbors K in Figure 5. It can be seen that the approximation algorithm produces very close results to the exact one particularly for larger K.

Table 2 reports the processing time for a 695×555 image at 8x upsampling rate. We compare three variants of the geodesic upsampling algorithm: 1) exact, 2) approximation using multi-pass geodesic distance transform, and 3) approximation using two-pass geodesic distance transform (see Section 3 for details). The approximation algorithm is about 100 times faster than the exact implementation. It has a processing speed of 3 frames per second (fps) whereas the processing time for the exact algorithm is 33 seconds for a single frame. By using the two pass approximation, the processing speed is further improved to 6 fps. Our algorithm is fully parallelizable where distance transform for each channel can be computed in parallel. We achieved 16.3 fps upsampling rate on a quad-core computer.

Table 2. **Computation analysis.** The table shows the processing speed (frames per second) achieved by the exact and approximation algorithms. See text for further details.

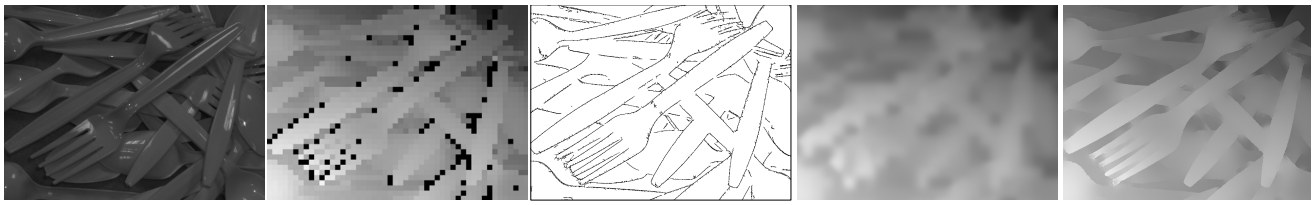
	exact	approx. using multi-pass GDT	approx. using two-pass GDT	multi-core
fps	0.03	3.0	6.0	16.3

4.4. Depth Upsampling Using Binary Edge Maps

In the last experiment, we show that our method is well suited for upsampling depth images using binary images and present an application for sensor fusion. Specifically, we upsample a depth image from a low resolution depth sensor with a high resolution depth boundary map from a multi-flash camera (MFC)[4] as shown in Figure 6. Instead of using both spatial distance and color difference for pixel affinity as in optical images, only spatial distance is used. In addition, the spatial distance is set to infinite for neighboring pixels across depth boundaries. This forces only those depth pixels in the same depth continuous region are used to compute the depths of pixels in the region, achieving boundary-confined upsampling. Note that this boundary-confined property is highly nontrivial for other filtering-based algorithms such as JBU as shown in Figure 6d.

Several advantages exist for upsampling using a depth

[4]MFC is a camera design that exploits illumination pattern change due to LED flashes from different directions to compute a depth boundary map. Note that the resolution of the boundary map is equal to the resolution of the camera and is typically high [15].



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

Figure 6. **Depth upsampling using binary edge maps given by multi-flash camera.** (a) Scene. (b) Input low resolution depth image. (c) Input high resolution depth boundary map from multi-flash camera. (d) 16x upsampled depth image using joint bilateral upsampling (e) 16x upsampled depth image using the proposed algorithm.

boundary map from MFC instead of using a conventional optical image. First, the boundary map directly defines where depth discontinuity occurs. In contrast, depth discontinuity can only be approximately found through color/intensity difference in an optical image, a process sensitive to texture surface. Second, the MFC depth boundary extraction is robust to dust, dirt, grease, and dim illumination, common in industrial environments [13]. Hence, this fusion scheme is better suited for industrial applications.

5. Conclusion

We presented a new joint filtering algorithm using geodesic distances for upsampling depth images using high resolution color images. We developed an efficient approximation to the upsampling filter and achieved real-time performance. We conducted extensive experimental comparison with existing methods and showed that the proposed algorithm advances the state of the art.

Acknowledgements: The authors would like to thank Jay Thornton, Shumpei Kameyama, and Makito Seki for their helpful comments and suggestions.

References

[1] A. Adams, J. Baek, and M. A. Davis. Fast high-dimensional filtering using the permutohedral lattice. In *EUROGRAPHICS*, 2010.

[2] X. Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *CVPR*, 2007.

[3] A. Buades and B. Coll. A non-local algorithm for image denoising. In *CVPR*, 2005.

[4] D. Chan, H. Buisman, C. Theobalt, and S. Thrun. A noise-aware filter for real-time depth upsampling. In *ECCV Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.

[5] A. Criminisi, T. Sharp, C. Rother, and P. Pérez. Geodesic image and video editing. *ACM Transaction on Graphics*, 29(5):134:1–134:15, 2010.

[6] J. Diebel and S. Thrun. An application of markov random fields to range sensing. In *NIPS*, 2005.

[7] J. Dolson, J. Baek, C. Plagemann, and S. Thrun. Upsampling range data in dynamic environments. In *CVPR*, 2010.

[8] G. Facciolo and V. Caselles. Geodesic neighborhoods for piecewise affine interpolation of sparse data. In *ICIP*, 2009.

[9] K. He, J. Sun, and X. Tang. Guided image filtering. In *ECCV*, 2010.

[10] H. Hirschmüller and D. Scharstein. Evaluation of cost functions for stereo matching. In *CVPR*, 2007.

[11] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. *ACM Transaction on Graphics*, 26(3), 2007.

[12] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transaction on Graphics*, 23(3):689–694, 2004.

[13] M.-Y. Liu, O. Tuzel, A. Veeraraghavan, Y. Taguchi, T. K. Marks, and R. Chellappa. Fast object localization and pose estimation in heavy clutter for robotic bin picking. *IJRR*, 31(8):951–973, 2012.

[14] J. Park, H. Kim, Y.-W. Tai, M. S. Brown, and I. Kweon. High quality depth map upsampling for 3D-TOF cameras. In *ICCV*, 2011.

[15] R. Raskar, K.-H. Tan, R. Feris, J. Yu, and M. Turk. Non-photorealistic camera: Depth edge detection and stylized rendering using multi-flash imaging. *ACM Transaction on Graphics*, 23(3):679–688, 2004.

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

[17] M. Thorup and U. Zwick. Approximate distance oracles. In *STOC*, 2001.

[18] P. J. Toivanen. New geodesic distance transforms for gray-scale images. *Pattern Recognition Letters*, 17(5):437–450, 1996.

[19] J. Yang, X. Ye, K. Li, and C. Hou. Depth recovery using an adaptive color-guided auto-regressive model. In *ECCV*, 2012.

[20] Q. Yang. A non-local cost aggregation method for stereo matching. In *CVPR*, 2012.

[21] Q. Yang, K.-H. Tan, and N. Ahuja. Real-time O(1) bilateral filtering. In *CVPR*, 2009.

[22] Q. Yang, R. Yang, J. Davis, and D. Nister. Spatial-depth super resolution for range images. In *CVPR*, 2007.

[23] L. Yatziv, A. Bartesaghi, and G. Sapiro. O(N) implementation of the fast marching algorithm. *Journal of Computational Physics*, 212(2):393–399, 2006.

[24] L. Yatziv and G. Sapiro. Fast image and video colorization using chrominance blending. *TIP*, 15(5):1120–1129, 2006.