# Depth Super Resolution by Rigid Body Self-Similarity in 3D

Michael Hornáček[1,*], Christoph Rhemann[2], Margrit Gelautz[1], and Carsten Rother[2]

[1]Vienna University of Technology
[2]Microsoft Research Cambridge

## Abstract

*We tackle the problem of jointly increasing the spatial resolution and apparent measurement accuracy of an input low-resolution, noisy, and perhaps heavily quantized depth map. In stark contrast to earlier work, we make no use of ancillary data like a color image at the target resolution, multiple aligned depth maps, or a database of high-resolution depth exemplars. Instead, we proceed by identifying and merging patch correspondences within the input depth map itself, exploiting patchwise scene self-similarity across depth such as repetition of geometric primitives or object symmetry. While the notion of 'single-image' super resolution has successfully been applied in the context of color and intensity images, we are to our knowledge the first to present a tailored analogue for depth images. Rather than reason in terms of patches of 2D pixels as others have before us, our key contribution is to proceed by reasoning in terms of patches of 3D points, with matched patch pairs related by a respective 6 DoF rigid body motion in 3D. In support of obtaining a dense correspondence field in reasonable time, we introduce a new 3D variant of PatchMatch. A third contribution is a simple, yet effective patch upscaling and merging technique, which predicts sharp object boundaries at the target resolution. We show that our results are highly competitive with those of alternative techniques leveraging even a color image at the target resolution or a database of high-resolution depth exemplars.*

## 1. Introduction

With the advent of inexpensive 3D cameras like the Microsoft Kinect, depth measurements are becoming increasingly available for low-cost applications. Acquisitions made by such consumer 3D cameras, however, remain afflicted by less than ideal attributes. Random errors are a common problem. Low spatial resolution is an issue particularly with time of flight (ToF) cameras, e.g., $200 \times 200$ for the PMD CamCube 2.0 or $176 \times 144$ for the SwissRanger SR3000. In depth maps recovered using stereo techniques, depth resolution decreases as a function of increasing depth from the camera. Common avenues to jointly increasing the spatial resolution and apparent measurement accuracy of a depth map—a problem referred to as depth super resolution (SR)—involve leveraging ancillary data such as a color or intensity image at the target resolution, multiple aligned depth maps, or a database of high-resolution depth exemplars (patches). Such ancillary data, however, is often unavailable or difficult to obtain.

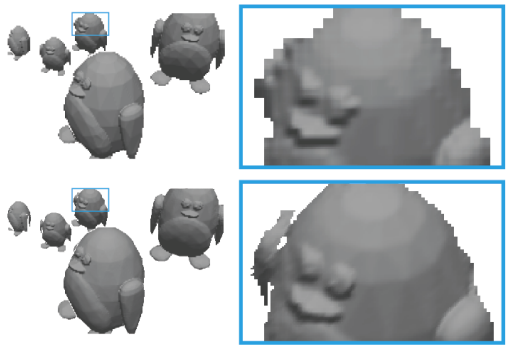


Figure 1. Shaded mesh of nearest neighbor upscaling (top) of a noiseless synthetic input depth map and the output of our algorithm (bottom), both by a factor of 3. In our approach, fine details such as the penguin's eyes, beak, and the subtle polygons across its body are mapped from corresponding patches at lesser depth, and boundaries appear more natural.

In this work, we consider the question of how far one can push depth SR using no ancillary data, proceeding instead by identifying and merging patch correspondences from within the input depth map itself. Our observation is that—even in the absense of object repetition of the sort exemplified in Figure 1—real-world scenes tend to exhibit patchwise 'self-similarity' such as repetition of geometric primitives (e.g., planar surfaces, edges) or object symmetry (consider a face, a vase). Man-made scenes or objects

*Michael Hornáček is funded by Microsoft Research through its European Ph.D. scholarship programme.

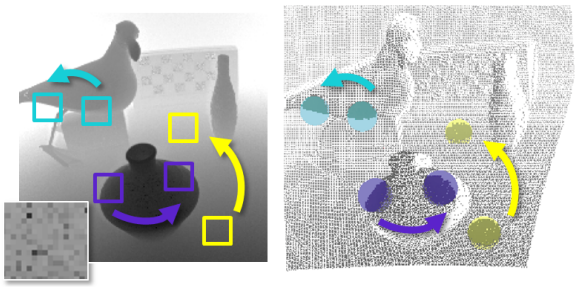


Figure 2. In the depth map (left), the three pairs of 2D patches depicted are dissimilar with respect to depth values. In the corresponding point cloud (right), the analogous 3D patch pairs are similar as point sets related by appropriate rigid body motions in 3D. The inlet shows part of the vase, contrast-stretched for greater clarity; pixel noise is clearly visible.

are often 'self-similar' by design; consider, for instance, the keys of a keyboard. It is primarily this observation that we exploit in this paper, coupled with the fact that under perspective projection, an object patch at lesser depth with respect to the camera is acquired with a higher spatial resolution than a corresponding patch situated at greater depth. The key contribution of our work is to proceed not by reasoning in terms of patches of 2D pixels, but rather in terms of patches of 3D points. It is reasoning in this manner that allows us to exploit scene self-similarity across depth. In addition, we introduce a new 3D variant of PatchMatch to obtain a dense correspondence field in reasonable time and a simple, yet effective patch upscaling and merging technique to generate the output SR depth map.

The notion of 'single-image' SR has already successfully been applied in the context of color and intensity images in the work of Glasner *et al*. [8]. Their guiding observation is that within the same image there is often a large across-scale redundancy at the 2D pixel patch level; for instance, an image of a leafy forest is likely to contain a large number of small patches with various configurations of greens and browns that happen to recur across scales of the image. Their strategy is to search for corresponding $5 \times 5$ pixel patches across a discrete cascade of downscaled copies of the input image and to exploit sub-pixel shifts between correspondences. An SR framework reasoning in terms of small $n \times n$ pixel patches, however, faces serious problems in the context of depth SR. Figure 2 illustrates three fundamental problems of matching 3D points using $n \times n$ pixel patches: patch pairs (i) are situated at different depths or (ii) are subject to projective distortions owing to perspective projection, or (iii) they straddle object boundaries. The problem of projective distortions calls for a small patch size, which renders matching particularly sensitive to noise. We overcome these problems by reasoning in terms of 3D point patches, which we define as the respective inliers—from among the 3D points of the input depth map—within a fixed radius $r$ of a center point and which we match with respect to 3D point similarity over 6 DoF rigid body motions in 3D.

### 1.1. Related Work

A number of surveys of image SR techniques are available elsewhere, e.g., van Ouwerkerk [19] or Tian and Ma [18]. Glasner *et al*. [8], Yang *et al*. [20], and Freeman and Liu [7] are image SR techniques against which we compare our algorithm in Section 3, by treating input depth maps as intensity images. Freeman and Liu *et al*. and Yang *et al*. both rely on an external patch database.

Previous work on depth SR can broadly be categorized into methods that (i) use a guiding color or intensity image at the target resolution, (ii) merge information contained in multiple aligned depth maps, or (iii) call on an external database of high-resolution depth exemplars. We devote the remainder of this section to a discussion of representative or seminal techniques from the depth SR literature.

**Image at Target Resolution.** The most common depth SR strategy involves using an ancillary color or intensity image at the target resolution to guide the reconstruction of the SR depth map. The underlying assumption is that changes in depth are colocated with edges in the guiding image. Yang *et al*. [21] apply joint bilateral upscaling on a cost volume constructed from the low resolution input depth map, followed by Kopf *et al*. [11] in a more general framework. Diebel and Thrun [5] propose an MRF-based approach with a pairwise smoothness term whose contribution is weighted according to the edges in the high-resolution color image. Park *et al*. [13] take this idea further and use a non-local, highly-connected smoothness term that better preserves thin structures in the SR output.

**Multiple Depth Maps.** The Lidarboost approach of Schuon *et al*. [17] combines several depth maps acquired from slightly different viewpoints. The Kinectfusion approach of Izadi *et al*. [10] produces outstanding results by fusing a sequence of depth maps generated by a tracked Kinect camera into a single 3D representation in real-time.

**Database of Depth Exemplars.** Most closely akin to ours is the work of Mac Aodha *et al*. [12]. They propose to assemble the SR depth map from a collection of depth patches. Our approach likewise carries out depth SR by example, but with significant differences. One major difference is that we use patches only from within the input depth map itself, whereas Mac Aodha *et al*. use an external database of 5.2 million high-resolution synthetic, noise-free patches. Another difference is that they carry out their matching in image space over $3 \times 3$ pixel patches, while ours can have arbitrary size depending on the scale, density, and relative depth of point features one aims to capture.

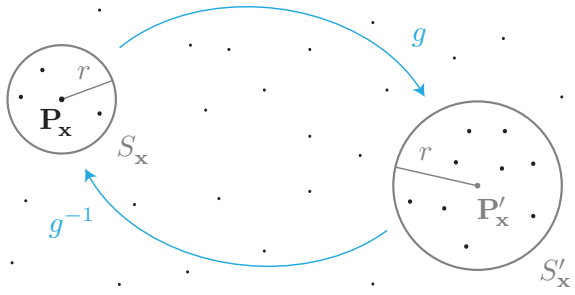



Figure 3. The rigid body motion $g$ relating the 3D point patches $S_\mathbf{x}, S'_\mathbf{x} \subset \mathbb{R}^3$. The point $\mathbf{P}_\mathbf{x} \in \mathbb{R}^3$ is the pre-image of the pixel $\mathbf{x}$ of the input depth map. Note that the center point $\mathbf{P}'_\mathbf{x} = g(\mathbf{P}_\mathbf{x})$ of the closer patch is by design not required to be one of the 3D points of the input depth map, hence $\mathbf{P}'_\mathbf{x} \notin S'_\mathbf{x}$ in general.

Accordingly, their approach is subject to the problems discussed in Section 1 that our reasoning in terms of 3D point patches overcomes. Note that enlarging the patches in their database would lead to an explosion of its size.

## 2. Algorithm

Owing to the perspective projection that underlies image formation, object patches situated at a lesser depth with respect to the camera are imaged with a higher spatial resolution (i.e., a greater point density) than corresponding object patches at greater depth. Our depth SR algorithm consists of two steps: (i) find, for each patch in the input depth map, a corresponding patch at lesser or equal depth with respect to the camera, and (ii) use the dense correspondence field to generate the SR output. We begin, in Section 2.1, by presenting our notion of '3D point patch' and the matching cost we propose to minimize. Next, we detail the first step of our algorithm in Section 2.2, and the second in Section 2.3.

### 2.1. 3D Point Patches

Let $g = (\mathtt{R}, \mathbf{t}) \in SE(3)$ denote a 6 DoF rigid body motion in 3D, where $\mathtt{R} \in SO(3)$ and $\mathbf{t} \in \mathbb{R}^3$. Let $\mathbf{x} = (x, y)^\top$ be a pixel of the input depth map. The goal of the dense correspondence search algorithm in Section 2.2 is to find an optimal rigid body motion $g$ for each pixel $\mathbf{x}$, mapping the patch corresponding to $\mathbf{x}$ to a valid matching patch at lesser or equal depth with respect to the camera. We shall understand the patch corresponding to $\mathbf{x}$—the *further*[1] patch, for brevity—to be the set $S_\mathbf{x} \subset \mathbb{R}^3$ of 3D points within a radius $r$ of the pre-image $\mathbf{P}_\mathbf{x} = Z_\mathbf{x} \cdot \mathtt{K}^{-1}(\mathbf{x}^\top, 1)^\top \in \mathbb{R}^3$ of $\mathbf{x}$, where $Z_\mathbf{x}$ is the depth encoded at $\mathbf{x}$ in the input depth map and $\mathtt{K}$ is the $3 \times 3$ camera calibration matrix (cf. Hartley and Zisserman [9]). We carry out radius queries using a $kd$-tree. The 3D points of the corresponding *closer* patch $S'_\mathbf{x}$ are those within the same radius $r$ of the point $\mathbf{P}'_\mathbf{x} = g(\mathbf{P}_\mathbf{x})$. An illustration of these notions is provided in Figure 3.

[1]We acknowledge that this is something of an abuse of terminology, since two points can be situated at equal depth with respect to the camera but be at different distances from it. Notwithstanding, it is in this sense that we shall mean 'closer' and 'further' in this paper.

**Matching Cost.** A common strategy for evaluating the similarity of two point sets is to compute the sum of squared differences (SSD) over each point in one point set with respect to its nearest neighbor (NN) point in the other (cf. Rusinkiewicz and Levoy [14]). We proceed in a similar manner, but normalize the result and allow for computing SSD in both directions in order to potentially obtain a stronger similarity measure, noting that we might be comparing point sets with significantly different point densities owing to relative differences in patch depth. Let $\mathrm{NN}_S(\mathbf{P})$ denote the function that returns the nearest neighbor to the point $\mathbf{P}$ in the set $S$. The function $c^b(\mathbf{x}; g)$ evaluates normalized SSD over the points of the further patch $S_\mathbf{x}$ subject to each point's respective nearest neighbor among the 'backward'-transformed points $g^{-1}(S'_\mathbf{x})$ of the closer patch:

$$c^b(\mathbf{x}; g) = \sum_{\mathbf{P} \in S_\mathbf{x}} \left\| \mathbf{P} - \mathrm{NN}_{g^{-1}(S'_\mathbf{x})}(\mathbf{P}) \right\|_2^2 / |S_\mathbf{x}|. \quad (1)$$

Analogously, the function $c^f(\mathbf{x}; g)$ evaluates normalized SSD over the points of the closer patch $S'_\mathbf{x}$ subject to their respective nearest neighbors among the 'forward'-transformed points $g(S_\mathbf{x})$ of the further patch:

$$c^f(\mathbf{x}; g) = \sum_{\mathbf{P}' \in S'_\mathbf{x}} \left\| \mathbf{P}' - \mathrm{NN}_{g(S_\mathbf{x})}(\mathbf{P}') \right\|_2^2 / |S'_\mathbf{x}|. \quad (2)$$

For $g$ to be deemed *valid* at $\mathbf{x}$, we require that the depth of the sphere center point of the matched patch be less than or equal to that of the pre-image of $\mathbf{x}$. Moreover, we require that their relative distance be at least $r$ in order to avoid minimizing cost trivially by matching to oneself, and that $|S'_\mathbf{x}| \geq |S_\mathbf{x}| \geq 3$ to benefit from greater point density or from sub-pixel point shifts at equal density, and for reasons discussed below. Given a pixel $\mathbf{x}$ and a rigid body motion $g$, we compute the matching cost $c(\mathbf{x}; g)$ according to

$$c(\mathbf{x}; g) = \begin{cases} \alpha \cdot c^b(\mathbf{x}; g) + \alpha' \cdot c^f(\mathbf{x}; g) & \text{if valid} \\ \infty & \text{otherwise} \end{cases}, \quad (3)$$

where $\alpha \in [0, 1]$ and $\alpha' = 1 - \alpha$.

### 2.2. Dense Correspondence Search

We introduce a new 3D variant of the PatchMatch algorithm (cf. Barnes *et al.* [1]) in the aim of assigning to each pixel $\mathbf{x}$ of the input depth map a 6 DoF rigid body motion in 3D, mapping $S_\mathbf{x}$ to a valid matching patch $S'_\mathbf{x}$ at equal or lesser depth with respect to the camera. PatchMatch was first introduced as a method for obtaining dense approximate nearest neighbor fields between pairs of $n \times n$ pixel patches in 2D, assigning to each pixel $\mathbf{x}$ in an image $A$

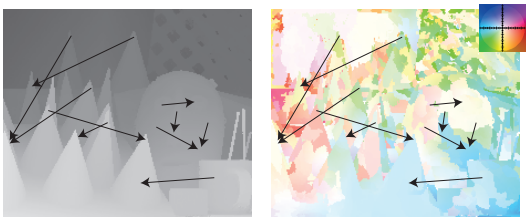


Figure 4. A filled variant of the disparity map of the Middlebury Cones data set (left) as input and a visualization of projected 3D displacements of the output of our dense correspondence search using conventional optical flow coloring, both overlayed sparsely with arrows for greater clarity (right). Note that cone tips map to one another and that the flow field is spatially coherent.

a displacement vector mapping the patch centered at $\mathbf{x}$ to a matching patch in an image $B$ with the objective of reconstructing one image in terms of patches from the other. Although PatchMatch has since been generalized and applied to a variety of other problems (cf. Barnes *et al*. [2] or Besse *et al*. [3]), a common thread between variants of PatchMatch—in which ours is no exception—is a random (or semi-random) initialization step followed by $i$ iterations of propagation and refinement. We explain each step in greater detail in the remainder of this section. An example of a projected displacement field obtained using our 3D variant of PatchMatch is shown in Figure 4.

**Initialization.** In contrast to PatchMatch variants that carry out initialization using altogether random states, we adopt a semi-random initialization strategy. In our experiments, we found this led to faster convergence when dealing with our high-dimensional state space. Specifically, for each pixel $\mathbf{x}$ we randomly select another pixel $\mathbf{x}'$ of the input depth map such that the depth of $\mathbf{P}_{\mathbf{x}'}$ is less than or equal to that of $\mathbf{P}_{\mathbf{x}}$, giving us a translation vector (3 DoF). We then compute the rotation minimizing arc length between the patch normal vector at $\mathbf{P}_{\mathbf{x}}$ and that at $\mathbf{P}_{\mathbf{x}'}$ (2 DoF), and choose a random angular perturbation around the normal of $\mathbf{P}_{\mathbf{x}}$ (1 DoF). We pack these elements into a rigid body motion. A normal vector for each $\mathbf{P}_{\mathbf{x}}$ is precomputed via RANSAC plane fitting over the 3D points in $S_{\mathbf{x}}$ (and is the reason why we require that $|S_{\mathbf{x}}| \geq 3$ in Section 2.1), which is made to point towards the camera.

**Propagation.** In keeping with classical PatchMatch (cf. Barnes *et al*. [1]), we traverse the pixels $\mathbf{x}$ of our input depth map in scanline order—upper left to lower right for even iterations, lower right to upper left for odd—and adopt the rigid body motion assigned to a neighboring pixel if doing so yields an equal or lower cost. Note that as a consequence, we propagate over pixels for which $|S_{\mathbf{x}}| < 3$, which we treat as so-called flying pixels, since such pixels are always assigned infinite cost by $c(\mathbf{x}; g)$ in (3).

**Refinement.** Immediately following propagation at a given pixel $\mathbf{x}$, we independently carry out $k$ iterations of additional initialization and of perturbation of the translational and rotational components of $g_{\mathbf{x}}$, adopting the initialization or perturbation if doing so yields an equal or lower cost. Translational perturbation (3 DoF) consists of checking whether hopping from $\mathbf{P}'_{\mathbf{x}}$ to one of its $k$-NN points $\mathbf{P}_{\mathbf{x}'}$—which we obtain by again making use of a $kd$-tree—yields an equal or lower cost. Rotational perturbation, which we carry out in a range that decreases with every iteration $k$, consists of random rotation around the normal at $\mathbf{P}_{\mathbf{x}}$ (1 DoF) and of random perturbation of the remaining two degrees of freedom of the rotation. We carry out and evaluate all three types of perturbations independently.

## 2.3. Patch Upscaling and Merging

Having assigned a motion $g_{\mathbf{x}} \in SE(3)$ to each pixel $\mathbf{x}$ of the input depth map, we generate an SR depth map by merging interpolated depth values of the 'backward'-transformed points $g_{\mathbf{x}}^{-1}(S'_{\mathbf{x}})$ of each valid matched patch. We begin, for each $\mathbf{x}$, by (i) determining—with the help of contour polygonalization—the spatial extent of $S_{\mathbf{x}}$ at the target resolution, giving an 'overlay mask' over which we then (ii) generate an 'overlay patch' by interpolating depth values from the points $g_{\mathbf{x}}^{-1}(S'_{\mathbf{x}})$. Next, we (iii) populate the SR depth map by merging the interpolated depth values of overlapping overlay patches, with the influence of each valid overlay patch weighted as a function of patch similarity. Finally, we (iv) clean the SR depth map in a postprocessing step, removing small holes that might have arisen at object boundaries as a consequence of polygonalization.

**Overlay Masks.** The 2D pixels $\mathbf{x}$ of the input depth map to which the 3D points of $S_{\mathbf{x}}$ project define the spatial extent of $S_{\mathbf{x}}$ at the input resolution (cf. Figure 5). It is only these pixels, at the input resolution, that the 'backward'-transformed points $g_{\mathbf{x}}^{-1}(S'_{\mathbf{x}})$ of the matched patch are allowed to influence, since it is over these pixels that we compute the matching cost. Upscaling the mask by the SR factor using NN interpolation gives a mask at the target resolution, but introduces disturbing jagged edges. Accordingly, we carry out a polygon approximation (cf. Douglas and Peucker [6]) of this NN upscaled mask, constrained such that approximated contours be at a distance of at most the SR factor—corresponding to a single pixel at the input resolution—from the NN upscaled contours. We ignore recovered polygonalized contours whose area is less than or equal to the square of the SR factor, thereby removing flying pixels. This polygonalized mask—to which we refer as the overlay mask of $\mathbf{x}$—consists of all SR pixels $\hat{\mathbf{x}}$ that fall into one of the remaining polygonalized contours but fall into no contour that is nested inside another, in order to handle holes like in the lamp in Figure 5.

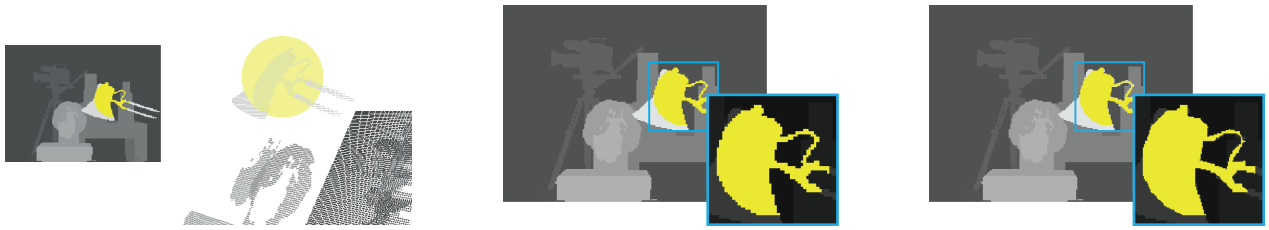


Figure 5. Overlay Masks. Left: A patch of points $S_{\mathbf{x}}$ in the input point cloud and its corresponding pixel mask in the raster of the input depth map (depicted in yellow). Center: NN upscaling of the depth map and mask by a factor of 2. Right: Corresponding polygon approximation of the NN upscaled mask, which we term the 'overlay mask' corresponding to $\mathbf{x}$. In the merging step, it is only the SR pixels $\hat{\mathbf{x}}$ of the overlay mask of $\mathbf{x}$ that the 'backward'-transformed points $g_{\mathbf{x}}^{-1}(S'_{\mathbf{x}})$ of the matched patch are allowed to influence.

**Overlay Patches.** We interpolate, for the SR pixels $\hat{\mathbf{x}}$ of the overlay mask corresponding to $\mathbf{x}$, depth values from the 'backward'-transformed points $g_{\mathbf{x}}^{-1}(S'_{\mathbf{x}})$. Since points transformed according to a rigid body motion in 3D are not guaranteed to project to a regular grid in general, we interpolate over the depth values of these transformed points using barycentric coordinates on a Delaunay triangulation of their projections to image space (cf. Sambridge *et al*. [15]).

**Merging.** The SR depth map is computed by working out, for each SR pixel $\hat{\mathbf{x}}$, a weighted average of the corresponding interpolated depth values from the overlapping overlay patches. The weight $\omega_{\mathbf{x}}$ of the interpolated depth values of the overlay patch assigned to $\mathbf{x}$ is given by $\exp(-\gamma \cdot c^b(\mathbf{x}; g_{\mathbf{x}}))$, where $\gamma \in \mathbb{R}^+$ controls the falloff to 0. If $c^b(\mathbf{x}; g_{\mathbf{x}}) > \beta$, $\beta \in \mathbb{R}^+$, we instead use the overlay patch at $\mathbf{x}$ given by the identity motion, ensuring that patches for which no good match was found do not undergo heavy degradation. We check against $c^b(\mathbf{x}; g_{\mathbf{x}})$ from (1) since it gives an indication of how satisfied the input points are with the match without penalizing the addition of new detail from $S'_{\mathbf{x}}$. As in Section 2.2, if $|S_{\mathbf{x}}| < 3$ then we consider $\mathbf{x}$ a flying pixel, and set $\omega_{\mathbf{x}} = 0$.

**Postprocessing.** Since our polygon approximation guarantees only that the outlines of the polygon be within the SR factor of the outlines of the NN upscaled mask, it is possible that no overlay mask cover a given SR pixel. Such holes can be filled using morphological dilation carried out iteratively, with the dilation affecting only pixels identified as holes. Another possible cause for holes is if pixels within an overlay mask could not be interpolated owing to the spatial distribution of the projected points. In that event, we dilate within the overlay mask with highest weight, again only over pixels identified as holes. Note that no postprocessing was performed in the output in Figure 1.

## 3. Evaluation

We evaluate our method using depth data from stereo, ToF, laser scans and structured light. We carry out a quantitative evaluation in Section 3.1, and provide a qualitative evaluation in the section thereafter. Unless otherwise stated, we performed no preprocessing. In all our experiments, we carried out 5 iterations of PatchMatch, with $k = 3$, and set $\alpha = 0.5$. Setting appropriate parameters $r$, $\beta$, and $\gamma$ is largely intuitive upon visualization of the input point cloud, and depends on the scale, density, and relative depth of point features one aims to capture. In Section 3.1, all algorithm parameters were kept identical across Middlebury and laser scan tests, respectively. We give additional information on parameters and show additional results on our website.

### 3.1. Quantitative Evaluation

Following the example of Mac Aodha *et al*. [12] we provide a quantitative evaluation of our technique on Cones, Teddy, Tsukuba and Venus of the Middlebury stereo data set (cf. Scharstein and Szeliski [16]). For Middlebury tests, we ran our algorithm on filled ground truth data—the same used in Mac Aodha *et al*.—downscaled by NN interpolation by a factor of 2 and 4 and subsequently super resolved by the same factor, respectively, which we compare to ground truth. Table 1 shows root mean squared error (RMSE) scores. Among depth SR methods that leverage a color or intensity image at the target resolution, we compare against Diebel and Thrun [5] and Yang *et al*. [21]; among techniques that make use of an external database we compare against Mac Aodha *et al*., and against Yang *et al*. [20] and Freeman and Liu [7] from the image SR literature. We also compare against the approach of Glasner *et al*. [8]. We compare against NN upscaling to provide a rough baseline, although it introduces jagged edges and does nothing to improve the apparent depth measurement accuracy. Table 1 also gives RMSE scores for three depth maps obtained from laser scans detailed in Mac Aodha *et al*., which we downscaled and subsequently super resolved by a factor of 4. For the laser scans we compare to the original resolution since ground truth data was not available. In Table 2, we provide percent error scores—giving the percentage of pixels for which the absolute difference in disparity exceeds 1—for Middlebury. All RMSE and percent error scores were computed on 8 bit disparity maps . The data sets—with the

exception of results on the algorithm of Glasner *et al*. [8]—and the code used in carrying out the quantitative evaluation are from Mac Aodha *et al*. [12] and were generously provided by the authors.[2]

Although popular in the depth SR literature, RMSE scores over depth or disparity maps are dominated by misassignments at the boundaries of objects separated by large depth differences; given two data sets with equal percent error, a data set where boundaries are gently blurred will have lower RMSE than one with boundaries that are sharp. Even so, our RMSE scores fare highly competitively with those of alternative techniques. In percent error, we are the top performer among example-based methods, and on a few occasions outperform the image-guided techniques.

### 3.2. Qualitative Evaluation

In Figure 6 we show results on a data set of two similar egg cartons situated at different depths, obtained using the stereo algorithm of Bleyer *et al*. [4]. Our result is visually superior to that of our competitors, and is the only one to succeed in removing noise. Note the patch artefacts for Mac Aodha *et al*. in the zoom. In Figure 7, we consider a noisy ToF data set from [12]. We see that although our depth map appears pleasing, it in fact remains gently noisy if shaded as a mesh, owing to the great deal of noise in the input. However, if we apply the same bilateral filtering as Mac Aodha *et al*. [12], our result when shaded—although not as smooth over the vase—preserves edges better (e.g., at the foot) without introducing square patch artefacts. Note that Glasner *et al*. do not succeed in removing visible noise in their depth map, and introduce halo artefacts at the boundaries. Figure 8 provides a comparison over the noiseless, yet quantized Cones data set. Note that although Glasner *et al*. [8] perform well in RMSE, their method produces poor object boundaries.

## 4. Conclusion

Inspired by the work of Glasner *et al*. [8] on single-image super resolution for color and intensity images, we presented a tailored depth super resolution algorithm that makes use of only the information contained in the input depth map. We introduced a new 3D variant of PatchMatch for recovering a dense matching between pairs of closer-further corresponding 3D point patches related by 6 DoF rigid body motions in 3D and presented a technique for upscaling and merging matched patches that predicts sharp object boundaries at the target resolution. In our evaluation, we showed our results to be highly competitive with methods leveraging ancillary data.

[2]The RMSE scores published in Mac Aodha *et al*. [12] were subject to a subtle image resizing issue. Details and updated numbers are available at `http://visual.cs.ucl.ac.uk/pubs/depthSuperRes/supp/index.html`.

## References


[1] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *SIGGRAPH*, 2009.

[2] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein. The generalized PatchMatch correspondence algorithm. In *ECCV*, 2010.

[3] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz. Pmbp: Patchmatch belief propagation for correspondence field estimation. In *BMVC*, 2012.

[4] M. Bleyer, C. Rhemann, and C. Rother. PatchMatch Stereo - Stereo matching with slanted support windows. In *BMVC*, 2011.

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

[6] D. Douglas and T. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica*, 10(2):112–122, 1973.

[7] W. T. Freeman and C. Liu. Markov random fields for super-resolution and texture synthesis. In *Advances in Markov Random Fields for Vision and Image Processing*. MIT Press, Berlin, 2011.

[8] D. Glasner, S. Bagon, and M. Irani. Super-resolution from a single image. In *ICCV*, 2009.

[9] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

[10] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera. In *UIST*, 2011.

[11] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele. Joint bilateral upsampling. *SIGGRAPH*, 2007.

[12] O. Mac Aodha, N. D. Campbell, A. Nair, and G. J. Brostow. Patch based synthesis for single depth image super-resolution. In *ECCV*, 2012.

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

[14] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *3DIM*, 2001.

[15] M. Sambridge, J. Braun, and H. McQueen. Geophysical parametrization and interpolation of irregular data using natural neighbours. *Geophysical Journal International*, 122(3):837–857, 1995.

[16] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *CVPR*, 2003.

[17] S. Schuon, C. Theobalt, J. Davis, and S. Thrun. LidarBoost: Depth superresolution for ToF 3D shape scanning. In *CVPR*, 2009.

[18] J. Tian and K.-K. Ma. A survey on super-resolution imaging. *Signal, Image and Video Processing*, 5(3):329–342, 2011.

[19] J. van Ouwerkerk. Image super-resolution survey. *Image and Vision Computing*, 24(10):1039 – 1052, 2006.

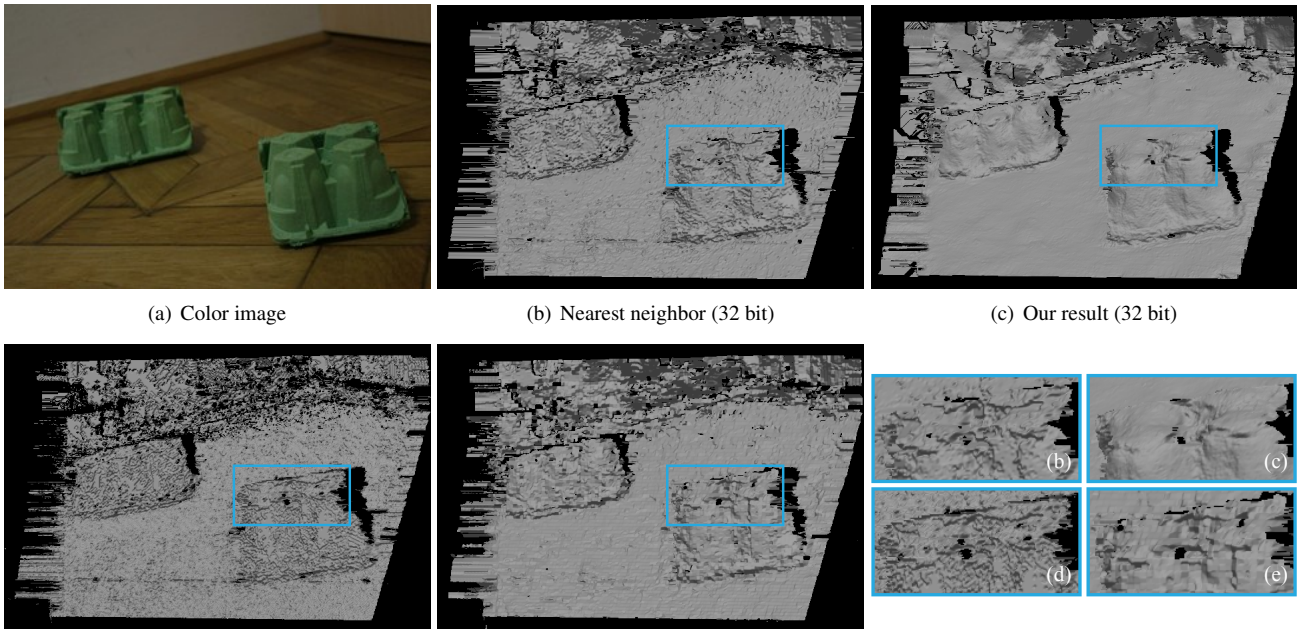
[20] J. Yang, J. Wright, T. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, 2010.

| | 2x | | | | 4x | | | | 4x | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Cones | Teddy | Tsukuba | Venus | Cones | Teddy | Tsukuba | Venus | Scan 21 | Scan 30 | Scan 42 |
| Nearest Neighbor | 1.094 | 0.815 | 0.612 | 0.268 | 1.531 | 1.129 | 0.833 | 0.368 | 0.018 | 0.016 | 0.040 |
| Diebel and Thrun [5] | 0.740 | 0.527 | 0.401 | 0.170 | 1.141 | 0.801 | 0.549 | 0.243 | N/A | N/A | N/A |
| Yang *et al*. [21] | 0.756 | 0.510 | 0.393 | 0.167 | 0.993 | 0.690 | 0.514 | 0.216 | N/A | N/A | N/A |
| Yang *et al*. [20] | 2.027 | 1.420 | 0.705 | 0.992 | 2.214 | 1.572 | 0.840 | 1.012 | 0.030 | 0.035 | 0.054 |
| Freeman and Liu [7] | 1.447 | 0.969 | 0.617 | 0.332 | 1.536 | 1.110 | 0.869 | 0.367 | 0.019 | 0.017 | 0.075 |
| Glasner *et al*. [8] | **0.867** | **0.596** | **0.482** | **0.209** | 1.483 | 1.065 | 0.832 | 0.394 | 1.851 | 1.865 | 1.764 |
| Mac Aodha *et al*. [12] | 1.127 | 0.825 | 0.601 | 0.276 | 1.504 | **1.026** | 0.833 | **0.337** | **0.017** | **0.017** | 0.045 |
| Our Method | 0.994 | 0.791 | 0.580 | 0.257 | **1.399** | 1.196 | **0.727** | 0.450 | 0.021 | 0.018 | **0.030** |

Table 1. Root mean squared error (RMSE) scores. Yang *et al*. [20] and Freeman and Liu [7] are image SR methods and Mac Aodha *et al*. [12] a depth SR method, all of which require an external database. Diebel and Thrun [5] and Yang *et al*. [21] are depth SR methods that use an image at the target resolution. Glasner *et al*. [8] is an image SR technique that uses patches from within the input image. For most data sets, our method is competitive with the top performer. Laser scan tests on the image-guided techniques were not possible for want of images at the target resolution. Best score is indicated in bold for the example-based methods, which we consider our main competitors.

| | 2x | | | | 4x | | | |
|---|---|---|---|---|---|---|---|---|
| | Cones | Teddy | Tsukuba | Venus | Cones | Teddy | Tsukuba | Venus |
| Nearest Neighbor | 1.713 | 1.548 | 1.240 | 0.328 | 3.121 | 3.358 | 2.197 | 0.609 |
| Diebel and Thrun [5] | 3.800 | 2.786 | 2.745 | 0.574 | 7.452 | 6.865 | 5.118 | 1.236 |
| Yang *et al*. [21] | 2.346 | 1.918 | 1.161 | 0.250 | 4.582 | 4.079 | 2.565 | 0.421 |
| Yang *et al*. [20] | 61.617 | 54.194 | 5.566 | 46.985 | 63.742 | 55.080 | 7.649 | 47.053 |
| Freeman and Liu [7] | 6.266 | 4.660 | 3.240 | 0.790 | 15.077 | 12.122 | 10.030 | 3.348 |
| Glasner *et al*. [8] | 4.697 | 3.137 | 3.234 | 0.940 | 8.790 | 6.806 | 6.454 | 1.770 |
| Mac Aodha *et al*. [12] | 2.935 | 2.311 | 2.235 | 0.536 | 6.541 | 5.309 | 4.780 | **0.856** |
| Our Method | **2.018** | **1.862** | **1.644** | **0.377** | **3.271** | **4.234** | **2.932** | 3.245 |

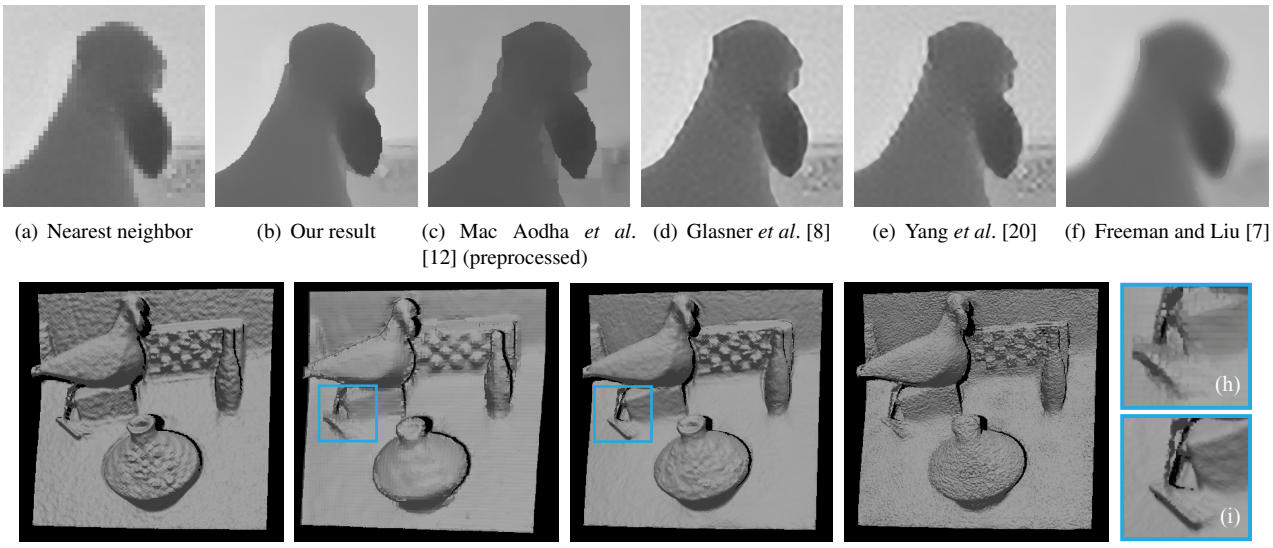
Table 2. Percent error scores. Our method is the top performer among example-based methods and on a few occasions outperforms Diebel and Thrun [5] and Yang *et al*. [21]. Results provided for Yang *et al*. [20] suffer from incorrect absolute intensities.




(a) Color image (b) Nearest neighbor (32 bit) (c) Our result (32 bit)

(d) Glasner *et al*. [8] (8 bit) (e) Mac Aodha *et al*. [12] (preprocessed, 32 bit) (f) Zooms

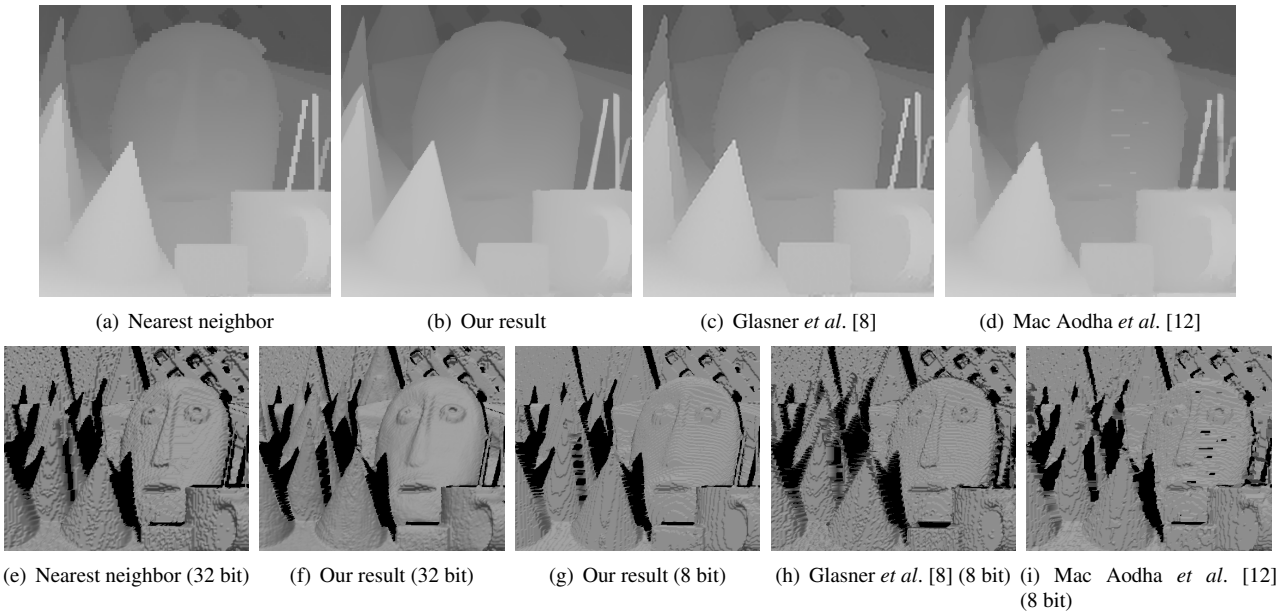
Figure 6. 2x nearest neighbor upscaling (b) and SR (c-e) on a stereo data set of two similar egg cartons obtained using the method of Bleyer *et al*. [4]. Note that (e) was preprocessed using a bilateral filter (window size 5, spatial deviation 0.5, range deviation 0.001).




(a) Nearest neighbor (b) Our result (c) Mac Aodha *et al*. [12] (preprocessed) (d) Glasner *et al*. [8] (e) Yang *et al*. [20] (f) Freeman and Liu [7]

(g) Nearest neighbor (preprocessed, 32 bit) (h) Mac Aodha *et al*. [12] (preprocessed, 32 bit) (i) Our result (preprocessed, 32 bit) (j) Our result (no preprocessing, 32 bit) (k) Zooms

Figure 7. Above, we provide zooms on a region of interest of the noisy PMD CamCube 2.0 ToF data set shown in Figure 2 for 4x nearest neighbor upscaling in (a) and 4x SR otherwise. A depth map zoom for Mac Aodha *et al*. was available only with bilateral preprocessing (window size 5, spatial deviation 3, range deviation 0.1). Below, we show shaded meshes for the preprocessed result of Mac Aodha *et al*. [12] and for our method with and without the same preprocessing ((h) is not aligned with the other meshes because we obtained the rendering from the authors). Note that although we in (i) perform worse than (h) on the vase, we preserve fine detail better and do not introduce square patch artefacts.



(a) Nearest neighbor (b) Our result (c) Glasner *et al*. [8] (d) Mac Aodha *et al*. [12]

(e) Nearest neighbor (32 bit) (f) Our result (32 bit) (g) Our result (8 bit) (h) Glasner *et al*. [8] (8 bit) (i) Mac Aodha *et al*. [12] (8 bit)

Figure 8. Above, zooms on a region of interest of the noiseless, though quantized Middlebury Cones data set. 2x SR was carried out (in our case, using the parameters from the quantitative evaluation) on the 2x nearest neighbor downscaling of the original, depicted in (a). Our method produces the sharpest object boundaries. Below, the corresponding shaded meshes. We show our 8 bit quantized mesh in (g) for comparison. Our method performs the best smoothing even after quantization (particularly over the cones), although it lightly smooths away the nose for the parameters used, which were kept the same for all Middlebury tests. We provide additional results on our website.

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