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

Michael Hornáček\textsuperscript{1,\*}, Christoph Rhemann\textsuperscript{2}, Margrit Gelautz\textsuperscript{1}, and Carsten Rother\textsuperscript{2}

\textsuperscript{1}Vienna University of Technology
\textsuperscript{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.

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 absence 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.
The most common depth approach of 3 pixel patches, while of a center point and which we match with respect et al. 

Most closely akin to ours et al. 

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 × 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 × 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 × 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 × 3 pixel patches, while ours can have arbitrary size depending on the scale, density, and relative depth of point features one aims to capture.
are those within the same radius \( r \) of the point \( \mathbf{P}'_x = g(\mathbf{P}_x) \). An illustration of these notions is provided in Figure 3.

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 \( g = (\mathbf{R}, \mathbf{t}) \in SE(3) \) denote a 6 DoF rigid body motion in 3D, where \( \mathbf{R} \in SO(3) \) 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 \( S'_x \subset \mathbb{R}^3 \) of 3D points within a radius \( r \) of the pre-image \( P'_x = Z_x \cdot \mathbf{K}^{-1}(x, y, 1)^\top \in \mathbb{R}^3 \) of \( \mathbf{x} \), where \( Z_x \) is the depth encoded at \( \mathbf{x} \) in the input depth map and \( \mathbf{K} \) 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 \( S_x \)

\(^1\)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. \(^1\)Let \( \text{NN}_S(\mathbf{P}) \) 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 \( S'_x \) subject to each point’s respective nearest neighbor among the ‘backward’.transformed points \( g^{-1}(S'_x) \) of the closer patch:

\[
c^b(\mathbf{x}; g) = \sum_{\mathbf{P} \in S'_x} \frac{||\mathbf{P} - \text{NN}_{g^{-1}(S'_x)}(\mathbf{P})||^2_2}{|S'_x|}.
\]

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 \( |S'_x| \geq |S_x| \geq 3 \) 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

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

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. [11]) in the aim of assigning to each pixel \( \mathbf{x} \) of the input depth map a 6 DoF rigid body motion in 3D, mapping \( S_x \) to a valid matching patch \( S'_x \) 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 \( A \)

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

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.
Immediately following propagation at a given pixel \( x \), we independently carry out \( k \) iterations of additional initialization and of perturbation of the translational and rotational components of \( g_x \), adopting the initialization or perturbation if doing so yields an equal or lower cost. Translational perturbation (3 DoF) consists of checking whether hopping from \( P'_x \) to one of its \( k \)-NN points \( P'_{x'} \)—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 \( P_x \) (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.

### Refinement

Having assigned a motion \( g_x \in SE(3) \) to each pixel \( x \) of the input depth map, we generate an SR depth map by merging interpolated depth values of the ‘backward’-transformed points \( g_x^{-1}(S'_x) \) of each valid matched patch. We begin, for each \( x \), by (i) determining—with the help of contour polygonalization—the spatial extent of \( S_x \) at the target resolution, giving an ‘overlay mask’ over which we then (ii) generate an ‘overlay patch’ by interpolating depth values from the points \( g_x^{-1}(S'_x) \). 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.

### 2.3. Patch Upscaling and Merging

Having assigned a motion \( g_x \in SE(3) \) to each pixel \( x \) of the input depth map, we generate an SR depth map by merging interpolated depth values of the ‘backward’-transformed points \( g_x^{-1}(S'_x) \) of each valid matched patch. We begin, for each \( x \), by (i) determining—with the help of contour polygonalization—the spatial extent of \( S_x \) at the target resolution, giving an ‘overlay mask’ over which we then (ii) generate an ‘overlay patch’ by interpolating depth values from the points \( g_x^{-1}(S'_x) \). 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 \( x \) of the input depth map to which the 3D points of \( S_x \) project define the spatial extent of \( S_x \) at the input resolution (cf. Figure 5). It is only these pixels, at the input resolution, that the ‘backward’-transformed points \( g_x^{-1}(S'_x) \) 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 \( x \)—consists of all SR pixels \( x \) 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.
Overlay Patches. We interpolate, for the SR pixels \( \tilde{x} \) of the overlay mask corresponding to \( x \), depth values from the ‘backward’-transformed points \( g_x^{-1}(S'_x) \). 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 \( \tilde{x} \), a weighted average of the corresponding interpolated depth values from the overlapping overlay patches. The weight \( \omega_x \) of the interpolated depth values of the overlay patch assigned to \( x \) is given by \( \exp(-\gamma \cdot c^h(x; g_x)) \), where \( \gamma \in \mathbb{R}^+ \) controls the falloff to 0. If \( c^h(x; g_x) > \beta \), \( \beta \in \mathbb{R}^+ \), we instead use the overlay patch at \( x \) given by the identity motion, ensuring that patches for which no good match was found do not undergo heavy degradation. We check against \( c^h(x; g_x) \) 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 \( S'_x \). As in Section 2.2, if \( |S_x| < 3 \) then we consider \( x \) a flying pixel, and set \( \omega_x = 0 \).

Postprocessing. Since our polygon approximation guarantees only that the outlines of the polygon be within the SR factor of the outlines of the NN upsampled 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 downsampled 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 pro-
vided by the authors.2

Although popular in the depth SR literature, RMSE
scores over depth or disparity maps are dominated by mis-
assignments 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 bound-
aries. 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 up-
scaling and merging matched patches that predicts sharp ob-
ject boundaries at the target resolution. In our evaluation,
we showed our results to be highly competitive with meth-
ods leveraging ancillary data.

2The 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. Gold-
man. PatchMatch: A randomized correspondence algorithm
for structural image editing. SIGGRAPH, 2009.
stein. The generalized PatchMatch correspondence algo-

Patchmatch belief propagation for correspondence field esti-

Stereo matching with slanted support windows. In BMVC,
2011.
fields to range sensing. In NIPS, 2005.
the number of points required to represent a digitized line or
resolution and texture synthesis. In Advances in Markov
Random Fields for Vision and Image Processing. MIT Press,
Berlin, 2011.
Computer Vision. Cambridge University Press, second
P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison,
and A. Fitzgibbon. Kinectfusion: real-time 3D reconstruc-
tion and interaction using a moving depth camera. In UIST,
2011.
[12] O. Mac Aodha, N. D. Campbell, A. Nair, and G. J. Bros-
tow. Patch based synthesis for single depth image super-
quality depth map upsampling for 3D-TOF cameras. In
ICCV, 2011.
parametrization and interpolation of irregular data us-
ing natural neighbours. Geophysical Journal International,
Depth superresolution for ToF 3D shape scanning. In CVPR,
2009.
resolution via sparse representation. IEEE Transactions on
<table>
<thead>
<tr>
<th>Method</th>
<th>2x Cones</th>
<th>2x Teddy</th>
<th>2x Tsukuba</th>
<th>2x Venus</th>
<th>4x Cones</th>
<th>4x Teddy</th>
<th>4x Tsukuba</th>
<th>4x Venus</th>
<th>Scan 21</th>
<th>Scan 30</th>
<th>Scan 42</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest Neighbor</td>
<td>1.094</td>
<td>0.815</td>
<td>0.612</td>
<td>0.268</td>
<td>1.531</td>
<td>1.129</td>
<td>0.833</td>
<td>0.368</td>
<td>0.018</td>
<td>0.016</td>
<td>0.040</td>
</tr>
<tr>
<td>Diebel and Thrun [5]</td>
<td>0.740</td>
<td>0.527</td>
<td>0.401</td>
<td>0.170</td>
<td>1.141</td>
<td>0.801</td>
<td>0.549</td>
<td>0.243</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Yang et al. [21]</td>
<td>0.756</td>
<td>0.510</td>
<td>0.393</td>
<td>0.167</td>
<td>0.993</td>
<td>0.690</td>
<td>0.514</td>
<td>0.216</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Method</th>
<th>2x Cones</th>
<th>2x Teddy</th>
<th>2x Tsukuba</th>
<th>2x Venus</th>
<th>4x Cones</th>
<th>4x Teddy</th>
<th>4x Tsukuba</th>
<th>4x Venus</th>
<th>Scan 21</th>
<th>Scan 30</th>
<th>Scan 42</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yang et al. [20]</td>
<td>2.027</td>
<td>1.420</td>
<td>0.705</td>
<td>0.992</td>
<td>2.214</td>
<td>1.572</td>
<td>0.840</td>
<td>1.012</td>
<td>0.030</td>
<td>0.035</td>
<td>0.054</td>
</tr>
<tr>
<td>Freeman and Liu [7]</td>
<td>1.447</td>
<td>0.969</td>
<td>0.617</td>
<td>0.332</td>
<td>1.536</td>
<td>1.110</td>
<td>0.869</td>
<td>0.367</td>
<td>0.019</td>
<td>0.017</td>
<td>0.075</td>
</tr>
<tr>
<td>Glasner et al. [8]</td>
<td>0.857</td>
<td>0.596</td>
<td>0.482</td>
<td>0.209</td>
<td>1.483</td>
<td>1.065</td>
<td>0.832</td>
<td>0.394</td>
<td>1.851</td>
<td>1.865</td>
<td>1.764</td>
</tr>
<tr>
<td>Mac Aodha et al. [12]</td>
<td>1.127</td>
<td>0.825</td>
<td>0.601</td>
<td>0.276</td>
<td>1.504</td>
<td>1.026</td>
<td>0.833</td>
<td>0.337</td>
<td>0.017</td>
<td>0.017</td>
<td>0.045</td>
</tr>
<tr>
<td>Our Method</td>
<td>0.994</td>
<td>0.791</td>
<td>0.580</td>
<td>0.257</td>
<td>1.399</td>
<td>1.196</td>
<td>0.727</td>
<td>0.450</td>
<td>0.021</td>
<td>0.018</td>
<td>0.030</td>
</tr>
</tbody>
</table>

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.
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.

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.