Depth Super Resolution by Rigid Body Self-Similarity in 3D (CVPR 2013)

M. Hornáček\textsuperscript{1} \hspace{1cm} C. Rhemann\textsuperscript{2} \hspace{1cm} M. Gelautz\textsuperscript{1} \hspace{1cm} C. Rother\textsuperscript{2}

\textsuperscript{1}Vienna University of Technology
Vienna, Austria

\textsuperscript{2}Microsoft Research Ltd.
Cambridge, United Kingdom

June 20, 2013

Microsoft Research
Objective

Input:
Single low-resolution, noisy, and perhaps heavily quantized depth map

Objective:
Jointly increase spatial resolution and apparent measurement accuracy of input
Motivating Example: 3x Nearest Neighbor Upscaling
Motivating Example: 3x SR Output of Our Algorithm
Related Work: Guiding Image at Target Resolution

Figure: Yang et al. [21] iteratively refine low resolution input using aligned guiding color image at target resolution.
Related Work: Multiple Depth Maps

Figure: Izadi et al. [10] produce outstanding results by fusing a sequence of depth maps generated by a tracked Kinect camera into a single 3D representation.
Challenges: Ancillary Data or Multiple Depth Maps

Guiding image at target resolution or multiple depth maps often unavailable or difficult to obtain.
Figure: Assemble SR output using corresponding $5 \times 5$ pixel patches found across a discrete cascade of downscaled copies of input image.
Related Work: External Patch Database

Figure: Mac Aodha et al. [12] assemble SR output using external database of 5.2 million high-resolution synthetic, noise-free 2D pixel patches.
Challenges: 2D Pixel Patches

Proceeding ‘by example’—by assembling SR output from matched 2D pixel patches—poses its own challenges:

- Different patch depths (depth normalization?)
- Projective distortions (calls for a small patch size)
- Object boundaries (discontinuity handling?)
Challenges: 2D Pixel Patches
Our Contributions

‘Single image’ depth SR—using information only from input depth map—by:

- Reasoning in terms of 3D point patches
- New 3D variant of PatchMatch (cf. Barnes et al. [1])
- Simple, yet effective patch upscaling and merging technique
Our depth SR algorithm reduces to two steps:

1. **Dense correspondence search** via new 3D PatchMatch variant
2. **Patch upscaling and merging** to generate SR output
3D Point Patches

\[ g^{-1} \quad S_x \xrightarrow{g} \quad S'_x \]

\[ P_x \xrightarrow{r} \quad P'_x \]
3D Point Patches

‘Further’ Patch $S_x \subset \mathbb{R}^3$

Set of 3D points of input depth map within a fixed radius $r$ of pre-image $P_x = Z_x \cdot K^{-1}(x^T, 1)^T \in \mathbb{R}^3$ of $x$, where $Z_x$ is depth encoded at $x$ in input depth map and $K$ is $3 \times 3$ camera calibration matrix.

‘Closer’ Patch $S'_x \subset \mathbb{R}^3$

Set of 3D points of input depth map within the same $r$ of point $P'_x = g(P_x) \in \mathbb{R}^3$, where $g = (R, t) \in SE(3)$ is a 6 DoF rigid body motion in 3D such that depth of $P'_x$ be less than or equal to that of $P_x$. 
Patch Similarity: ‘Backward’ Cost $c^b(x; g)$

$$c^b(x; g) = \sum_{P \in S_x} \| P - \text{NN}_{g^{-1}(S'_x)}(P) \|_2^2 / |S_x|$$
Patch Similarity: ‘Backward’ Cost \( c^b(x; g) \)

\[
c^b(x; g) = \sum_{P \in S_x} \left\| P - \text{NN}_{g^{-1}(S'_{x})}(P) \right\|_2^2 / |S_x|
\]
Patch Similarity: ‘Backward’ Cost \( c^b(x; g) \)

\[
c^b(x; g) = \sum_{P \in S_x} \left\| P - \text{NN}_{g^{-1}(S'_{x})}(P) \right\|_2^2 / |S_x|
\]
**Patch Similarity: ‘Backward’ Cost $c^b(x; g)$**

The equation for the ‘backward’ cost is given by:

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

Where $S_x$ and $S'_{x}$ are subsets of points, and $\text{NN}_{g^{-1}(S'_{x})}(P)$ represents the nearest neighbor in the set $S'_{x}$ under the transformation $g^{-1}$.
Patch Similarity: ‘Backward’ Cost $c^b(x; g)$

$$c^b(x; g) = \sum_{P \in S_x} \frac{\|P - \text{NN}_{g^{-1}(S'_x)}(P)\|^2}{|S_x|}$$
Patch Similarity: ‘Backward’ Cost $c^b(x; g)$

\[
c^b(x; g) = \sum_{P \in S_x} \left\| P - \text{NN}_{g^{-1}(S'_x)}(P) \right\|_2^2 / |S_x|
\]
Patch Similarity: ‘Forward’ Cost $c^f(x; g)$

‘Backward’ cost $c^b(x; g)$ computes patch similarity without penalizing addition of new detail. To be more confident that such new detail is reasonable, we also compute analogous ‘forward’ cost $c^f(x; g)$. 
Patch Similarity: ‘Forward’ Cost $c^f(x; g)$

\[
c^f(x; g) = \sum_{P' \in S'_x} \left\| P' - \text{NN}_g(S_x)(P') \right\|_2^2 / |S'_x|
\]
Patch Similarity: ‘Forward’ Cost $c^f(x; g)$

$$c^f(x; g) = \sum_{P' \in S'_x} \left\| P' - \text{NN}_g(S_x)(P') \right\|_2^2 / |S'_x|$$
Patch Similarity: ‘Forward’ Cost $c^f(x; g)$

$$c^f(x; g) = \sum_{P' \in S'_x} \| P' - \text{NN}_{g(S_x)}(P') \|^2_2 / |S'_x|$$
Patch Similarity: ‘Forward’ Cost $c^f(x; g)$

$$
c^f(x; g) = \sum_{P' \in S'_x} \left\| P' - \text{NN}_{g(S_x)}(P') \right\|_2^2 / |S'_x|
$$

Hornáček et al.  Depth SR by Rigid Body Self-Similarity in 3D  26
Patch Similarity: ‘Forward’ Cost $c^f(x; g)$

$$c^f(x; g) = \sum_{P' \in \mathcal{S}_x} \left\| P' - \text{NN}_{\mathcal{S}_x}(P') \right\|_2^2 / |\mathcal{S}_x|$$
Patch Similarity: ‘Forward’ Cost $c^f(x; g)$

$$c^f(x; g) = \sum_{P' \in S'_x} \| P' - \text{NN}_g(S_x)(P') \|_2^2 / |S'_x|$$
Patch Similarity: ‘Forward’ Cost $c^f(x; g)$

\[
c^f(x; g) = \sum_{P' \in S'_x} \| P' - \text{NN}_g(S_x)(P') \|_2^2 / |S'_x|
\]
Patch Similarity: ‘Forward’ Cost $c^f(x; g)$

$$c^f(x; g) = \sum_{P' \in S_x'} \| P' - \text{NN}_{g(S_x)}(P') \|_2^2 / |S_x'|$$
Patch Similarity: ‘Forward’ Cost $c^f(x; g)$

$$c^f(x; g) = \sum_{P' \in S'_x} \left\| P' - \text{NN}_g(S_x)(P') \right\|_2^2 / |S'_x|$$
Patch Similarity: ‘Forward’ Cost $c^f(x; g)$

\[
c^f(x; g) = \sum_{P' \in S'_x} \|P' - \text{NN}_g(S_x)(P')\|_2^2 / |S'_x|
\]
Patch Similarity: ‘Forward’ Cost $c^f(x; g)$

$$c^f(x; g) = \sum_{P' \in S'_x} \left\| P' - \text{NN}_g(S_x)(P') \right\|_2^2 / |S'_x|$$
Patch Similarity: Matching Cost $c(x; g)$

We compute matching cost $c(x; g)$ according to

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

where $\alpha \in [0, 1]$ and $\alpha' = 1 - \alpha$. 
Patch Similarity: Validity of $g$ at $x$

We deem a rigid body motion $g$ valid at $x$ if

- $\|P_x - P'_x\|_2 \geq r$ to prevent trivial minimization
- $|S'_x| \geq |S_x| \geq 3$ to match to at least as many points
Assign to each input pixel $x$ a valid 6 DoF 3D rigid body motion $g_x$ by (semi-)random initialization followed by $i$ iterations propagation and refinement.
3D PatchMatch: Semi-Random Initialization
3D PatchMatch: Semi-Random Initialization
3D PatchMatch: Semi-Random Initialization (1/3)
3D PatchMatch: Semi-Random Initialization (2/3)
3D PatchMatch: Semi-Random Initialization (3/3)
3D PatchMatch: Propagation
3D PatchMatch: Propagation
3D PatchMatch: Propagation
3D PatchMatch: Propagation
3D PatchMatch: Propagation
3D PatchMatch: Propagation
3D PatchMatch: Propagation (Odd Iterations $i$)
3D PatchMatch: Propagation (Odd Iterations $i$)
3D PatchMatch: Propagation (Odd Iterations $i$)
3D PatchMatch: Propagation (Odd Iterations $i$)
3D PatchMatch: Propagation (Odd Iterations $i$)
3D PatchMatch: Propagation (Odd Iterations $i$)
3D PatchMatch: Propagation (Even Iterations $i$)
3D PatchMatch: Propagation (Even Iterations $i$)
3D PatchMatch: Propagation (Even Iterations $i$)
3D PatchMatch: Propagation (Even Iterations $i$)
3D PatchMatch: Propagation (Even Iterations $i$)
3D PatchMatch: Propagation (Even Iterations $i$)
3D PatchMatch: Refinement

We independently carry out $k$ iterations of additional initialization and of perturbation of the translational and rotational components of $g_x$. 
3D PatchMatch: Visualization
Putting It All Together?

Figure: Overlapping matches? Object boundaries?
Patch Upscaling and Merging: Overlay Masks (1/2)
Patch Upscaling and Merging: Overlay Masks (2/2)
Patch Upscaling and Merging: Overlay Patches

Hornáček et al. Depth SR by Rigid Body Self-Similarity in 3D
Patch Upscaling and Merging: Merging

SR output generated by weighted sum over overlapping overlay patches. Patch weight $\omega_x$ computed as function of $c^b(x; g_x)$ in order to promote addition of new detail:

$$\omega_x = \exp \left( -\gamma \cdot c^b(x; g_x) \right).$$

If $c^b(x; g_x) > \beta$, we instead use overlay patch at $x$ corresponding to identity motion.
Reminder: ‘Backward’ Cost $c^b(x; g)$

$$c^b(x; g) = \sum_{P \in S_x} \left\| P - \text{NN}_{g^{-1}(S'_{x})}(P) \right\|_2^2 / |S_x|$$
Qualitative Evaluation: Egg Cartons (Stereo)

Figure : Color image.
Qualitative Evaluation: Egg Cartons (Stereo)

Figure: 2x nearest neighbor (32 bit).
Qualitative Evaluation: Egg Cartons (Stereo)

Figure: 2x SR result of our method (32 bit).
Qualitative Evaluation: Egg Cartons (Stereo)

Figure: 2x SR result of Glasner et al. [8] (8 bit).
Qualitative Evaluation: Egg Cartons (Stereo)

Figure: 2x SR result of Mac Aodha et al. [12] (preprocessed, 32 bit).
Qualitative Evaluation: Egg Cartons (Stereo)

Figure: Zooms.
Qualitative Evaluation: Gull (ToF)

Figure: 4x nearest neighbor.
Qualitative Evaluation: Gull (ToF)

Figure: 4x nearest neighbor (zoom).
Qualitative Evaluation: Gull (ToF)

Figure: 4x SR result of our method (zoom).
Qualitative Evaluation: Gull (ToF)

Figure: 4x SR result of Mac Aodha et al. [12] (preprocessed, zoom).
Qualitative Evaluation: Gull (ToF)

Figure: 4x SR result of Glasner et al. [8] (zoom).
Qualitative Evaluation: Gull (ToF)

Figure: 4x SR result of Yang et al. [20] (zoom).
Qualitative Evaluation: Gull (ToF)

Figure: 4x SR result of Freeman and Liu [7] (zoom).
Qualitative Evaluation: Middlebury Cones (Struct. Light)

Figure: 2x nearest neighbor.
Qualitative Evaluation: Middlebury Cones (Struct. Light)

Figure: 2x nearest neighbor (zoom).
Qualitative Evaluation: Middlebury Cones (Struct. Light)

Figure: 2x SR result of our method (zoom).
Qualitative Evaluation: Middlebury Cones (Struct. Light)

Figure: 2x SR of Glasner et al. [8] (zoom).
Qualitative Evaluation: Middlebury Cones (Struct. Light)

Figure: 2x SR of Mac Aodha et al. [12] (zoom).
Qualitative Evaluation: Middlebury Cones (Struct. Light)

Figure: 2x SR of Yang et al. [20] (zoom).
Qualitative Evaluation: Middlebury Cones (Struct. Light)

Figure: 2x SR of Freeman and Liu [7] (zoom).
Qualitative Evaluation: Middlebury Cones (Struct. Light)

Figure: 2x SR of Diebel and Thrun [5] (zoom).
Qualitative Evaluation: Middlebury Cones (Struct. Light)

Figure: 2x SR of Yang et al. [21] (zoom).
Qualitative Evaluation: Middlebury Cones (Struct. Light)

Figure: 2x nearest neighbor (zoom, 32 bit).
Qualitative Evaluation: Middlebury Cones (Struct. Light)

Figure: 2x SR result of our method (zoom, 32 bit).
Qualitative Evaluation: Middlebury Cones (Struct. Light)

Figure: 2x SR result of our method (zoom, 8 bit).
Qualitative Evaluation: Middlebury Cones (Struct. Light)

Figure: 2x SR result of Glasner et al. [8] (zoom, 8 bit).
Qualitative Evaluation: Middlebury Cones (Struct. Light)

Figure: 2x SR result of Mac Aodha et al. [12] (zoom, 8 bit).
## Root Mean Square Error (RMSE): Middlebury

<table>
<thead>
<tr>
<th>Method</th>
<th>Cones</th>
<th>Teddy</th>
<th>Tsukuba</th>
<th>Venus</th>
<th>Cones</th>
<th>Teddy</th>
<th>Tsukuba</th>
<th>Venus</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>
</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>
</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>
</tr>
<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>
</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>
</tr>
<tr>
<td>Glasner et al. [8]</td>
<td>0.867</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>
</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>
</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>
</tr>
</tbody>
</table>
Percent Error: Middlebury

<table>
<thead>
<tr>
<th>Method</th>
<th>Cones</th>
<th>Teddy</th>
<th>Tsukuba</th>
<th>Venus</th>
<th>Cones</th>
<th>Teddy</th>
<th>Tsukuba</th>
<th>Venus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nearest Neighbor</td>
<td>1.713</td>
<td>1.548</td>
<td>1.240</td>
<td>0.328</td>
<td>3.121</td>
<td>3.358</td>
<td>2.197</td>
<td>0.609</td>
</tr>
<tr>
<td>Diebel and Thrun [5]</td>
<td>3.800</td>
<td>2.786</td>
<td>2.745</td>
<td>0.574</td>
<td>7.452</td>
<td>6.865</td>
<td>5.118</td>
<td>1.236</td>
</tr>
<tr>
<td>Yang et al. [21]</td>
<td>2.346</td>
<td>1.918</td>
<td>1.161</td>
<td>0.250</td>
<td>4.582</td>
<td>4.079</td>
<td>2.565</td>
<td>0.421</td>
</tr>
<tr>
<td>Yang et al. [20]</td>
<td>61.617</td>
<td>54.194</td>
<td>5.566</td>
<td>46.985</td>
<td>63.742</td>
<td>55.080</td>
<td>7.649</td>
<td>47.053</td>
</tr>
<tr>
<td>Mac Aodha et al. [12]</td>
<td>2.935</td>
<td>2.311</td>
<td>2.235</td>
<td>0.536</td>
<td>6.541</td>
<td>5.309</td>
<td>4.780</td>
<td>0.856</td>
</tr>
<tr>
<td>Our Method</td>
<td>2.018</td>
<td>1.862</td>
<td>1.644</td>
<td>0.377</td>
<td>3.271</td>
<td>4.234</td>
<td>2.932</td>
<td>3.245</td>
</tr>
</tbody>
</table>
Summary

We presented a ‘single image’ depth SR algorithm, making 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 a technique for upscaling and merging matches that predicts sharp object boundaries at the target resolution. We showed our results to be highly competitive with methods leveraging ancillary data.
Michael Hornáček is funded by Microsoft Research through its European Ph.D. scholarship programme.
References

Questions?