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

M. Hornáček[1] C. Rhemann[2] M. Gelautz[1] C. Rother[2]

[1]Vienna University of Technology
Vienna, Austria

[2]Microsoft Research Ltd.
Cambridge, United Kingdom

June 20, 2013

Objective

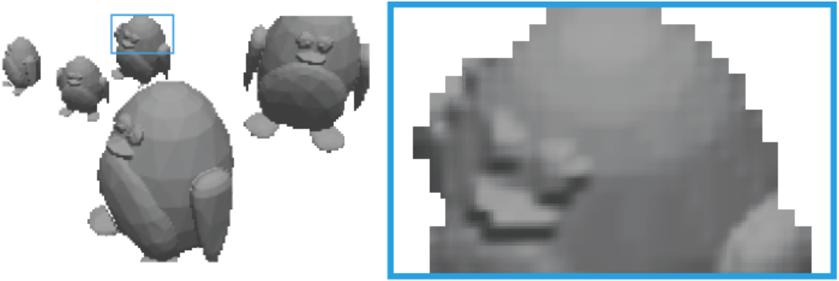
Input:

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

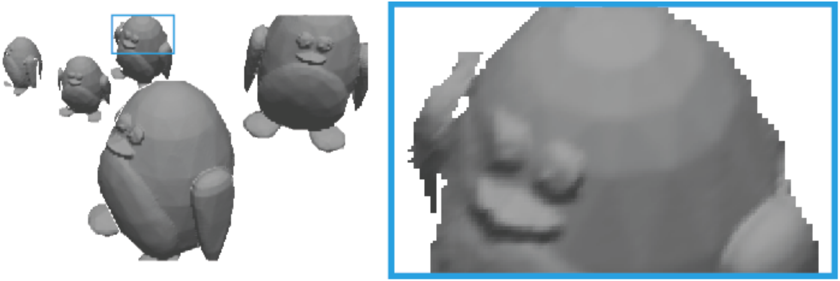
Objective:

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

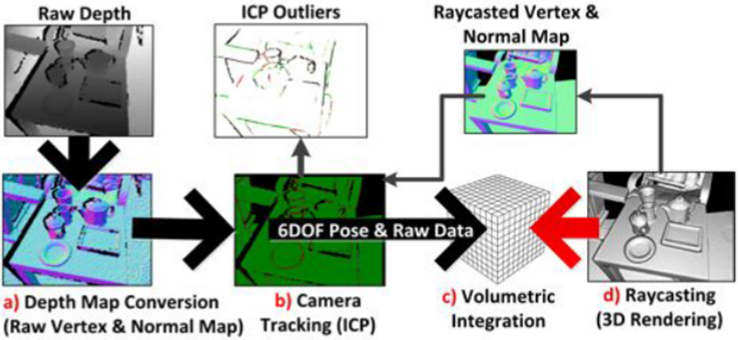


Figure : Izadi *et al.* [10] produce outstanding results by fusing a sequence of depth maps generated by a tracked Kinect camera into a single 3D representation.

Challenges: Ancillary Data or Multiple Depth Maps

Guiding image at target resolution or multiple depth maps often unavailable or difficult to obtain.

Related Work: 'Single Image' SR of Glasner *et al.*

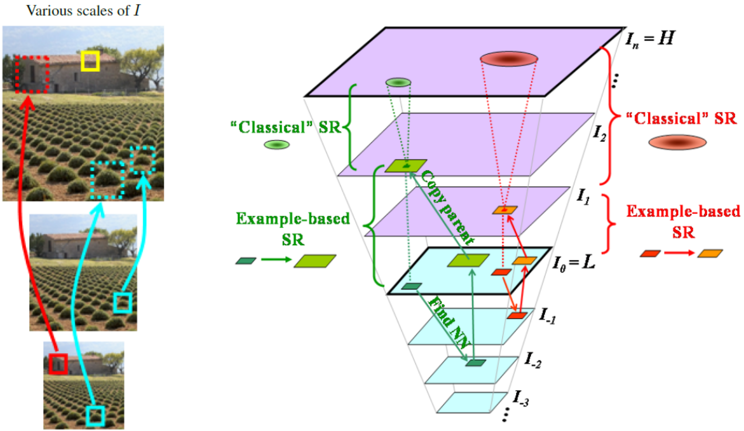


Figure : Assemble SR output using corresponding 5 × 5 pixel patches found across a discrete cascade of downscaled copies of input image.

Related Work: External Patch Database

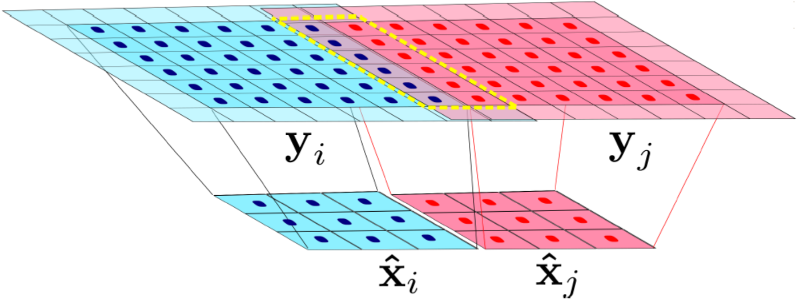


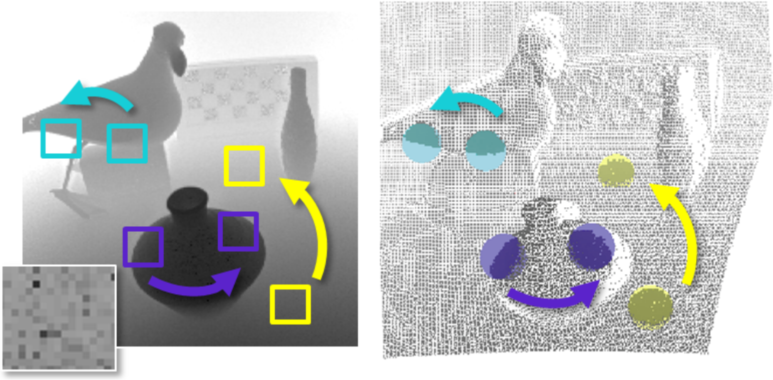
Figure : Mac Aodha *et al.* [12] assemble SR output using external database of 5.2 million high- resolution synthetic, noise-free 2D pixel patches.

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:

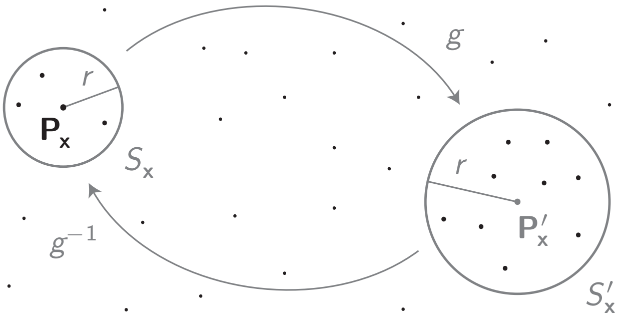
- Reasoning in terms of 3D point patches
- New 3D variant of PatchMatch (cf. Barnes *et al.* [1])
- Simple, yet effective patch upscaling and merging technique

Algorithm Overview

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



3D Point Patches

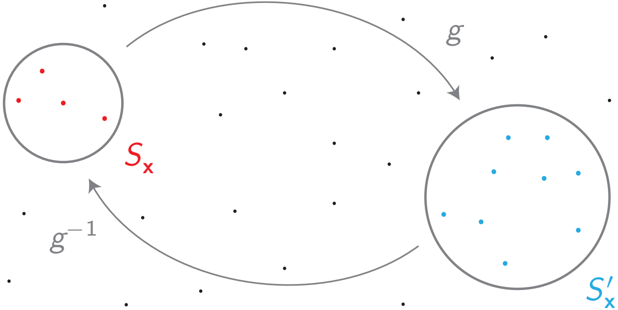
'Further' Patch $S_{\mathbf{x}} \subset \mathbb{R}^3$

Set of 3D points of input depth map within a fixed radius r of 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 depth encoded at $\mathbf{x}$ in input depth map and $\mathtt{K}$ is 3×3 camera calibration matrix.

'Closer' Patch $S'_{\mathbf{x}} \subset \mathbb{R}^3$

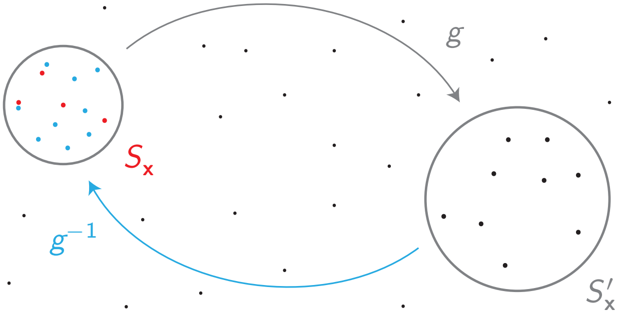
Set of 3D points of input depth map within the same r of point $\mathbf{P}'_{\mathbf{x}} = g(\mathbf{P}_{\mathbf{x}}) \in \mathbb{R}^3$, where $g = (\mathtt{R}, \mathbf{t}) \in SE(3)$ is a 6 DoF rigid body motion in 3D such that depth of $\mathbf{P}'_{\mathbf{x}}$ be less than or equal to that of $\mathbf{P}_{\mathbf{x}}$.

Patch Similarity: ‘Backward’ Cost $c^b(\mathbf{x}; g)$



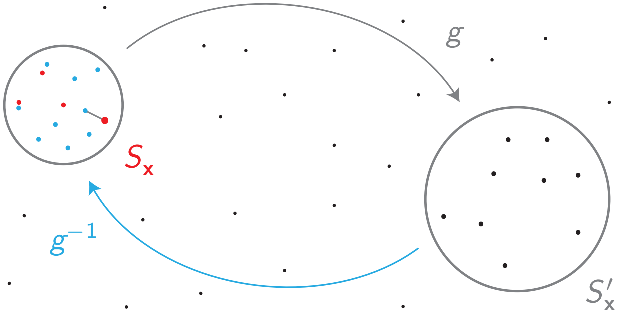
$$c^b(\mathbf{x}; g) = \sum_{\mathbf{P} \in S_{\mathbf{x}}} \left\| \mathbf{P} - \mathrm{NN}_{g^{-1}(S'_{\mathbf{x}})}(\mathbf{P}) \right\|_2^2 / |S_{\mathbf{x}}|$$

Patch Similarity: 'Backward' Cost $c^b(\mathbf{x}; g)$



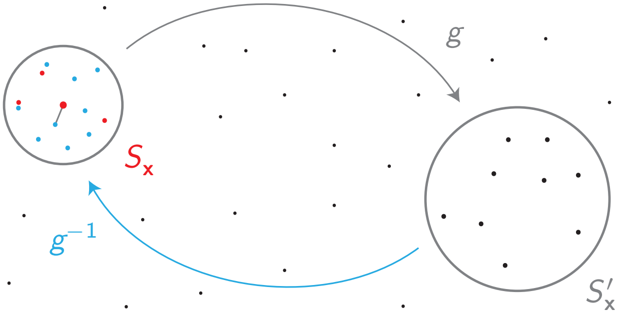
$$c^b(\mathbf{x}; g) = \sum_{\mathbf{P} \in S_{\mathbf{x}}} \left\| \mathbf{P} - \mathrm{NN}_{g^{-1}(S'_{\mathbf{x}})}(\mathbf{P}) \right\|_2^2 / |S_{\mathbf{x}}|$$

Patch Similarity: 'Backward' Cost $c^b(\mathbf{x}; g)$



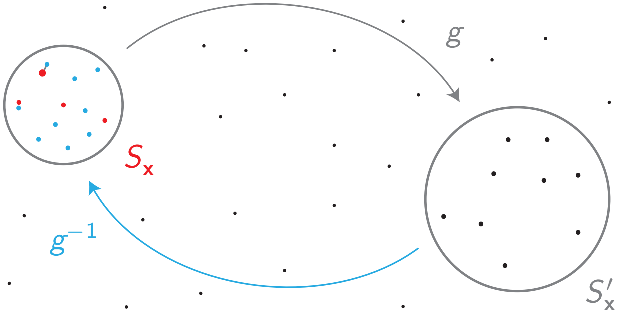
$$c^b(\mathbf{x}; g) = \sum_{\mathbf{P} \in S_\mathbf{x}} \left\| \mathbf{P} - \mathrm{NN}_{g^{-1}(S'_\mathbf{x})}(\mathbf{P}) \right\|_2^2 / |S_\mathbf{x}|$$

Patch Similarity: 'Backward' Cost $c^b(\mathbf{x}; g)$



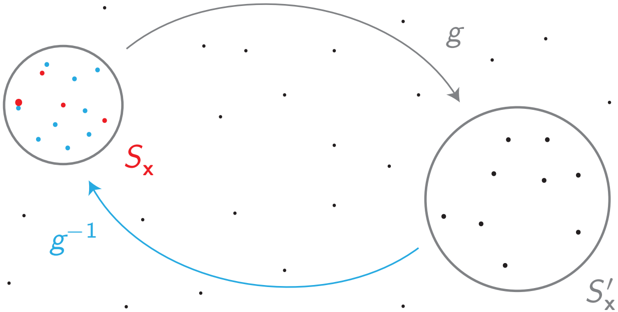
$$c^b(\mathbf{x}; g) = \sum_{\mathbf{P} \in S_\mathbf{x}} \left\| \mathbf{P} - \mathrm{NN}_{g^{-1}(S'_\mathbf{x})}(\mathbf{P}) \right\|_2^2 / |S_\mathbf{x}|$$

Patch Similarity: 'Backward' Cost $c^b(\mathbf{x}; g)$



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

Patch Similarity: 'Backward' Cost $c^b(\mathbf{x}; g)$

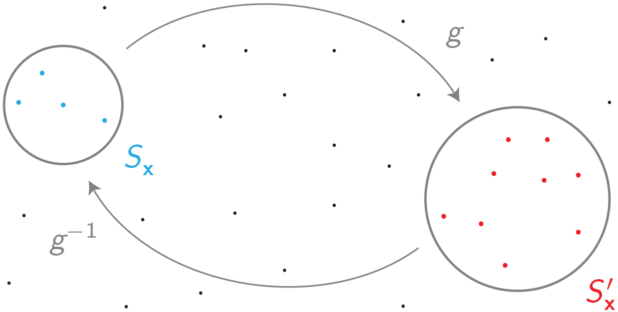


$$c^b(\mathbf{x}; g) = \sum_{\mathbf{P} \in S_\mathbf{x}} \left\| \mathbf{P} - \mathrm{NN}_{g^{-1}(S'_\mathbf{x})}(\mathbf{P}) \right\|_2^2 / |S_\mathbf{x}|$$

Patch Similarity: ‘Forward’ Cost $c^f(\mathbf{x}; g)$

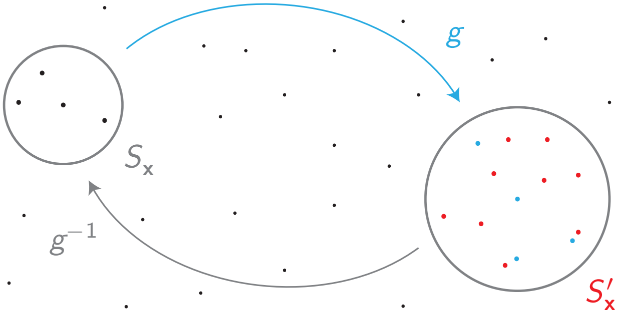
‘Backward’ cost $c^b(\mathbf{x}; g)$ computes patch similarity without penalizing addition of new detail. To be more confident that such new detail is reasonable, we also compute analogous ‘forward’ cost $c^f(\mathbf{x}; g)$.

Patch Similarity: 'Forward' Cost $c^f(\mathbf{x}; g)$



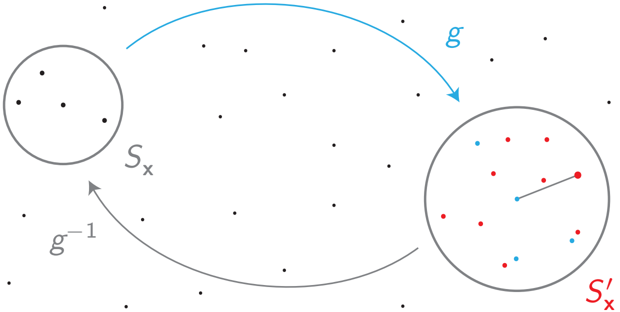
$$c^f(\mathbf{x}; g) = \sum_{\mathbf{P}' \in S'_{\mathbf{x}}} \left\| \mathbf{P}' - \mathrm{NN}_{g(S_{\mathbf{x}})}(\mathbf{P}') \right\|_2^2 / |S'_{\mathbf{x}}|$$

Patch Similarity: 'Forward' Cost $c^f(\mathbf{x}; g)$



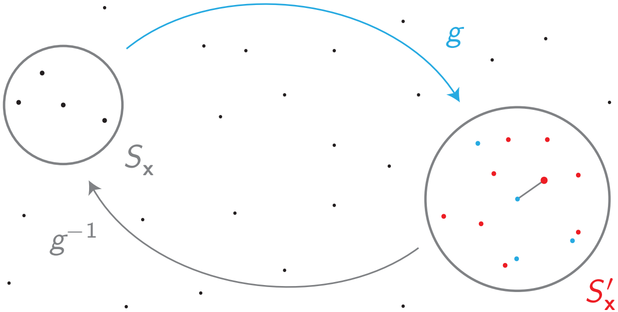
$$c^f(\mathbf{x}; g) = \sum_{\mathbf{P}' \in S'_\mathbf{x}} \left\| \mathbf{P}' - \mathrm{NN}_{g(S_\mathbf{x})}(\mathbf{P}') \right\|_2^2 / |S'_\mathbf{x}|$$

Patch Similarity: 'Forward' Cost $c^f(\mathbf{x}; g)$



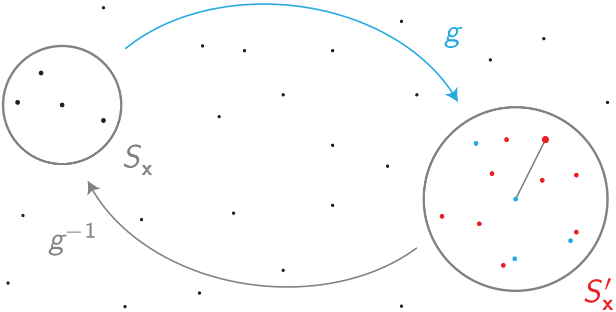
$$c^f(\mathbf{x}; g) = \sum_{\mathbf{P}' \in S'_\mathbf{x}} \left\| \mathbf{P}' - \mathrm{NN}_{g(S_\mathbf{x})}(\mathbf{P}') \right\|_2^2 / |S'_\mathbf{x}|$$

Patch Similarity: 'Forward' Cost $c^f(\mathbf{x}; g)$



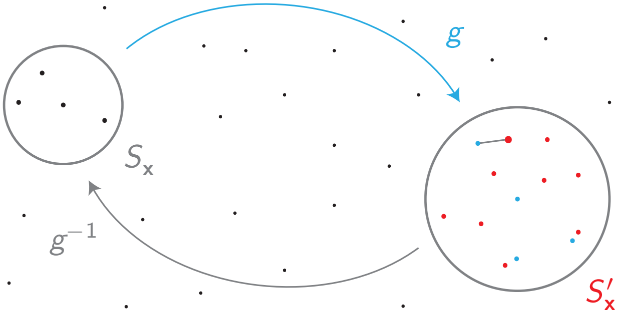
$$c^f(\mathbf{x}; g) = \sum_{\mathbf{P}' \in S'_\mathbf{x}} \left\| \mathbf{P}' - \mathrm{NN}_{g(S_\mathbf{x})}(\mathbf{P}') \right\|_2^2 / |S'_\mathbf{x}|$$

Patch Similarity: 'Forward' Cost $c^f(\mathbf{x}; g)$



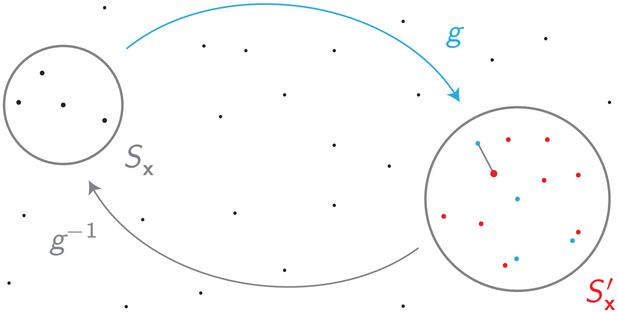
$$c^f(\mathbf{x}; g) = \sum_{\mathbf{P}' \in S'_\mathbf{x}} \left\| \mathbf{P}' - \mathrm{NN}_{g(S_\mathbf{x})}(\mathbf{P}') \right\|_2^2 / |S'_\mathbf{x}|$$

Patch Similarity: 'Forward' Cost $c^f(\mathbf{x}; g)$



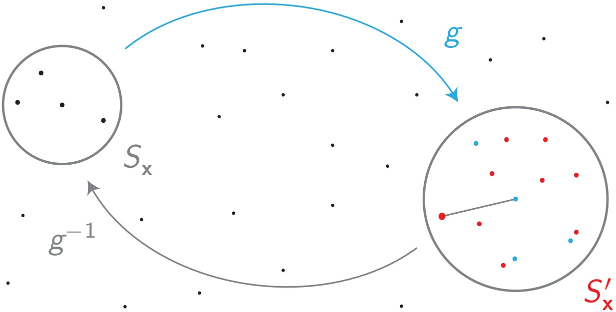
$$c^f(\mathbf{x}; g) = \sum_{\mathbf{P}' \in S'_{\mathbf{x}}} \left\| \mathbf{P}' - \mathrm{NN}_{g(S_{\mathbf{x}})}(\mathbf{P}') \right\|_2^2 / |S'_{\mathbf{x}}|$$

Patch Similarity: 'Forward' Cost $c^f(\mathbf{x}; g)$



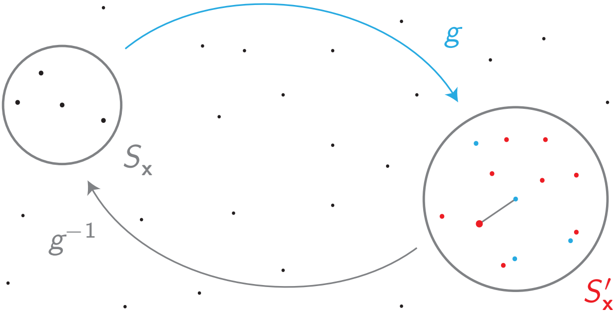
$$c^f(\mathbf{x}; g) = \sum_{\mathbf{P}' \in S'_{\mathbf{x}}} \left\| \mathbf{P}' - \mathrm{NN}_{g(S_{\mathbf{x}})}(\mathbf{P}') \right\|_2^2 / |S'_{\mathbf{x}}|$$

Patch Similarity: ‘Forward’ Cost $c^f(\mathbf{x}; g)$



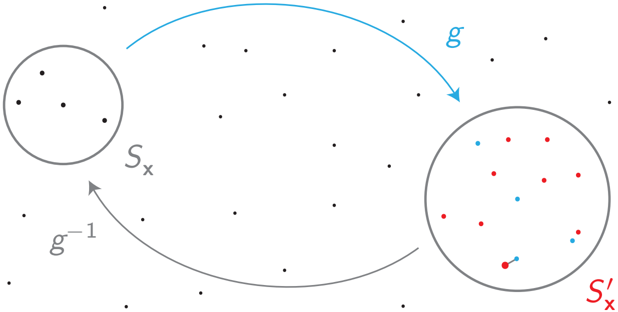
$$c^f(\mathbf{x}; g) = \sum_{\mathbf{P}' \in S'_{\mathbf{x}}} \left\| \mathbf{P}' - \mathrm{NN}_{g(S_{\mathbf{x}})}(\mathbf{P}') \right\|_2^2 / |S'_{\mathbf{x}}|$$

Patch Similarity: ‘Forward’ Cost $c^f(\mathbf{x}; g)$



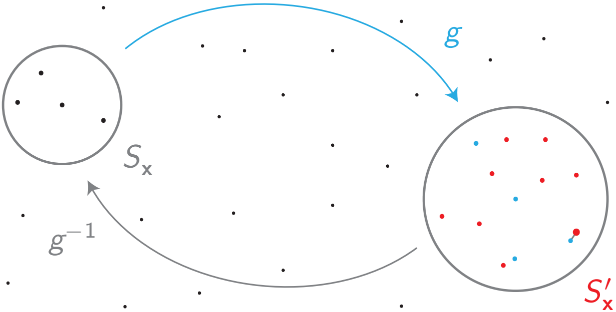
$$c^f(\mathbf{x}; g) = \sum_{\mathbf{P}' \in S'_\mathbf{x}} \left\| \mathbf{P}' - \mathrm{NN}_{g(S_\mathbf{x})}(\mathbf{P}') \right\|_2^2 / |S'_\mathbf{x}|$$

Patch Similarity: 'Forward' Cost $c^f(\mathbf{x}; g)$



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

Patch Similarity: ‘Forward’ Cost $c^f(\mathbf{x}; g)$



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

Patch Similarity: Matching Cost $c(\mathbf{x}; g)$

We compute 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 $\alpha' = 1 - \alpha$.

Patch Similarity: Validity of g at $\mathbf{x}$

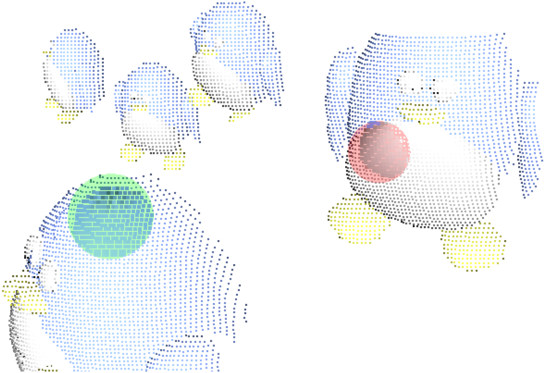
We deem a rigid body motion g valid at $\mathbf{x}$ if

- $\|\mathbf{P}_\mathbf{x} - \mathbf{P}'_\mathbf{x}\|_2 \geq r$ to prevent trivial minimization
- $|S'_\mathbf{x}| \geq |S_\mathbf{x}| \geq 3$ to match to at least as many points

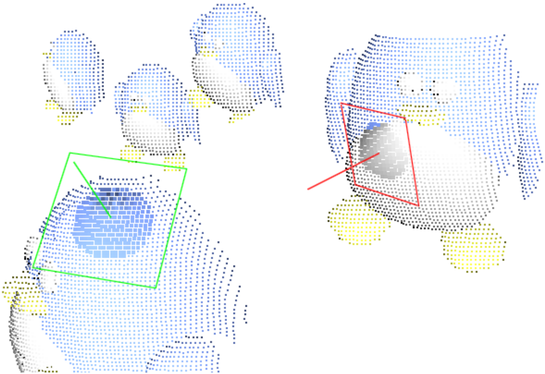
3D PatchMatch for Dense Correspondence Search

Assign to each input pixel $\mathbf{x}$ a valid 6 DoF 3D rigid body motion $g_{\mathbf{x}}$ by (semi-)random initialization followed by i iterations propagation and refinement.

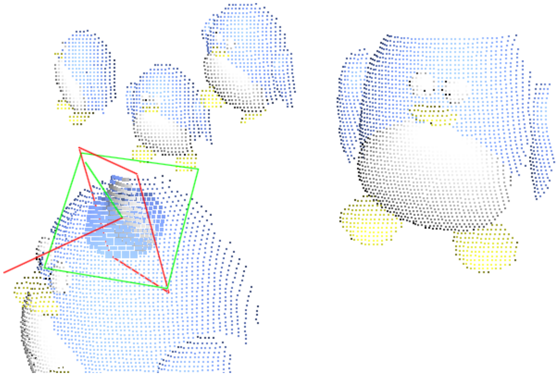
3D PatchMatch: Semi-Random Initialization



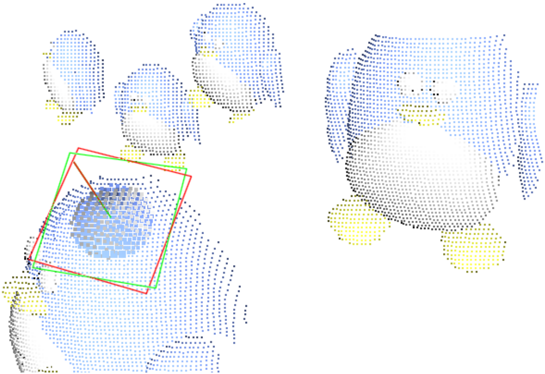
3D PatchMatch: Semi-Random Initialization



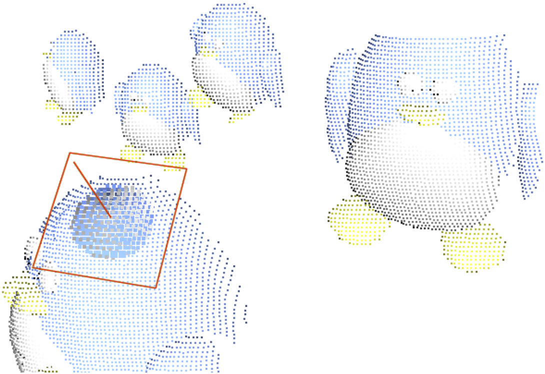
3D PatchMatch: Semi-Random Initialization (1/3)



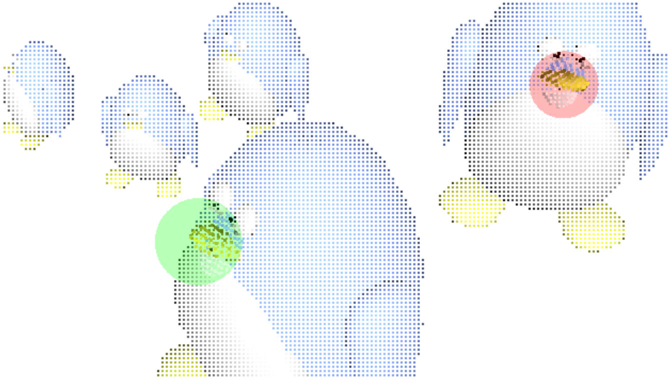
3D PatchMatch: Semi-Random Initialization (2/3)



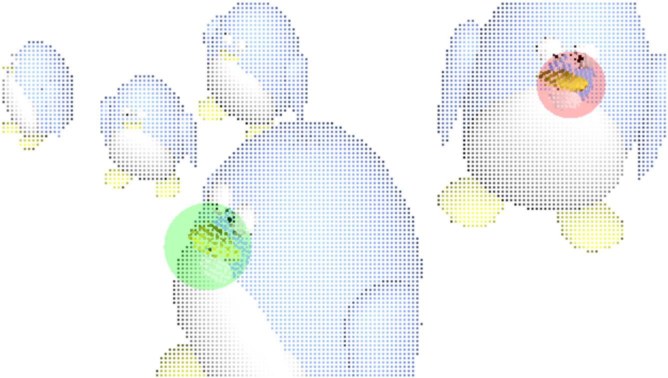
3D PatchMatch: Semi-Random Initialization (3/3)



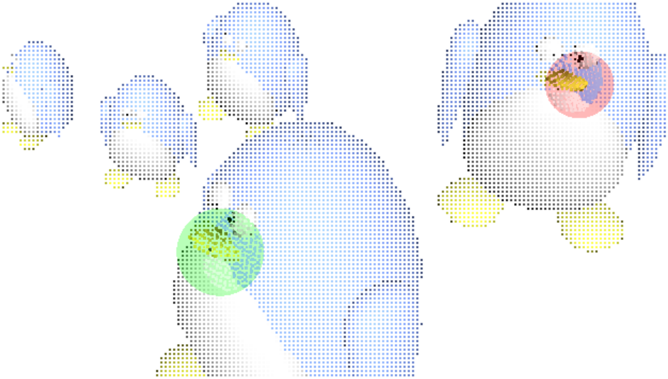
3D PatchMatch: Propagation



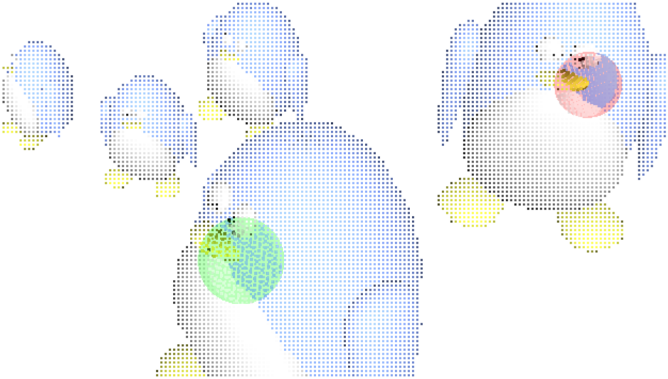
3D PatchMatch: Propagation



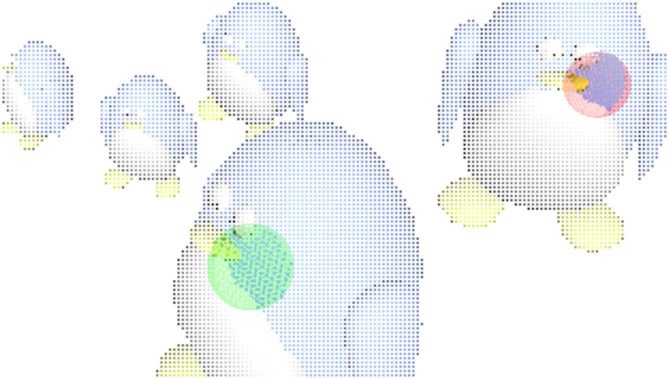
3D PatchMatch: Propagation



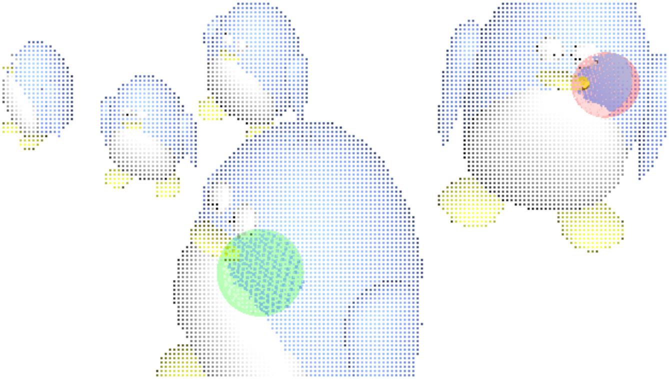
3D PatchMatch: Propagation



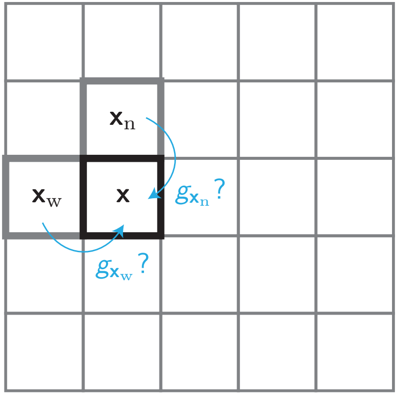
3D PatchMatch: Propagation



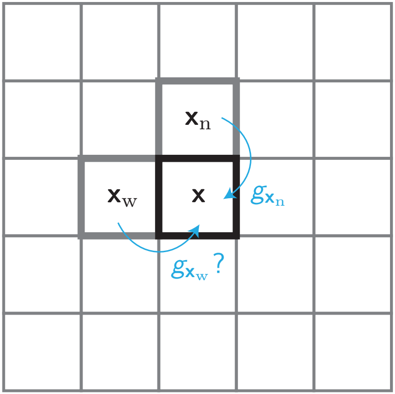
3D PatchMatch: Propagation



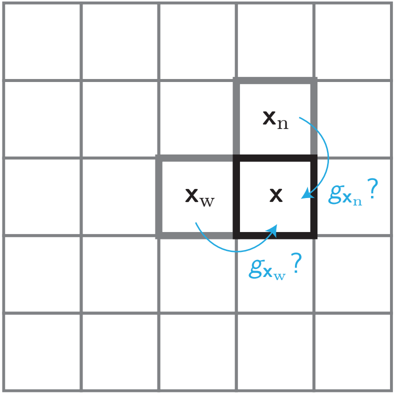
3D PatchMatch: Propagation (Odd Iterations *i*)



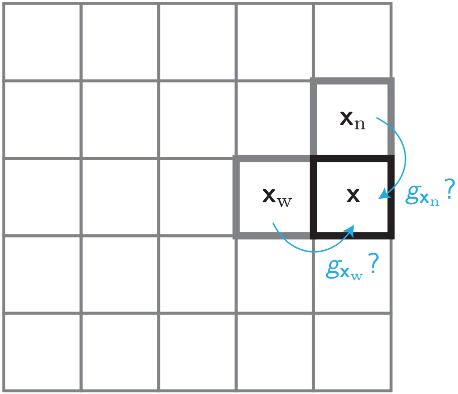
3D PatchMatch: Propagation (Odd Iterations i)



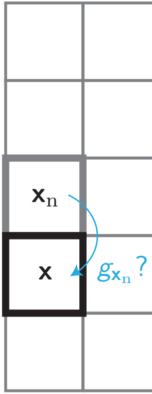
3D PatchMatch: Propagation (Odd Iterations *i*)



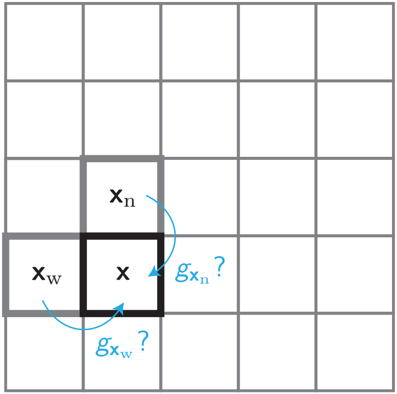
3D PatchMatch: Propagation (Odd Iterations i)



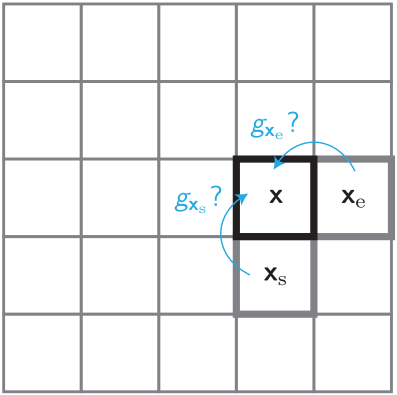
3D PatchMatch: Propagation (Odd Iterations i)



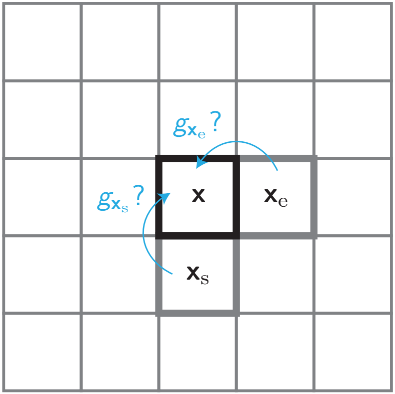
3D PatchMatch: Propagation (Odd Iterations i)



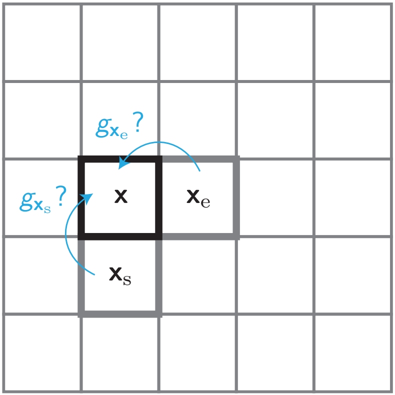
3D PatchMatch: Propagation (Even Iterations i)



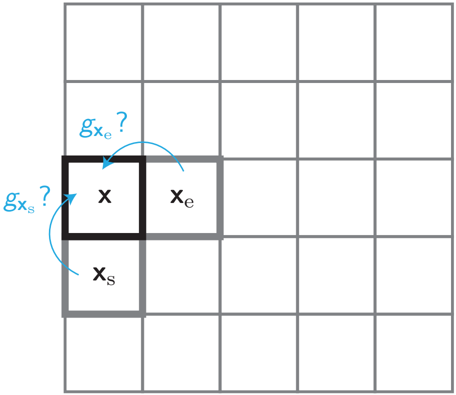
3D PatchMatch: Propagation (Even Iterations i)



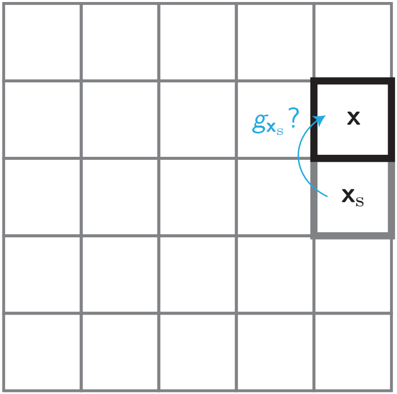
3D PatchMatch: Propagation (Even Iterations i)



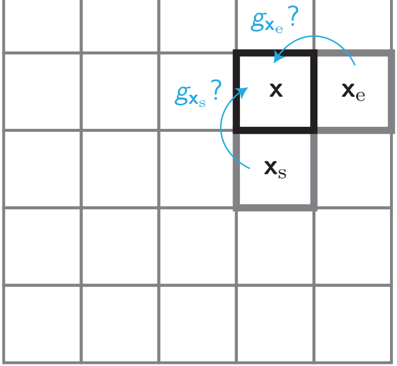
3D PatchMatch: Propagation (Even Iterations i)



3D PatchMatch: Propagation (Even Iterations i)



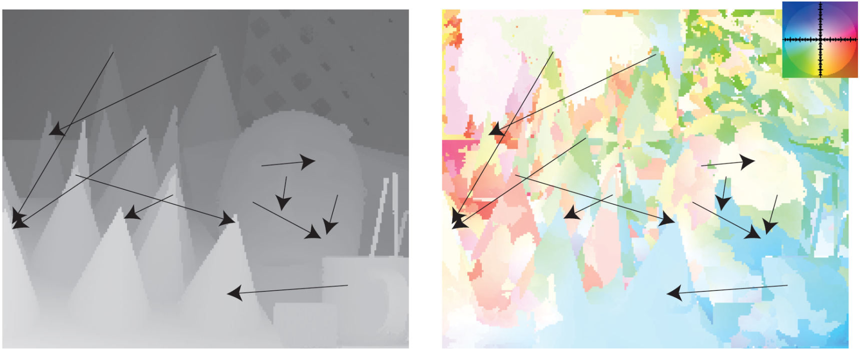
3D PatchMatch: Propagation (Even Iterations i)



3D PatchMatch: Refinement

We independently carry out k iterations of additional initialization and of perturbation of the translational and rotational components of $g_{\mathbf{x}}$.

3D PatchMatch: Visualization



Putting It All Together?

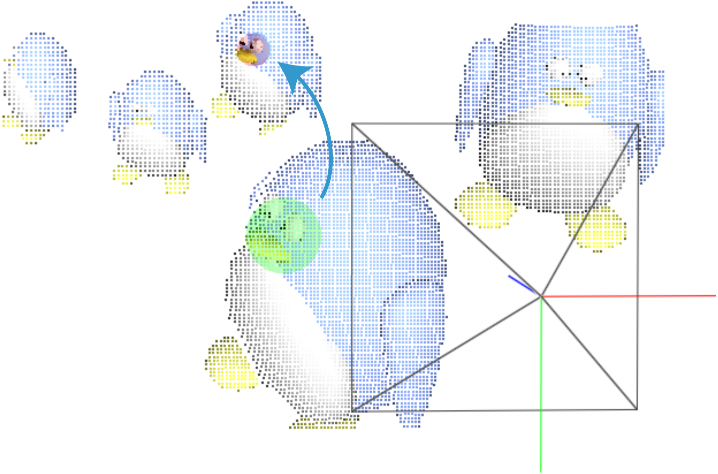
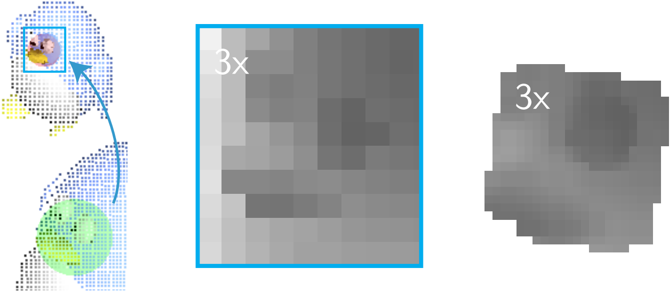


Figure : Overlapping matches? Object boundaries?

Patch Upscaling and Merging: Overlay Masks (1/2)

Patch Upscaling and Merging: Overlay Masks (2/2)

Patch Upscaling and Merging: Overlay Patches



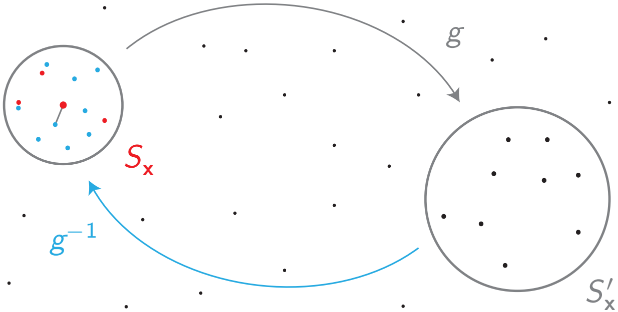
Patch Upscaling and Merging: Merging

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

$$\omega_{\mathbf{x}} = \exp\left(-\gamma \cdot c^b(\mathbf{x}; g_{\mathbf{x}})\right).$$

If $c^b(\mathbf{x}; g_{\mathbf{x}}) > \beta$, we instead use overlay patch at $\mathbf{x}$ corresponding to identity motion.

Reminder: 'Backward' Cost $c^b(\mathbf{x}; g)$



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

Qualitative Evaluation: Egg Cartons (Stereo)

Figure : Color image.

Qualitative Evaluation: Egg Cartons (Stereo)

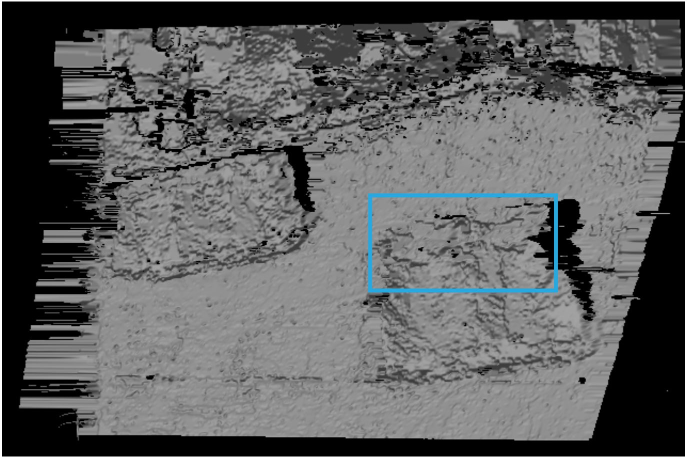


Figure : 2x nearest neighbor (32 bit).

Qualitative Evaluation: Egg Cartons (Stereo)

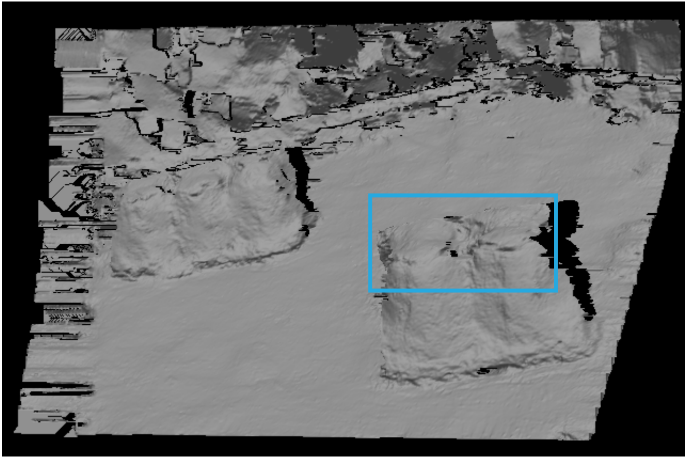


Figure : 2x SR result of our method (32 bit).

Qualitative Evaluation: Egg Cartons (Stereo)

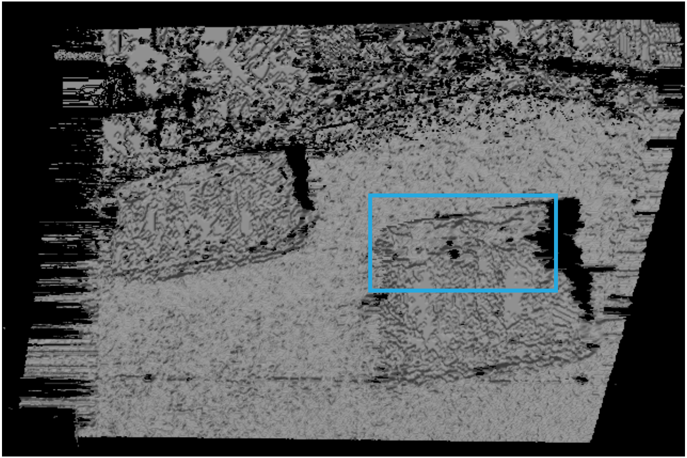


Figure : 2x SR result of Glasner *et al.* [8] (8 bit).

Qualitative Evaluation: Egg Cartons (Stereo)

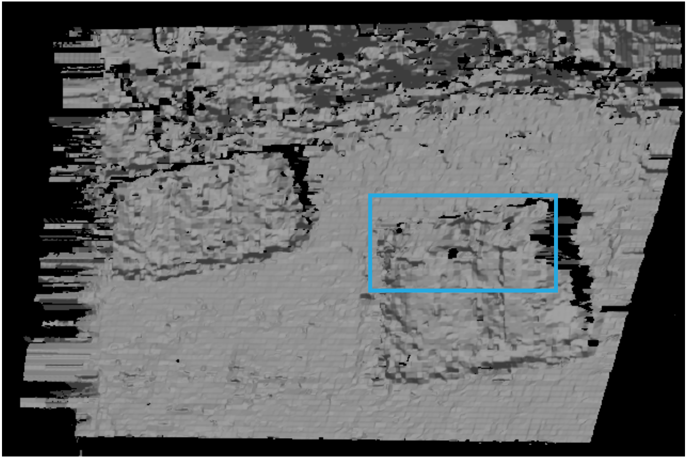


Figure : 2x SR result of Mac Aodha *et al.* [12] (preprocessed, 32 bit).

Qualitative Evaluation: Egg Cartons (Stereo)

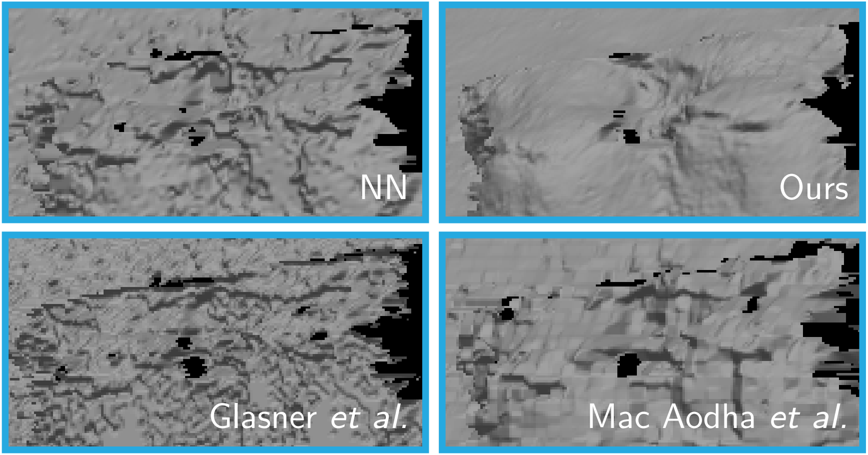


Figure : Zooms.

Qualitative Evaluation: Gull (ToF)

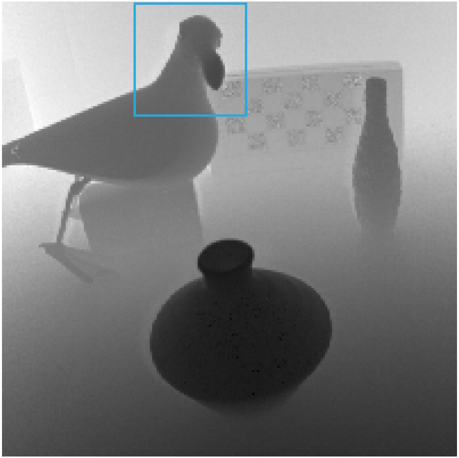


Figure : 4x nearest neighbor.

Qualitative Evaluation: Gull (ToF)

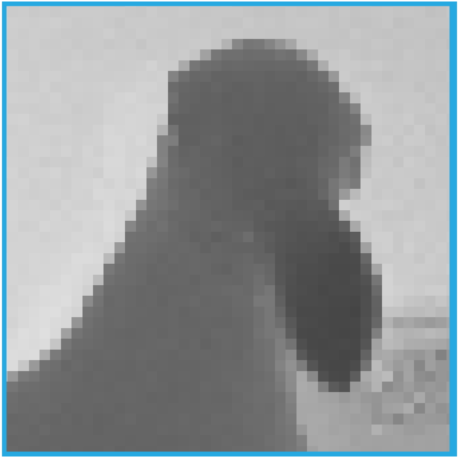


Figure : 4x nearest neighbor (zoom).

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)

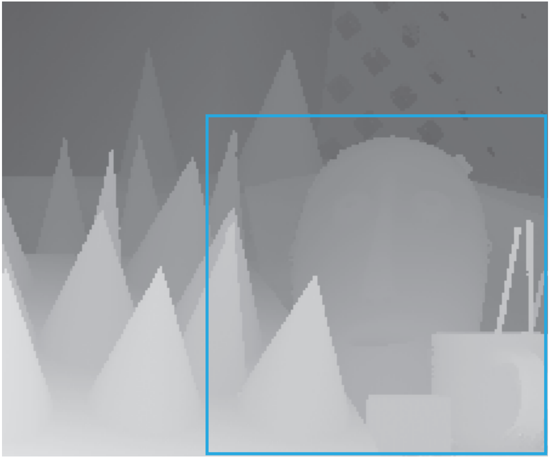


Figure : 2x nearest neighbor.

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)

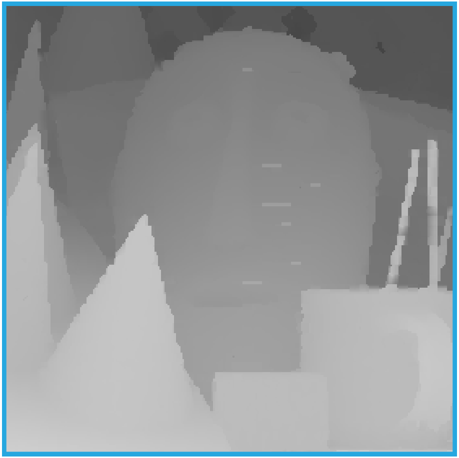


Figure : 2x SR of Mac Aodha *et al.* [12] (zoom).

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)

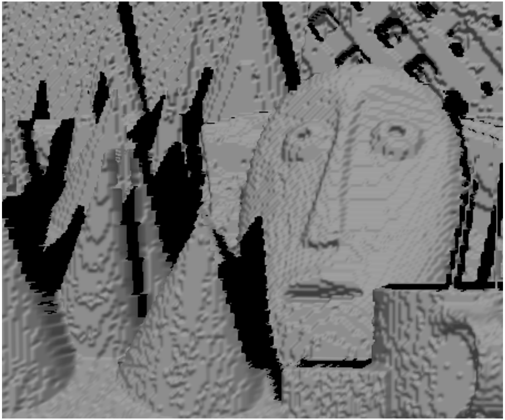


Figure : 2x nearest neighbor (zoom, 32 bit).

Qualitative Evaluation: Middlebury Cones (Struct. Light)

Figure : 2x SR result of our method (zoom, 32 bit).

Qualitative Evaluation: Middlebury Cones (Struct. Light)

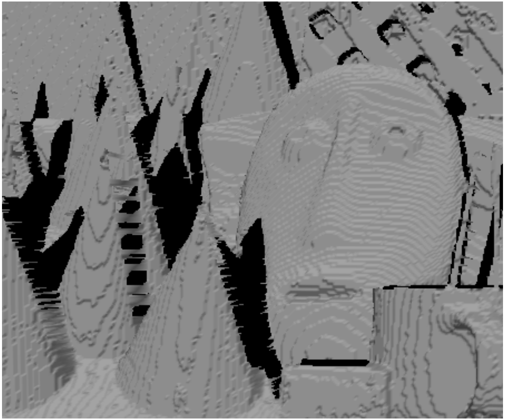


Figure : 2x SR result of our method (zoom, 8 bit).

Qualitative Evaluation: Middlebury Cones (Struct. Light)

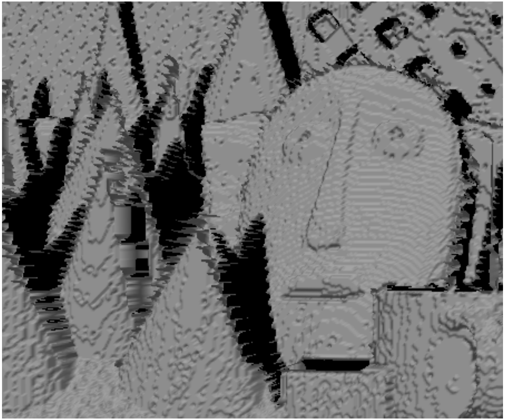


Figure : 2x SR result of Glasner *et al.* [8] (zoom, 8 bit).

Qualitative Evaluation: Middlebury Cones (Struct. Light)

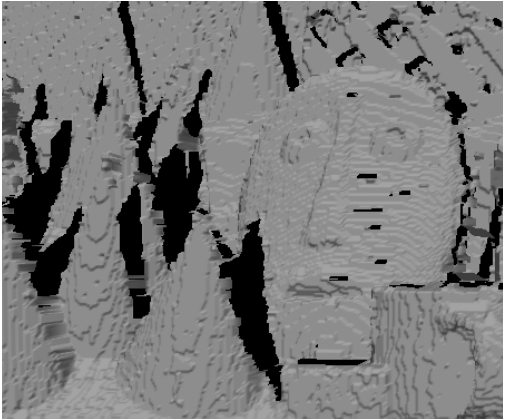


Figure : 2x SR result of Mac Aodha *et al.* [12] (zoom, 8 bit).

Root Mean Square Error (RMSE): Middlebury

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

Percent Error: Middlebury

	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

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.

Acknowledgement

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

References

Provided in: Michael Hornáček, Christoph Rhemann, Margrit Gelautz, Carsten Rother. Depth Super Resolution by Rigid Body Self-Similarity in 3D. In *CVPR*, 2013.

Questions?