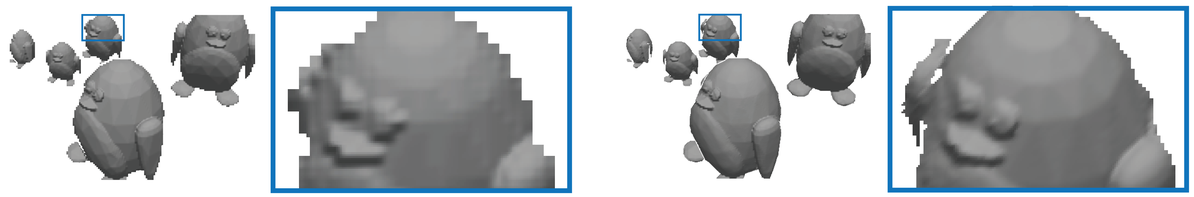
Depth Super Resolution by Rigid Body Self-Similarity in 3D

Michael Hornáček[1,*], Christoph Rhemann[2], Margrit Gelautz[1], Carsten Rother[2]
[1]Vienna University of Technology [2]Microsoft Research Cambridge

Microsoft Research

Objective

Input: Single low-resolution, noisy, and perhaps heavily quantized depth map
Objective: Jointly increase spatial resolution and apparent measurement accuracy (e.g., depth resolution)



Left: 3x nearest neighbor upscaling. Right: 3x SR output of our algorithm.

Contributions

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

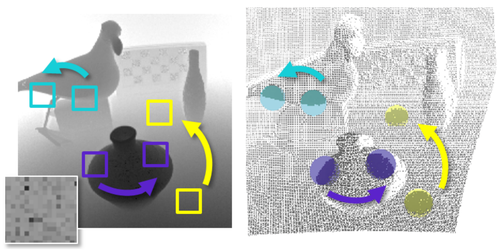
- Reasoning in terms of 3D point patches
- 3D variant of PatchMatch
- Patch upscaling and merging technique

Why is it Hard?

Most techniques rely on ancillary data that is often unavailable or difficult to obtain (e.g., aligned guiding image at target resolution).

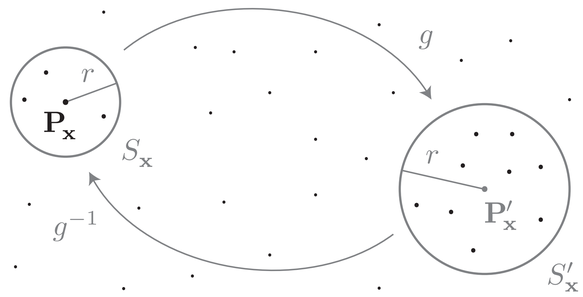
Proceeding 'by example'—by assembling SR output from matched 2D pixel patches—poses its own challenges:

- Different patch depths (depth normalization?)
- Projective distortions (calls for small patches)
- Object boundaries (discontinuity handling?)



Left: three dissimilar pairs of 2D pixel patches.
Right: analogous 3D point patch pairs similar.

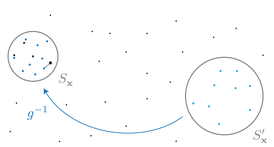
3D Point Patches



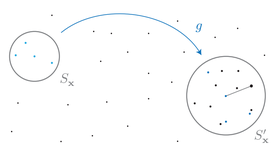
6 DoF 3D rigid body motion $g \in SE(3)$ relating 3D point patches $S_\mathbf{x}, S'_\mathbf{x} \subset \mathbb{R}^3$. Point $\mathbf{P}_\mathbf{x}$ is point encoded at pixel $\mathbf{x}$ of input depth map and is center point of 'further' patch $S_\mathbf{x}$. Point $\mathbf{P}'_\mathbf{x} = g(\mathbf{P}_\mathbf{x})$ is center point of 'closer' patch $S'_\mathbf{x}$. Radius r is kept same for all patches.

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

'Backward' cost $c^b(\mathbf{x}; g)$ computes patch similarity by SSD over nearest neighbors of $S_\mathbf{x}$ in $g^{-1}(S'_\mathbf{x})$, which does not penalize addition of new detail. To be more confident that such new detail is reasonable, we also compute analogous 'forward' cost $c^f(\mathbf{x}; g)$.



'Backward' cost $c^b(\mathbf{x};g)$.



'Forward' cost $c^f(\mathbf{x};g)$.

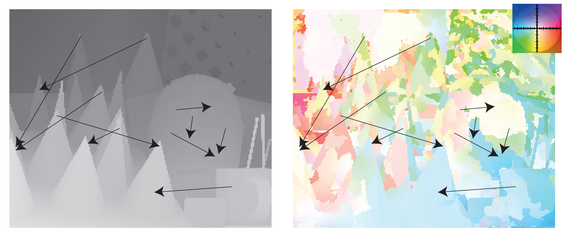
Matching cost $c(\mathbf{x}; g)$ over which we minimize given by convex combination of 'backward' and 'forward' cost.

Algorithm

Our proposed 'single image' depth SR algorithm reduces to two steps:

1. Obtain dense 6 DoF correspondence field over input pixels $\mathbf{x}$ using new 3D variant of PatchMatch algorithm of Barnes *et al.*
2. Populate SR pixels $\hat{\mathbf{x}}$ of output depth map at target resolution using novel patch upscaling and merging technique

Dense 6 DoF Correspondence Search



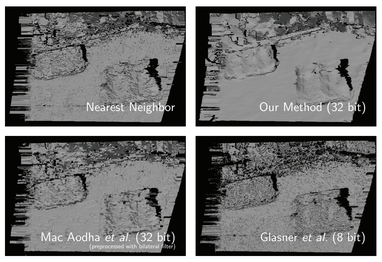
Visualization of projected 3D displacements of output dense 6 DoF rigid body assignment of our 3D PatchMatch variant.

Patch Upscaling and Merging

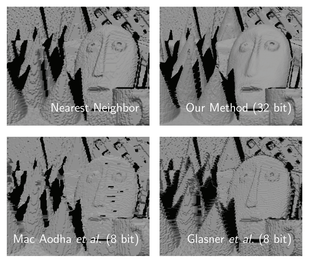
SR output generated by weighted sum over interpolated depth values of 'backward'-transformed points $g_\mathbf{x}^{-1}(S'_\mathbf{x})$. Patch weight computed as function of $c^b(\mathbf{x}; g_\mathbf{x})$ in order to promote addition of new detail.

At input resolution, 'backward'-transformed 3D points $g_\mathbf{x}^{-1}(S'_\mathbf{x})$ allowed to influence only 2D pixels corresponding to $S_\mathbf{x}$, since it is over these pixels that matching cost $c(\mathbf{x}; g_\mathbf{x})$ was computed. At target resolution, we carry out polygon approximation of pixel mask. It is over SR pixels $\hat{\mathbf{x}}$ of polygonalized mask that depth values from $g_\mathbf{x}^{-1}(S'_\mathbf{x})$ are interpolated.

Qualitative Results



2x NN and SR (stereo).



2x NN and SR (structured light).

4x NN and SR (ToF).

Quantitative Results (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	0.740	0.527	0.401	0.170	1.141	0.801	0.549	0.243
Yang *et al.*	0.756	0.510	0.393	0.167	0.993	0.690	0.514	0.216
Yang *et al.*	2.027	1.420	0.705	0.992	2.214	1.572	0.840	1.012
Freeman and Liu	1.447	0.969	0.617	0.332	1.536	1.110	0.869	0.367
Glasner *et al.*	**0.867**	**0.596**	**0.482**	**0.209**	1.483	1.065	0.832	0.394
Mac Aodha *et al.*	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

Root mean square error (RMSE).

	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	3.800	2.786	2.745	0.574	7.452	6.865	5.118	1.236
Yang *et al.*	2.346	1.918	1.161	0.250	4.582	4.079	2.565	0.421
Yang *et al.*	61.617	54.194	5.566	46.985	63.742	55.080	7.649	47.053
Freeman and Liu	6.266	4.660	3.240	0.790	15.077	12.122	10.030	3.348
Glasner *et al.*	4.697	3.137	3.234	0.940	8.790	6.806	6.454	1.770
Mac Aodha *et al.*	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

Percent error.

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