Live Dense Reconstruction and Tracking

Richard A. Newcombe

Imperial College, London

November 12, 2011

Outline

1. Multiple view stereo vs. SLAM
2. Live Dense Reconstruction Using a Single Passive Camera
3. DTAM: Dense Tracking and Mapping
4. KinectFusion: Real-Time Dense Surface Mapping and Tracking with Kinect
5. Real-time Surface Fusion using a Single RGB Camera
6. Conclusions and Future DTAM Approaches

Motivations

Live incremental scene reconstruction vs. Offline batch MVS

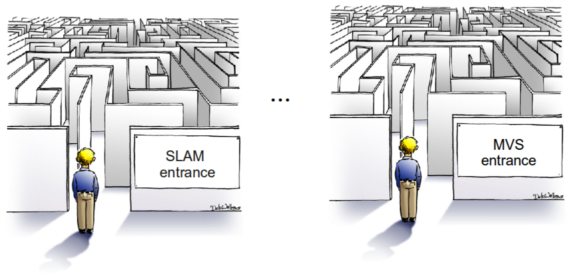


Figure: Setting off on a path to Live Dense Reconstruction.

Motivations

Scene Interaction vs. Navigation

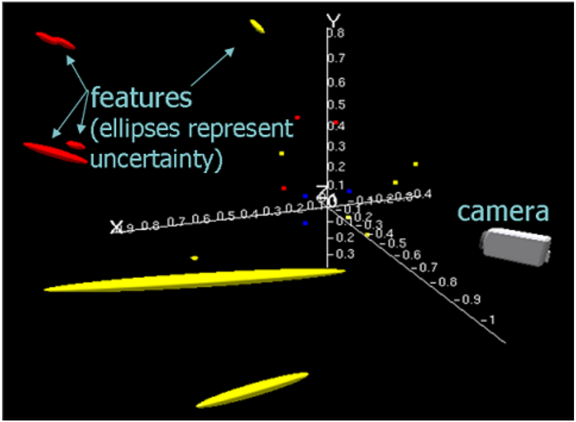
A robot needs sense of its surrounding surfaces if it is to competently interact with it. This is quite a different challenge to modelling the scene for navigation purposes alone.

Surface information for physical prediction

We can usefully recognize an object by utilising physical model properties – for example when we ask:
"Where is (the) chair?" (Visual recognition/search problem),
Do we really mean
"Where can I sit?" (Physically constrained embodied problem).

Real time, commodity SLAM system evolution

2003 Davison's Monoslam: importance of a cheap comodity sensor. Modelled and propagated joint uncertainty in real-time.

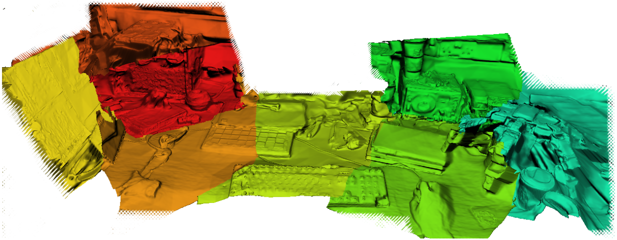


Real time, commodity SLAM system evolution

2007,2008 Klein and Murray's PTAM, also passive, optimised software using features of the CPU. Maps are much denser than monoSLAM, but still not surfaces.

Real time, commodity SLAM system evolution

2010 Newcombe and Davison, augmenting the sparse tracking and mapping with dense surface estimation method. Utilising GPU power, live but not real-time and no way to correct grossly wrong geometry.



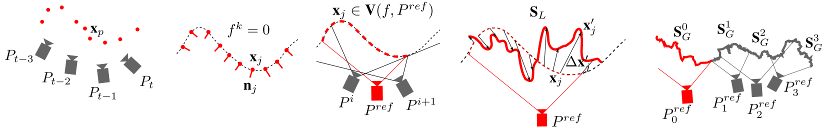
J. Stuehmer et al also augment the real-time SFM system but obtain real-time depth maps (without stiching/fusion). Also early work by Pollefeys et al 2007, on real-time reconstruction of Urban scenes.

Live Dense Reconstruction

CVPR 2010 work with A.J. Davison overview.

Hyrbrid approach

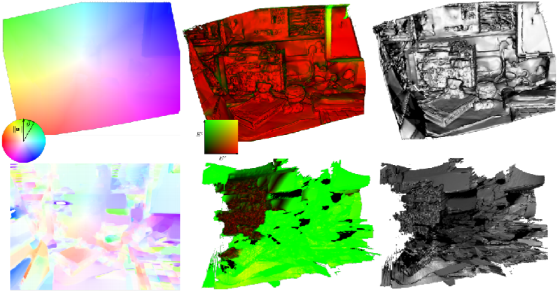
We use live estimated camera poses and sparse point clouds to build a coarse base mesh model. We then upgrade the surface reconstruction using variational optical flow to obtain dense correspondences with which depth maps are computed and stitched together.



Dense Correspondences from Optical Flow

Optical flow initialised with surface prediction

We found that the coarse surface prediction greatly improves optic flow computation when the displacement between views breaks the linearisation assumption that must hold in the coarsest level of the optic flow estimate.

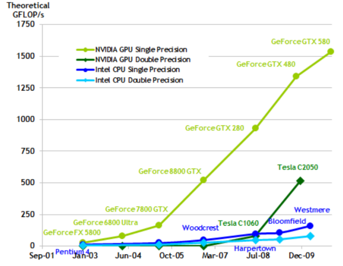


...Nothing more practical than a good theory

- The elegance of an algorithm is often tied to the current computing paradigm.
- A gradient descent/ascent optimisation of global energy functions for many low level vision problems exist that maps very efficiently to GPU hardware.

Amazing commodity hardware capabilities

GPGPU:
Massive processing capabilities



Key Technology

Powerful GPGPU processing

Liberates us from worrying (too much) about efficiency before understand the core approaches possible.

- e.g. MonoSLAM/PTAM struggles with 100s/1000s of point features but now we can integrate and track millions of points per second.
- Computational requirement hockey stick: once we get to a certain capability, certain representations are feasible that enable integration of all data all of the time.

Dense Tracking *and* Mapping

ICCV 2011 work with S.J. Lovegrove and A.J. Davison. In this work we optimise the live reconstruction pipeline by optimising directly for surfaces using an inverse depth map parametrisation of the scene.

Dense Mapping

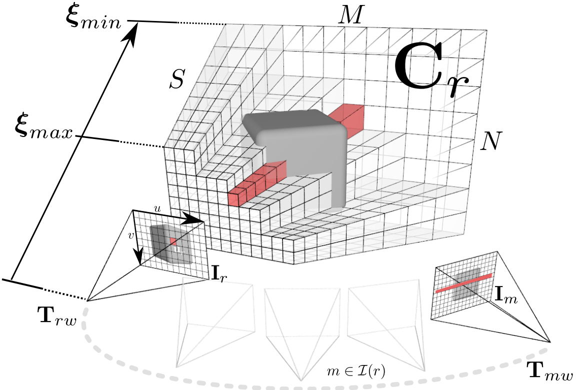
We exploit the epipolar constraint given currently estimated camera poses to obtain high quality depth maps from 100s of small baseline images.

Dense Tracking

We close the tracking and mapping loop by tracking the camera pose using the current dense surface prediction moving away from sparse features and point clouds altogether.

Cost volume data term

Build a cost volume from lots of weak data terms, and then using a simple discontinuity preserving smoothness prior, optimise global energy.



Using all possible frames from the live camera

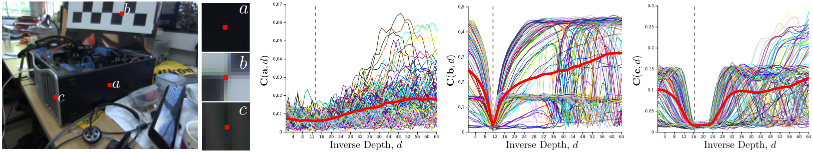


Figure: Plots for the single pixel photometric functions (absolute differences of RGB values) and the sum across multiple images (shown in thick red line).

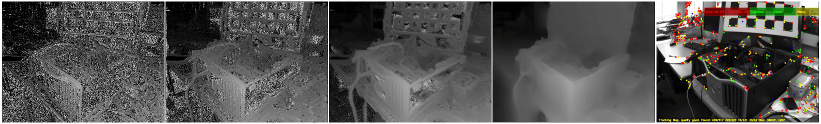


Figure: Per pixel inverse depth minimum for increasing numbers of data terms (shown in left three), in comparison to the sparse points found by binary data-association methods in PTAM.

Energy with the non-convex data term

We pair the cost volume with a (convex) discontinuity preserving spatial smoothness prior:

$$E_{\boldsymbol{\xi}} = \int_{\Omega} \left\{ g(\mathbf{u}) \|\nabla \boldsymbol{\xi}(\mathbf{u})\|_{\epsilon} + \lambda \mathbf{C}\left(\mathbf{u}, \boldsymbol{\xi}(\mathbf{u})\right) \right\} \mathrm{d}\mathbf{u} \ .$$

- The energy functional contains a non-convex photometric error data term across all images and a convex regulariser.
- Typically the data term would be linearised and solved in a coarse to fine scheme.

Tracking using the dense model

The dense surface prediction enables a simple way to perform camera tracking using all possible pixels in the live image: $\mathbb{SE}(3)$ pose esimtaiton using a $2.5D$ Luckas-Kanade style optimisation with a per pixel data error:

$$f_{\mathbf{u}}(\psi) = \mathbf{I}_l\left(\pi\left(\mathrm{K}\mathrm{T}_{lv}(\psi)\pi^{-1}\left(\mathbf{u}, \xi_v\left(\mathbf{u}\right)\right)\right)\right) - \mathbf{I}_v\left(\mathbf{u}\right).$$

This uses a predicted vertex map into the known previous frame, and a predicted RGB image in the same frame using opengl for rendering.

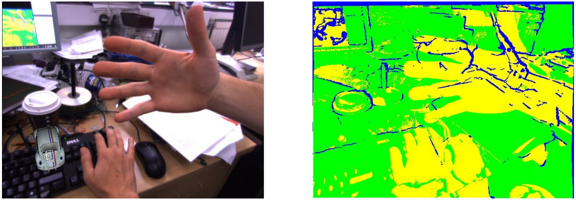


Figure: Gating given the predicted and live image (shown left).

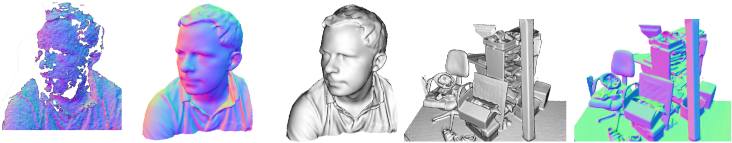
Tracking **without** binary data association

Increased **robustness** to fast motion over sparse SFM in PTAM tracking, and given the dense surface prediction we can do mixed and augmented reality.

Figure: Resilience to camera blur due to the massive redundancy obtained by using all the image data.

KinectFusion: Real-Time Dense Surface Mapping and Tracking

ISMAR 2011 work Shahram Izadi at MSRC. We exploit the massive redundancy in 30fps depth maps from the structured light based kinect device by fusing the data into a global implicit surface.



Tracking from the dense fused model

For the first time, we then track the current frame against the complete fused model massivley improving tracking ability with surprising results for global consistency. We use only depth data and the implementation is designed to exploit GPGPU.

Dense Mapping as Surface Reconstruction

- There are many techniques from computer vision and graphics for taking a noisy point cloud and turning it into a complete surface estimate.
- Representation is important, we don't want to be restricted in surface topology or precision.

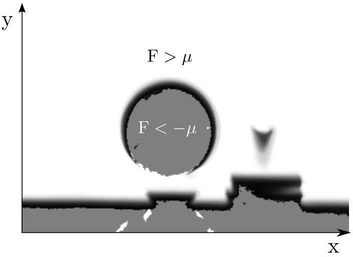
Use all data

We want to integrate over $640 \times 480 \times 30 \approx 9.2$ Million depth measurements per second on commodity hardware.

- Point clouds are *not* surfaces and meshes or parametric patches have problems with merging different topologies.

Signed Distance Function surface representations

We use a *truncated signed distance* function representation, $\mathrm{F}(\vec{x}) : \mathbb{R}^3 \mapsto \mathbb{R}$ for the estimated surface where $\mathrm{F}(\vec{x}) = 0$.



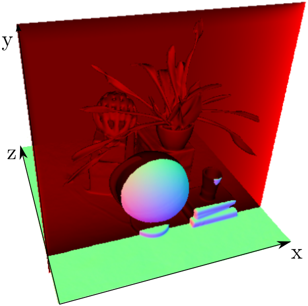
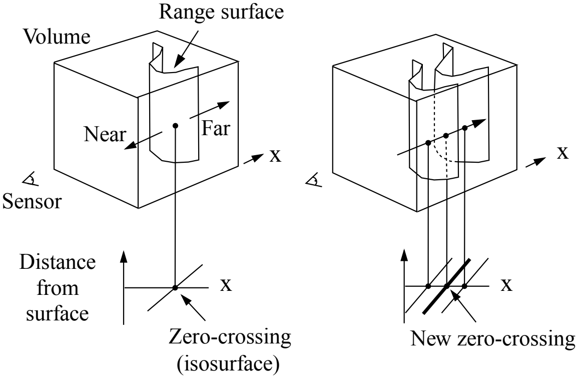


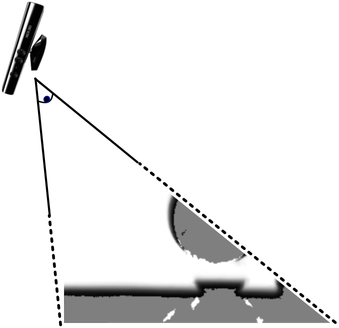
Figure: A cross section through a 3*D* Signed Distance Function of the surface shown.

Surface reconstruction via depth map fusion

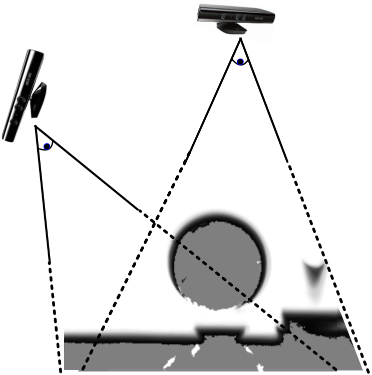
Curless and Levoy (1996) introduced very simple method for fusing depth maps into a global surface using the signed distance function representation.



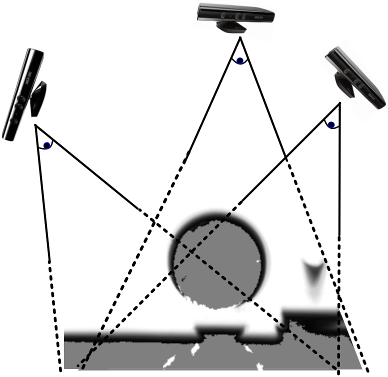
SDF Fusion



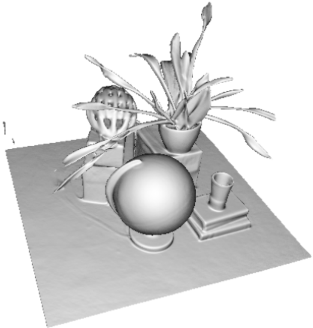
SDF Fusion



SDF Fusion

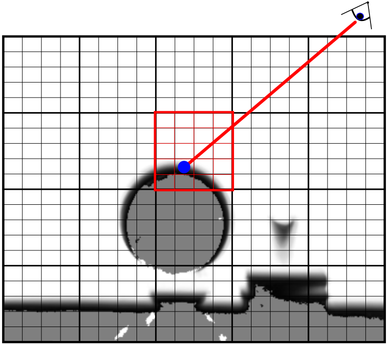


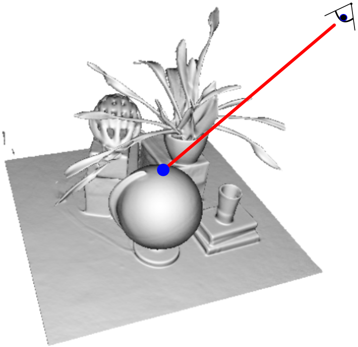
SDF Fusion



Similar to volumetric denoising of the SDF under an $\mathcal{L}_2$ norm data-cost with no regularisation: Can be computed online as data comes in using weighted average.

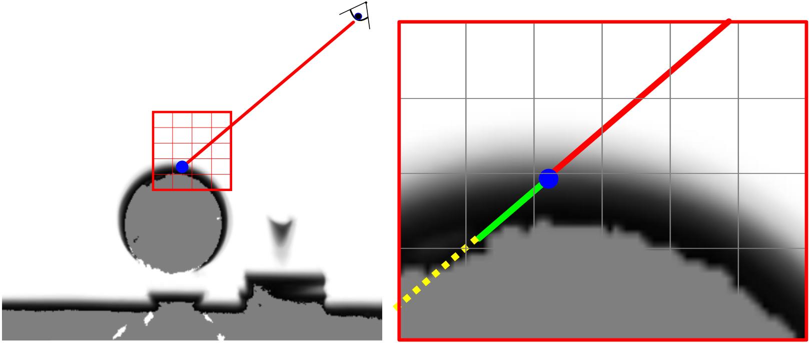
Rendering a surface represented in SDF





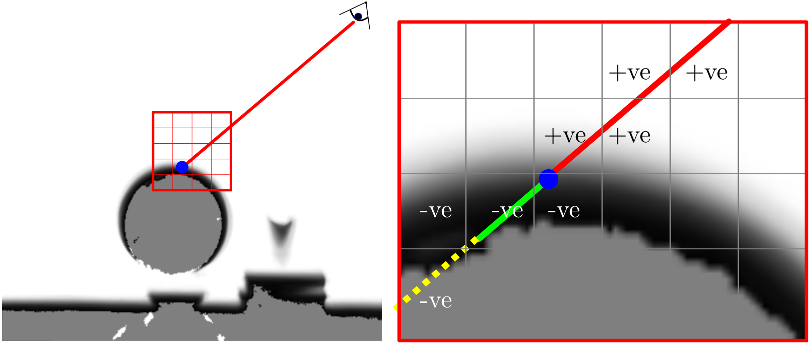
A regular grid holds a discretistion of the SDF. Ray-casting of iso-surfaces (S. Parker et al. 1998) is an established technique in graphics.

Rendering a surface represented in SDF



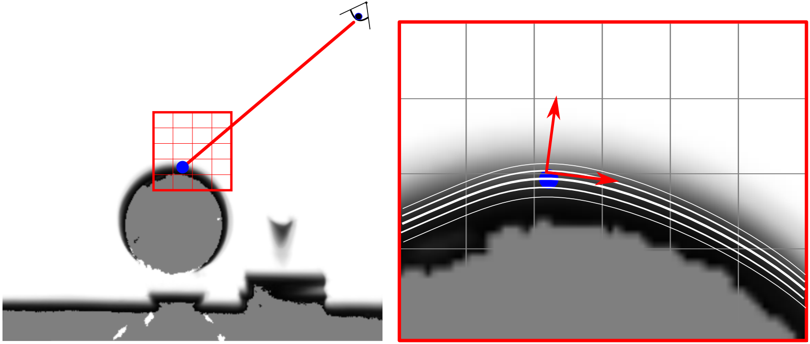
A regular grid holds a discretistion of the SDF. Ray-casting of iso-surfaces S. (Parker et al. 1998) is an established technique in graphics.

Rendering a surface represented in SDF



Interpolation reduces quantisation artefacts, and we can use the SDF value in a given voxel to skip along the ray if we are far from a surface.

Rendering a surface represented in SDF



Near the level sets near the zero crossing are parallel. The SDF field implicitly represents the surface normal.

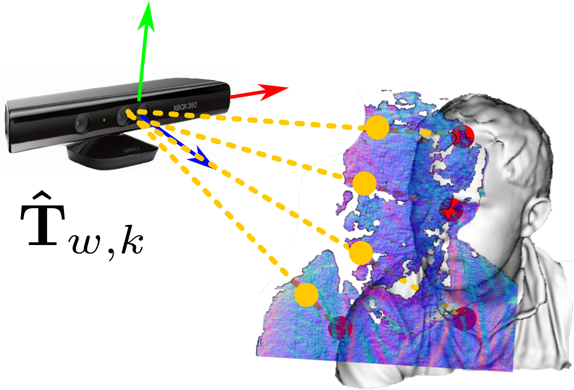
Tracking as Depth Map to Dense surface alignment

- Use all available depth data.
- Using only depth data, we can use Iterated Closest Point (ICP) based surface alignment introduced by P. Besl and N. McKay (1992).

Surface Alignment Outline

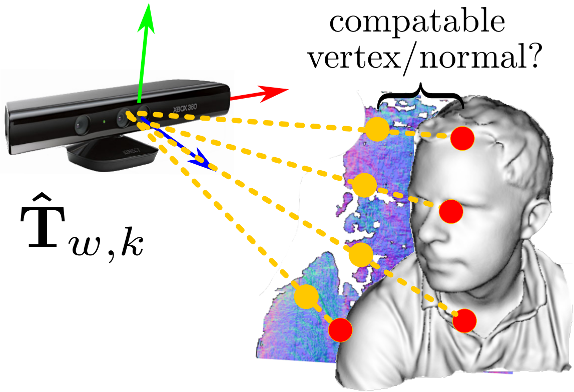
1. Obtain correspondences between a surface measurement and the surface model. We use projective data-association (G. Blais and M. D. Levine. 1995) to obtain fast dense correspondences.
2. Find the transform for the surface measurement that minimises the surface-model correspondence distance (we use the point-plane metric by Y. Chen and G. Medioni, 1992).

Projective Data Association

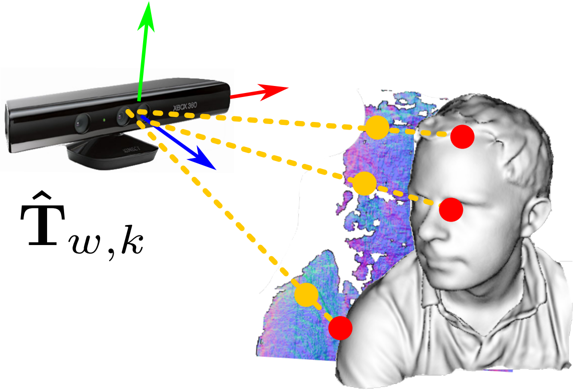


$$\hat{\mathrm{T}}_{w,k}$$

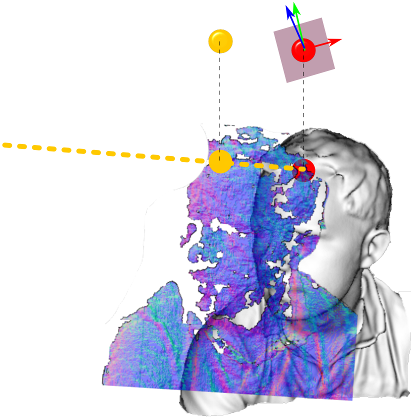
Projective Data Association



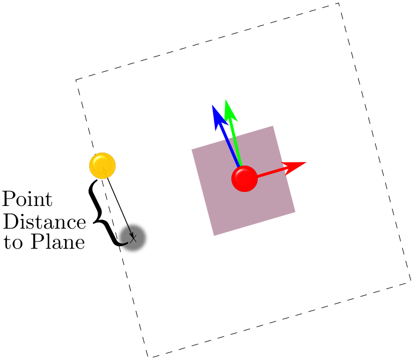
Projective Data Association



Point Plane Metric

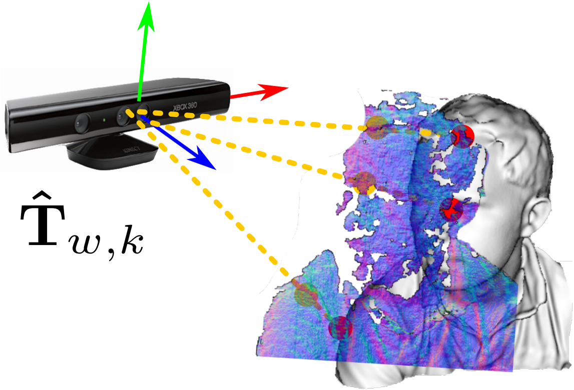


Point Plane Metric

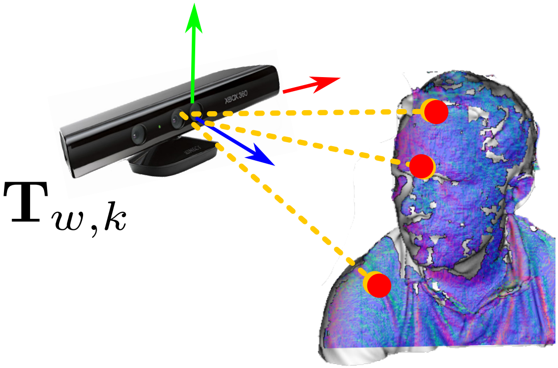


Point-plane metric allows surfaces to *slide* over each other and compliments the projective data-association method.

Minimising the point plane error



Minimising the point plane error



Useful properties

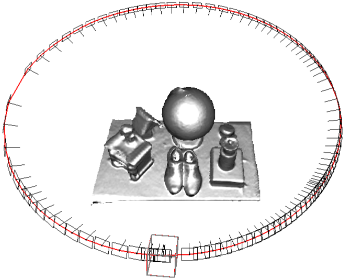
We performed a number of experiments to investigate useful properties of the system.

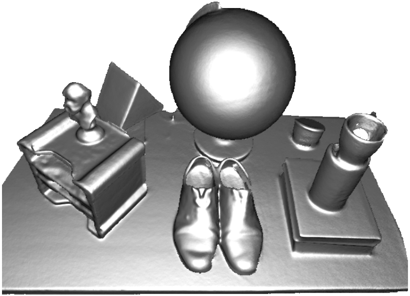
- Drift free tracking
- Scalable dense tracking and mapping
- Joint tracking/mapping convergence

Frame-Frame vs. *Frame-Model* Tracking

Low Drift Tracking with KinectFusion

Frame-Model tracking provides drift free, higher accuracy tracking than Frame-Frame (Scan matching).

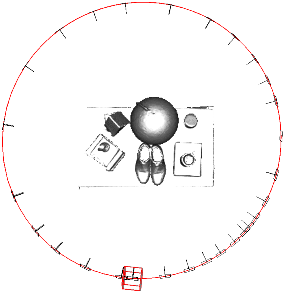


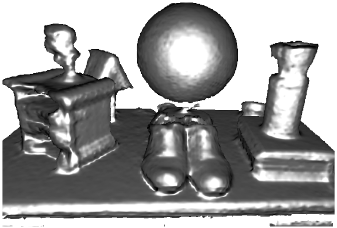


Scalability

Scalability and Robustness

System scales elegantly for limited hardware: frame dropping and reduction in voxel resolution: example $1/64^{th}$ memory and keeping every 6^{th} frame.

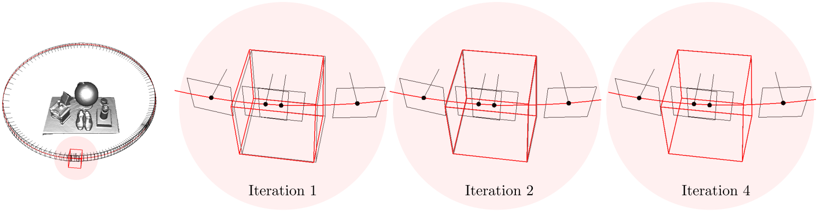




Alternating Joint optimisation

Geometry/Tracking Convergence

Joint Convergence without explicit joint optimisation. To a minimum of point plane and joint reconstruction error (although the point of convergence may not be the global minimum).

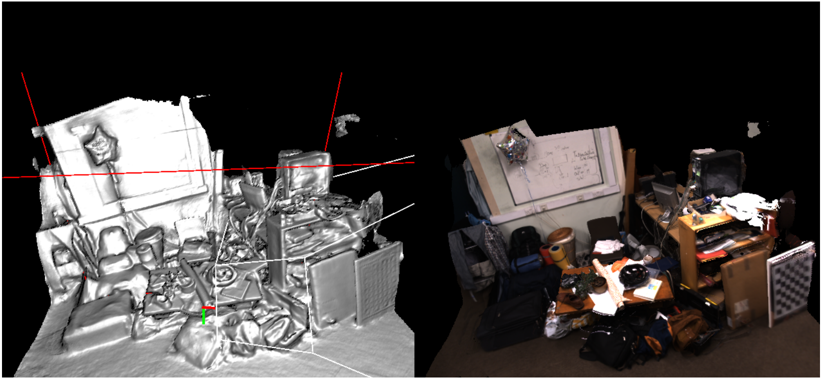


Real-time Surface Fusion using a Single RGB Camera

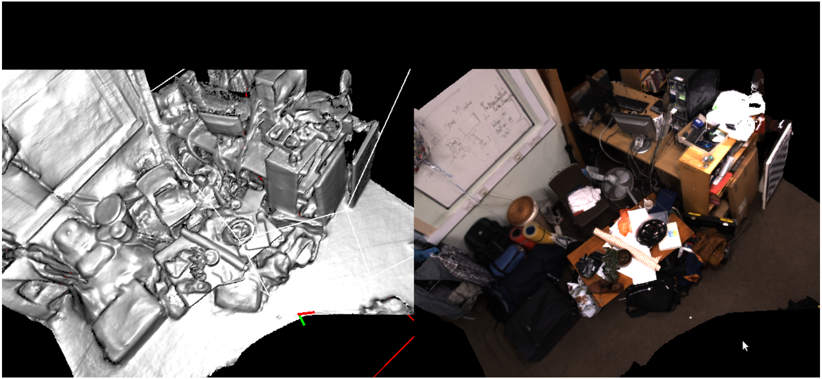
The KinectFusion work allowed us to investigate a simple surface representation, and given we have abstracted the data coming from the camera as depth images in DTAM, we can apply the same technique to obtain excellent near real-time results using a single passive camera.

Using the SDF with passive MVS data

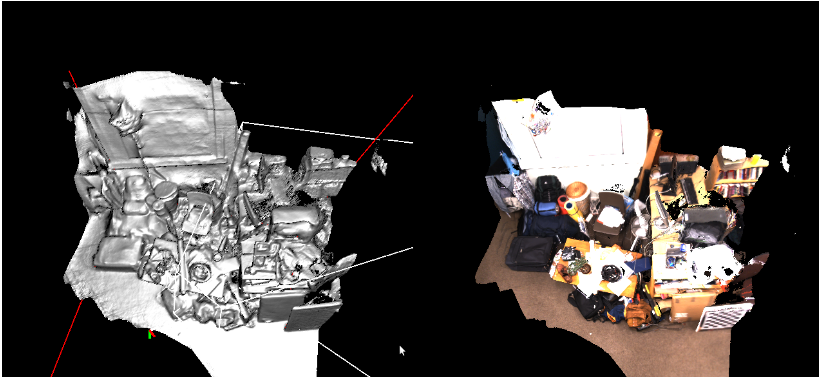
We are now investigating whether work by Pock and Zach (2008) on Globally optimal range fusion using a $\mathbf{TV}\text{-}\mathcal{L}_1$ based energy with a raw stereo data based SDF can be improved by first optimising the depth maps in the inverse depth formulation prior to fusion using the simple weighted $\mathbf{L}_2$ norm based reconstruction.



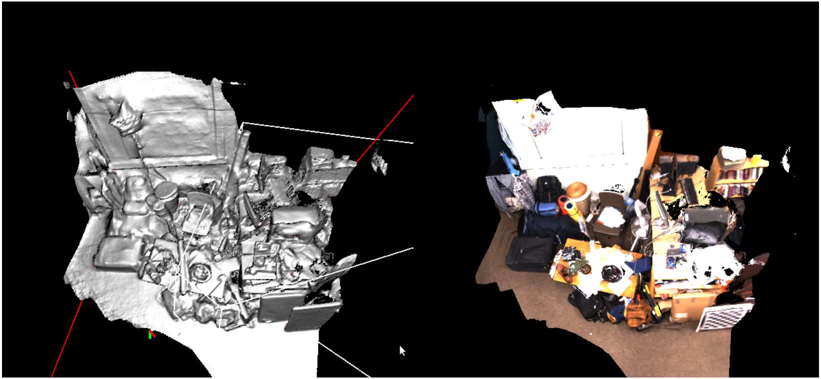
Real-time Surface Fusion using a Single RGB Camera



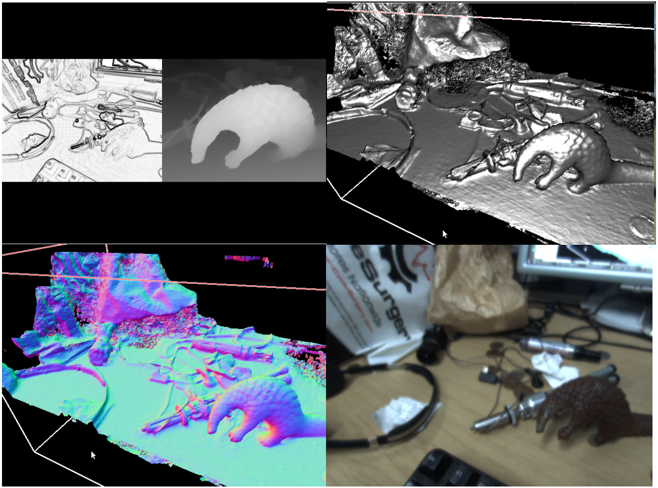
Real-time Surface Fusion using a Single RGB Camera



Real-time Surface Fusion using a Single RGB Camera



Real-time Surface Fusion using a Single RGB Camera



Evaluating live dense reconstruction and tracking systems

We can obtain ground truth for a given **trajectory** observing a dense **surface** is possible using robot arms and laser scanning, but posibly a *different* approach is required to the usual offline MVS datasets.

- Motion Blur and image degradation is normal in real-time imagery
- Systems that provide real-time feedback on performance and guiding robot/user may well take a different trajectory to obtain optimal performance.

Investigating Real-world performance

- City of Sights by Lukas Gruber TU Graz provides an interesting example of what is possible with rapid prototyping, in this case with paper models, but could be via $3D$ printing technology.

Discussions on LDR and Dense VSLAM Approaches

- Storing the non-parametric surface efficiently – exploiting sparsity and redundancy of SDF surfaces.
- Loop closure and drift correcton for Dense Surface SLAM systems is a novel issue.
- Inferring more of the physical scene using a single moving camera.

Thanks and Questions?

I thank my supervisor Andrew Davison and colleagues Steven Lovegrove, Ankur Handa and the Robot Vision lab for collaborations and useful discussions, and many of the speakers here today for their guidence in this interesting topic!