

CSE 576 *KinectFusion:* Real-Time Dense Surface Mapping and Tracking

Richard A. Newcombe (RSE and GRAIL labs)

PhD. work from Imperial College, London

Microsoft Research, Cambridge

May 6, 2013



Table of Contents

- 1 Why we're interested in Real-Time tracking and mapping
- 2 The Kinect Revolution! (Or how commodity depth cameras have changed things)
- 3 Kinect Fusion System Overview
- 4 Real-time Surface Mapping
- 5 Real-time Surface Mapping
- 6 Experimental Results



Outline

- 1 Why we're interested in Real-Time tracking and mapping
- 2 The Kinect Revolution! (Or how commodity depth cameras have changed things)
- 3 Kinect Fusion System Overview
- 4 Real-time Surface Mapping
- 5 Real-time Surface Mapping
- 6 Experimental Results



Infrastructure free camera tracking and mapping the geometry of the world

The problem and research interest:

Given only a set of images captured by one or more cameras taken in a static scene, can you:

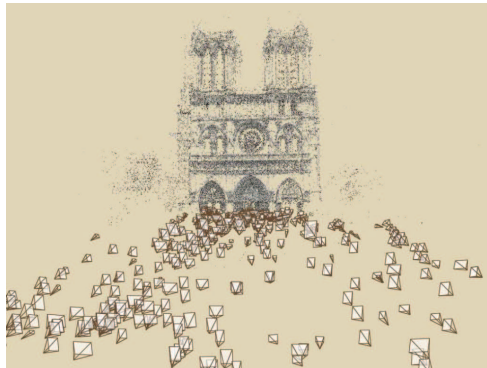
- 1 Estimate the relative poses of the camera for the different frames?
- 2 Estimate the geometry observed by the cameras?

- Image measurements can be augmented with other locally acquired measurements (inertial measurements).
- Cameras can range from passive RGB to depth sensing devices
- Infrastructure based alternatives require costly installation of satellites for pose estimation, and require contact based geometry estimation.



Example Offline Pipeline Applications

- We want to estimate the geometry of a scene and a locations of our cameras in the scene.



[Noah Snavely, Steven M. Seitz, Richard Szelisk Sigrahp 2006]

Dense Real-time SLAM Motivation

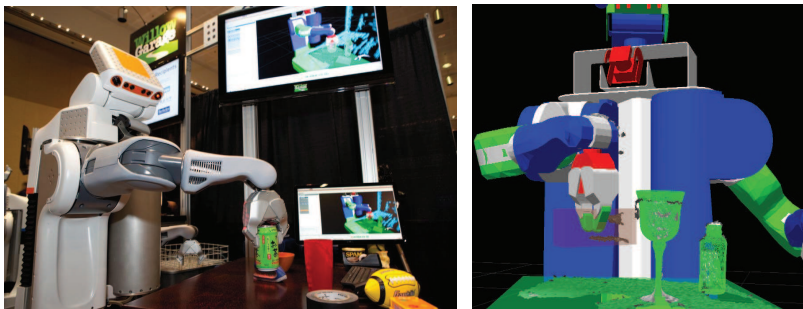
What are the new applications that become available when structure and motion are estimated in real-time?

What are the new applications that become available when the real-time structure estimation is a dense surface reconstruction?

What are the useful constraints that become available when assuming the input data is video rate from a moving body?

Dense SLAM Motivation 1: Robotics

Model base robotics paradigm requires up to date model of its surrounding surfaces and estimation of the robot motion if it is to competently interact with it or avoid collisions during navigation.



Dense SLAM Motivation 1: Human-Computer Interaction

Immersive mixed and Augmented Reality **requires high accuracy sensor pose** and surface geometry estimation.



Motivation 2: Physically constrained vision

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).



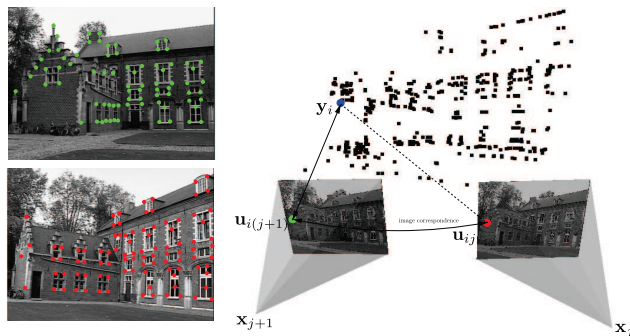
[Grabner, Ga, Van Gool "What makes a chair a chair?", CVPR 2011]

What if you can track a camera in real-time and you have modelled the world, can that help in object recognition?

Joint Camera Pose and Point Geometry Estimation

-Online (Structure from Motion)

-Obtain image correspondences across N views

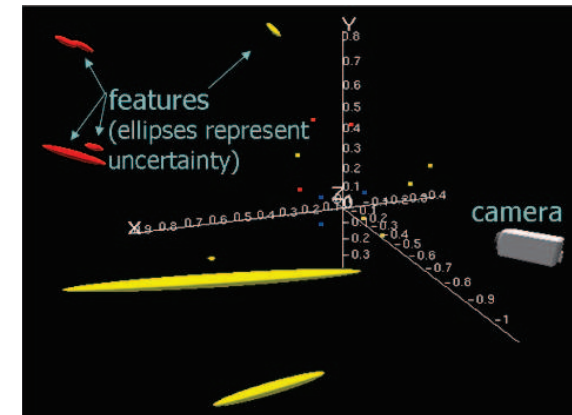


[Adapted from Pollefeys et al. 1999]

-Estimate 3D points y and camera poses x from image correspondences u
-Solve by minimising non-linear **2D Point Reprojection Error**

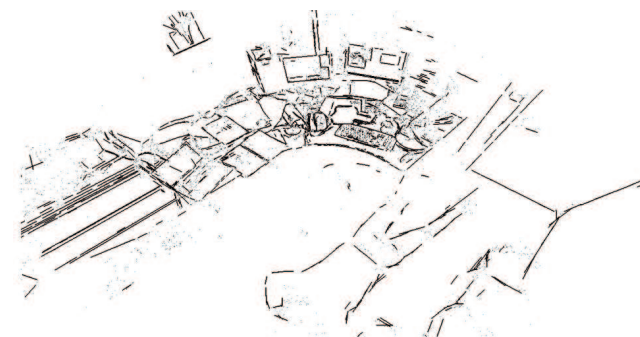
Real time, commodity SLAM system evolution: MonoSLAM (2003)

2003 Davison's Monoslam: importance of a cheap commodity sensor



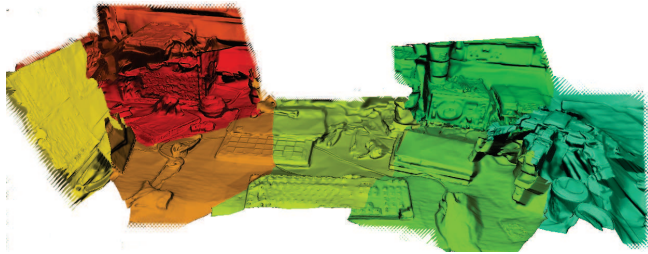
Real time, commodity SLAM system evolution: PTAM (2007)

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: LDR (2010)

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.



Research Live *dense* reconstruction from a passive camera is gathering pace (see upcoming Workshop at ICCV this year). However, passive methods will always fail when light levels are too low.



Table of Contents

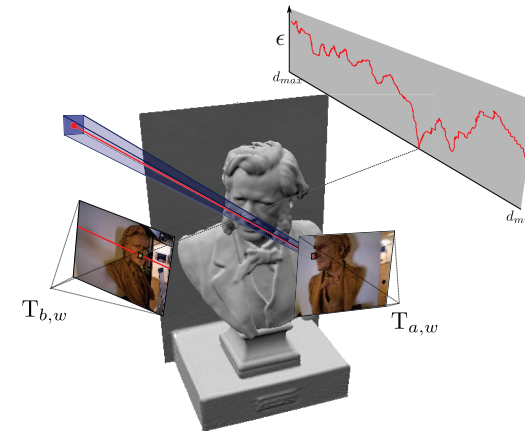
- 1 Why we're interested in Real-Time tracking and mapping
- 2 The Kinect Revolution! (Or how commodity depth cameras have changed things)
- 3 Kinect Fusion System Overview
- 4 Real-time Surface Mapping
- 5 Real-time Surface Mapping
- 6 Experimental Results



LDR Augments Sparse Mapping with Dense MVS

Multiple View Stereo

- A Reference pixel induces a photo-consistency error function
- Correspondence exists along epipolar line (*If...*)



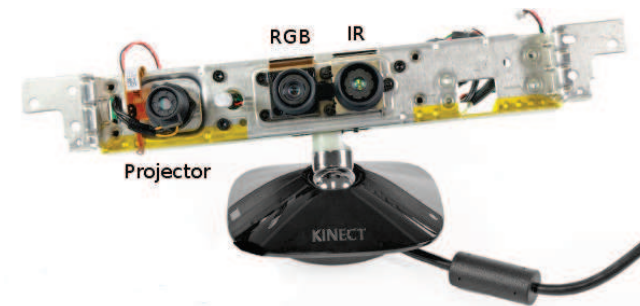
- Computationally Expensive



Key Technology (1): Commodity Stereo Estimation Hardware

Depth cameras have become commodity devices:

- Producing a 640×480 depth image at 30fps with dedicated ASIC
- Uses *single camera* projected light Stereo technique

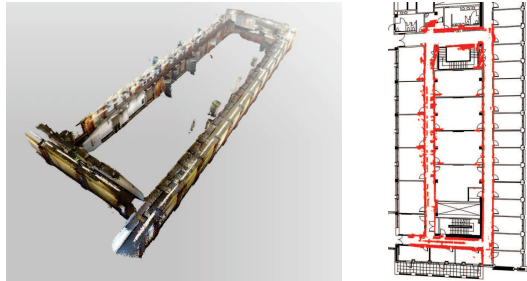


How it works: For each IR pixel ASIC selects depth using correlation over known speckled light pattern calibrated for 2^{11} depth planes.



Real time, commodity SLAM system evolution: RGB-D Mapping (2010)

2010 Peter Henry et al. (UW, CSE) demonstrated the first full commodity depth camera pipeline.



- RGBD-Mapping combines a sparse feature mapping system with dense RGB-D frame-frame alignment.
- The Dense depth data is however not fused into a full model in real-time but through an off-line *surfel* reconstruction step
- The system has limited dense reconstruction quality but is capable of large scale drift free mapping.

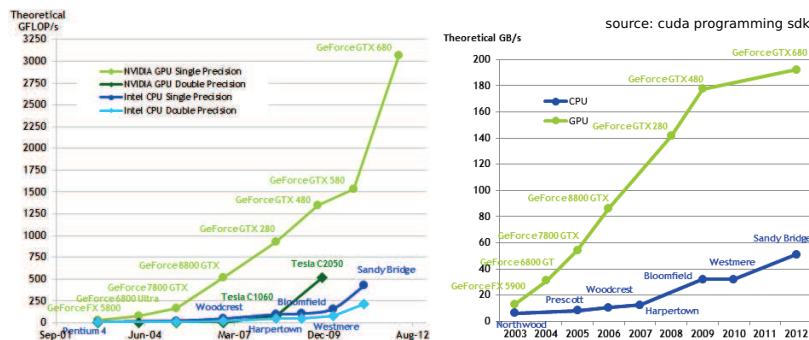
Key Technology (2): Powerful GPGPU processing



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
- However, with the right surface representation we can model and track millions of points per second.

Key Technology (2): Powerful GPGPU processing



- GPUs provide very high FLOP and memory throughput rates but only for specific, simplified processing and memory transfer patterns.
- GPUs are very good for trivially parallelizable operations with little branching in the code when operating over homogeneously accessible blocks of memory

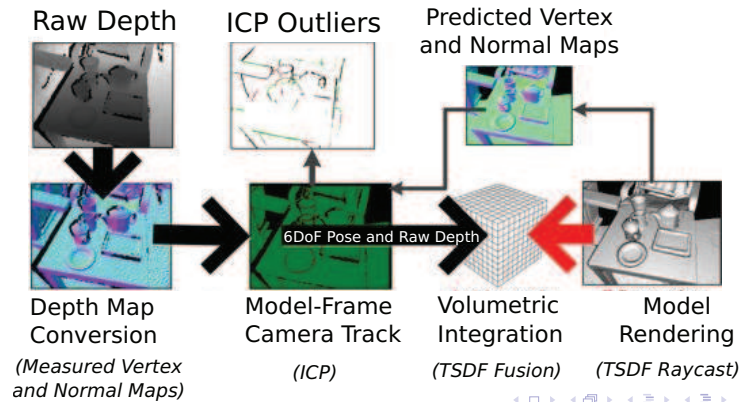
Table of Contents

- 1 Why we're interested in Real-Time tracking and mapping
- 2 The Kinect Revolution! (Or how commodity depth cameras have changed things)
- 3 Kinect Fusion System Overview
- 4 Real-time Surface Mapping
- 5 Real-time Surface Mapping
- 6 Experimental Results

What is KinectFusion?

Two *simple* interleaved components

- 1 Building a dense surface model from a set of depth frames with estimated camera poses.
- 2 Given a dense surface model, estimate the current camera pose by aligning the depth frame in the dense model.



Richard A. Newcombe (RSE and GRAIL labs)

CSE 576 KinectFusion: Real-Time Dense Surface Mapping and Tracking

Real time, commodity SLAM system evolution

- "KinectFusion: Real-Time Dense Surface Mapping and Tracking", Newcombe et al. 2011 ISMAR, the core system paper.
- Also, "KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera", Izadi et al. 2011 UIST, with extended applications of the core system.

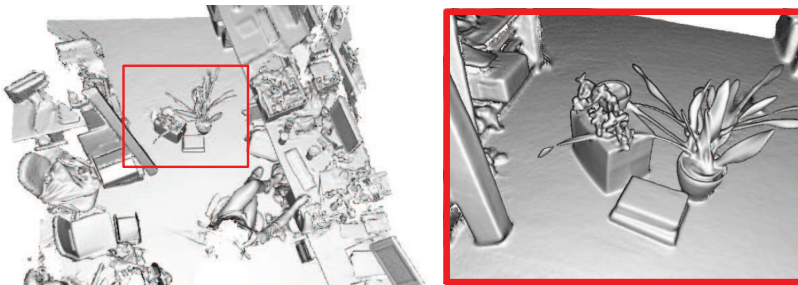


Richard A. Newcombe (RSE and GRAIL labs)

CSE 576 KinectFusion: Real-Time Dense Surface Mapping and Tracking

Real time, commodity SLAM system evolution

- Single commodity depth camera only system (Works in the dark)




Richard A. Newcombe (RSE and GRAIL labs)

CSE 576 KinectFusion: Real-Time Dense Surface Mapping and Tracking

Camera Motion: pose over time

For frame k the pose of the camera (this refers in this case to the infra-red sensor of the Kinect camera) is given by the six degree of freedom rigid body transform:



$$\mathbf{T}_{w,k} = \begin{bmatrix} \mathbf{R}_{w,k} & \mathbf{t}_{w,k} \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \text{SE}_3$$

$$\text{SE}_3 := \{\mathbf{R}, \mathbf{t} \mid \mathbf{R} \in \text{SO}_3, \mathbf{t} \in \mathbb{R}^3\}$$

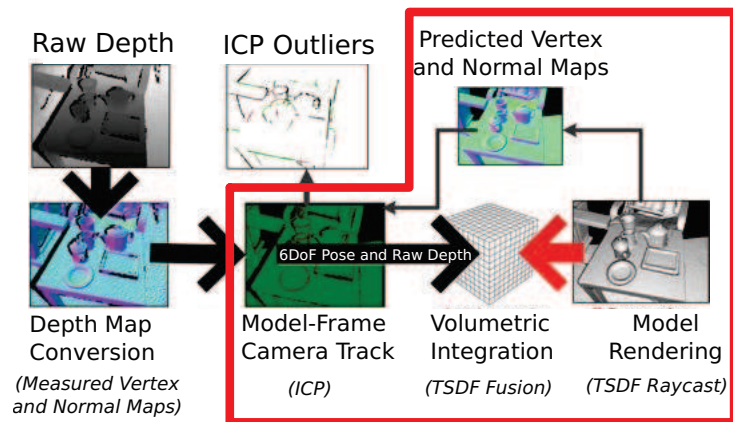
Depth map to Dense 3D surface measurement

We can transform any depth map from its local frame depth map into a global frame surface measurement.

Richard A. Newcombe (RSE and GRAIL labs)

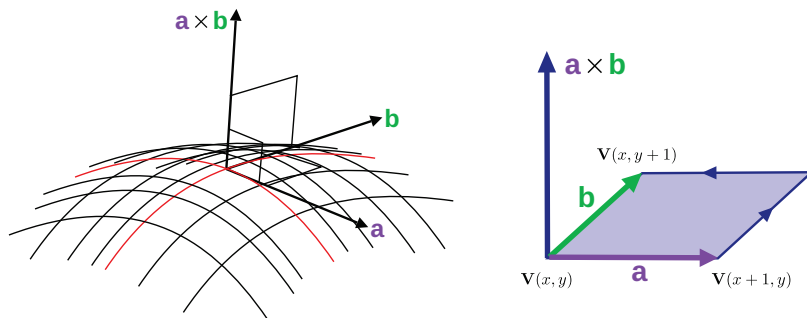
CSE 576 KinectFusion: Real-Time Dense Surface Mapping and Tracking

From Depth to a Dense Oriented Point Cloud



Approximating surface normals

- We can estimate the surface normal from neighbouring pairs of 3D points by exploiting the regular grid structure



- Unit normal: $\mathbf{N}(x, y) = \frac{\mathbf{a} \times \mathbf{b}}{\|\mathbf{a} \times \mathbf{b}\|}$

From Depth to a Dense Oriented Point Clouds

- Each valid depth map value $D(u)$ at pixel $u = (x, y)^T$ provides a 3D point measurement in camera frame:

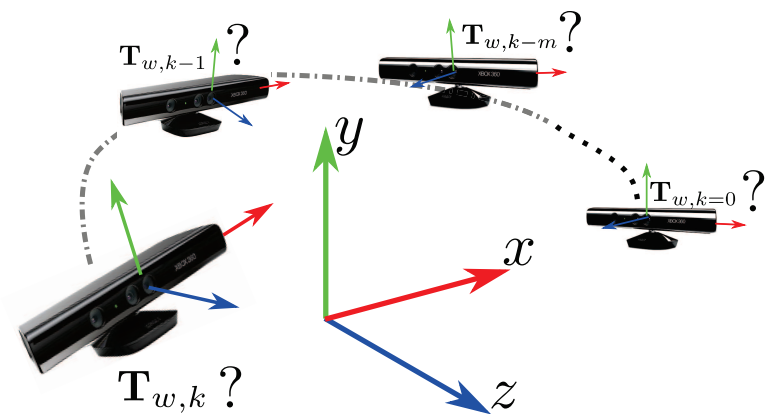
$$v = K^{-1} [x, y, 1]^T D(x, y)$$

- Requires known camera intrinsics, K . With focal length f_0, f_1 and principle point p_0, p_1 :

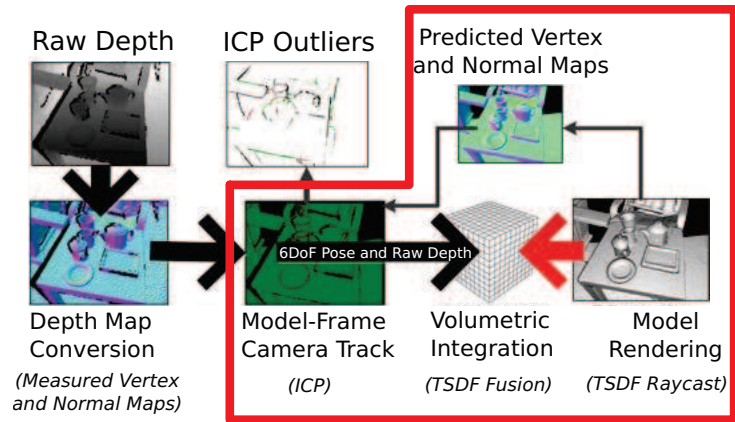
$$K \equiv \begin{pmatrix} f_0 & 0 & p_0 \\ 0 & f_1 & p_1 \\ 0 & 0 & 1 \end{pmatrix}$$

- The depth map at time k provides a *scale correct* 3D point measurement at each pixel; a vertex map \mathbf{V}_k .
- Transformation of the 3D point from the camera to world frame is $\mathbf{v}_w = \mathbf{T}_{w,k} \mathbf{v}_k$

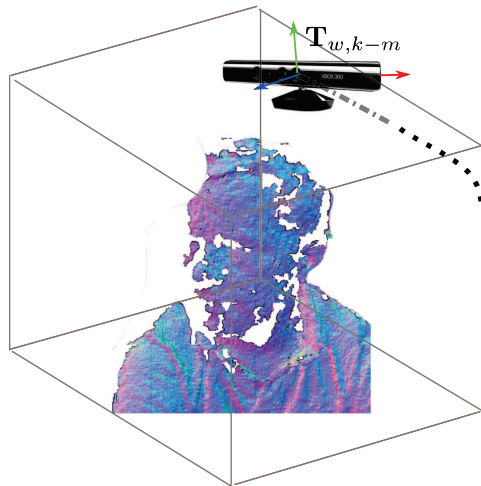
Joint Estimation Problem: What is the camera motion and surface geometry?



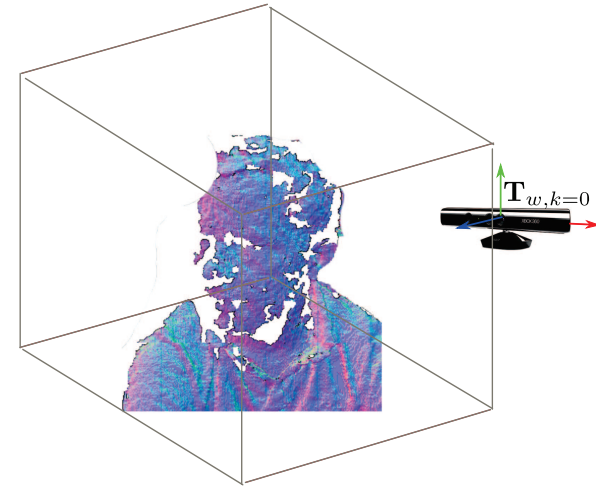
Joint Estimation Problem: What is the camera motion and surface geometry?



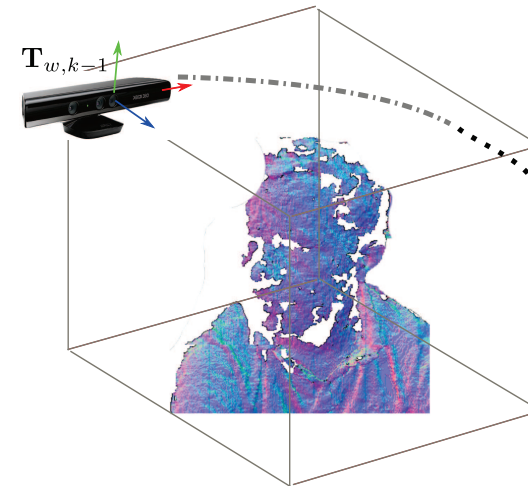
Knowing camera motion...



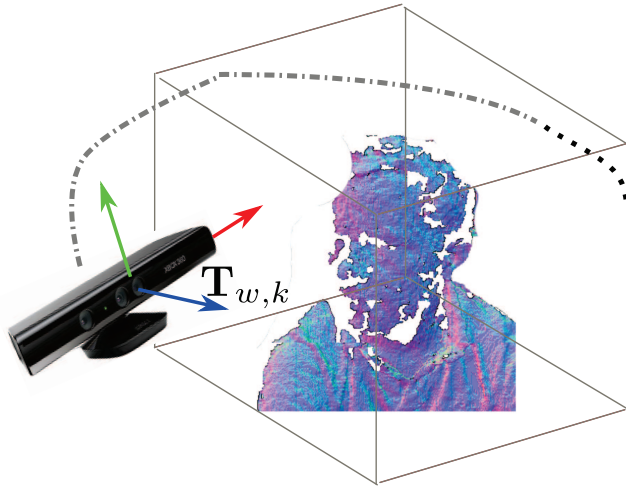
Knowing camera motion, enables model reconstruction...



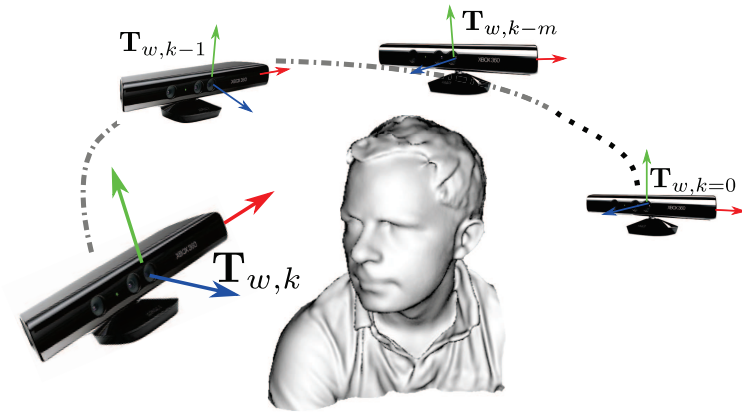
Knowing camera motion...



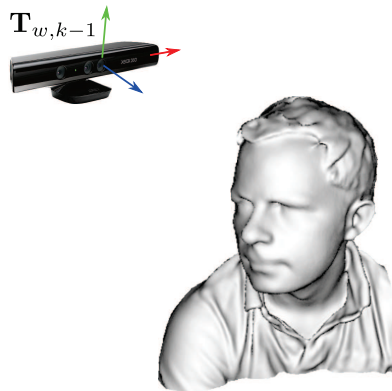
Knowing camera motion...



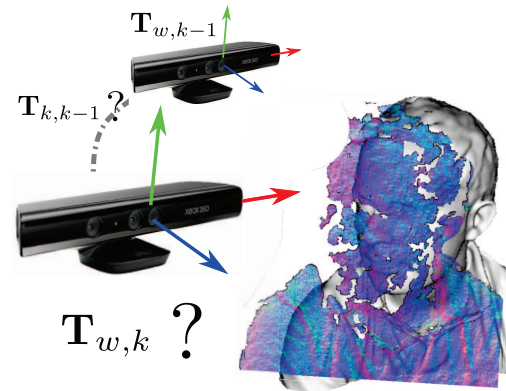
...enables measurement fusion (surface reconstruction)...



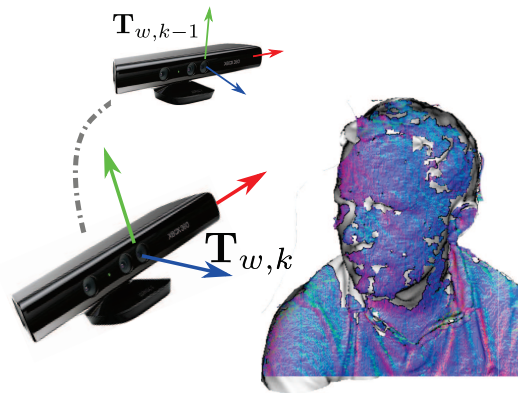
...also, given a known model...



...we can align a new surface measurement...



...minimising the predicted surface measurement error...



...giving us a best current pose estimate, enabling fusion.

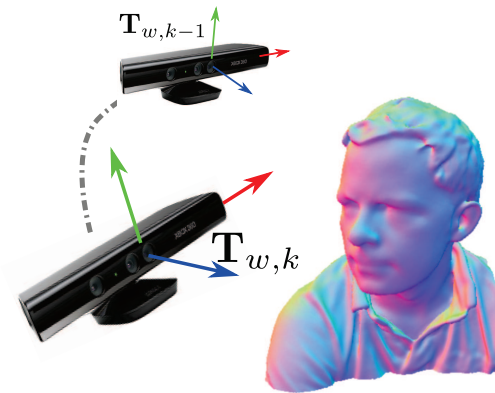


Table of Contents

- 1 Why we're interested in Real-Time tracking and mapping
- 2 The Kinect Revolution! (Or how commodity depth cameras have changed things)
- 3 Kinect Fusion System Overview
- 4 Real-time Surface Mapping
- 5 Real-time Surface Mapping
- 6 Experimental Results

Dense Mapping as Surface Reconstruction

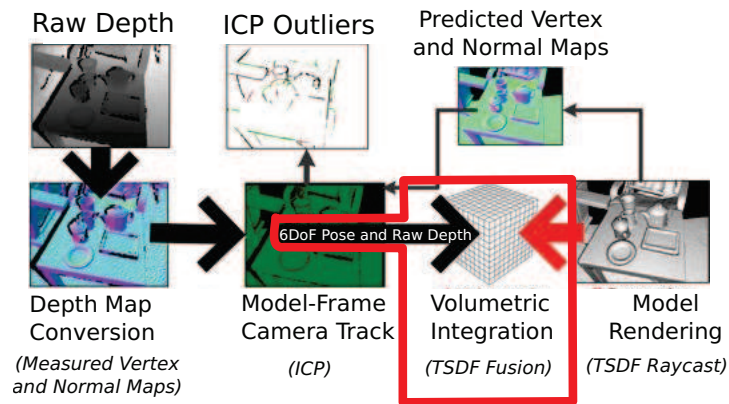
- There are many techniques from computer vision and graphics for taking a noisy point cloud and turning it into a complete surface estimate.
- Representation is important, we don't want to be restricted in surface topology or precision.
- We want to use all the data available.

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
- Changing topology is costly and complicated for parametric or explicit mesh representations.

Dense Mapping as Surface Reconstruction



Truncated Signed Distance Function surface representations

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

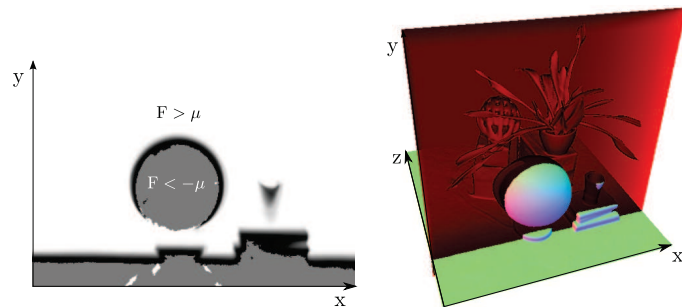
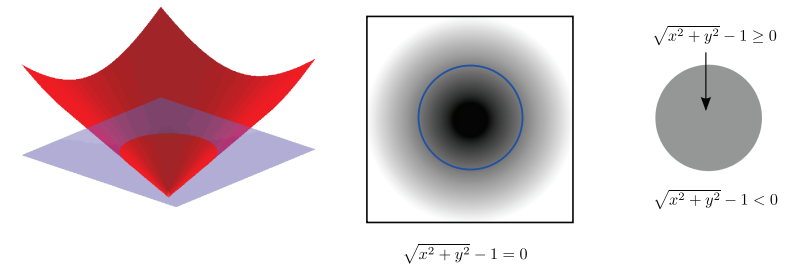


Figure: A cross section through a 3D Signed Distance Function of the surface shown.

Implicit Surface Representation

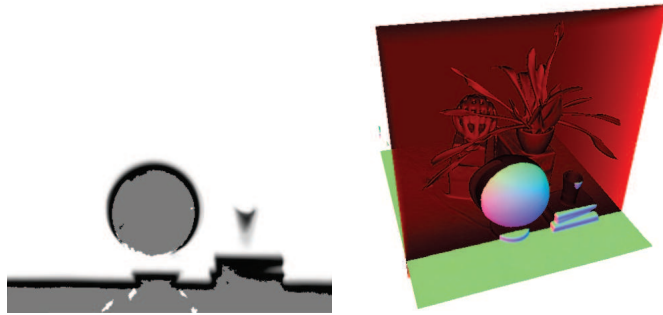
We can implicitly define the geometry as the level-set of a function:



Signed Distance Function surfaces

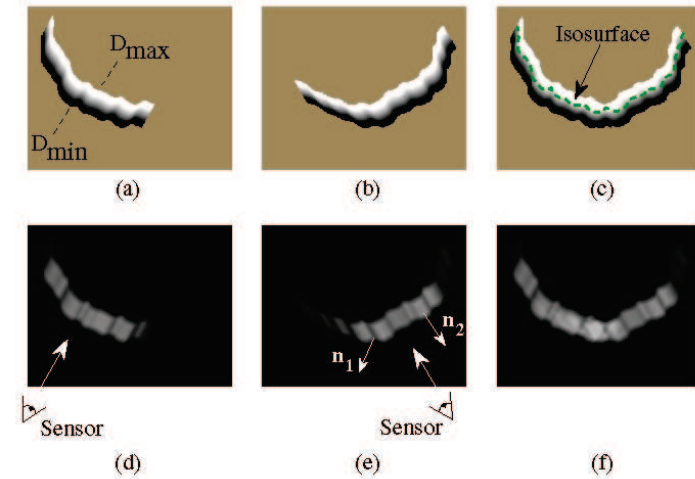


Signed Distance Function surfaces



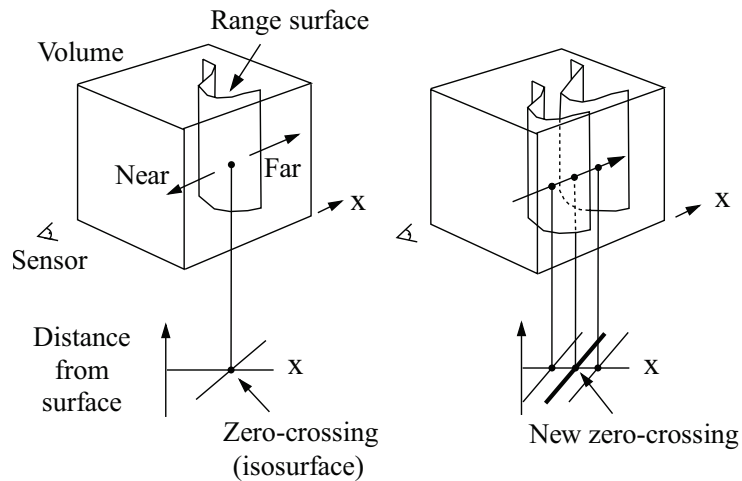
Surface reconstruction via depth map fusion

Curless and Levoy (1996) introduced signed distance function fusion.

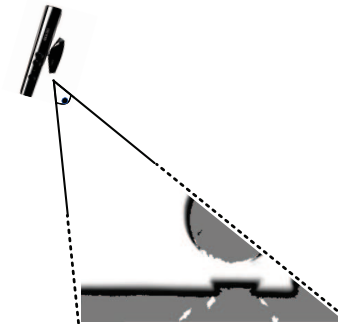


Surface reconstruction via depth map fusion

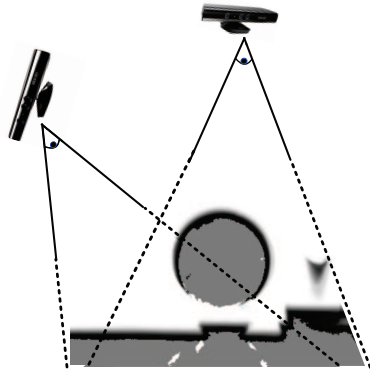
Curless and Levoy (1996) introduced signed distance function fusion.



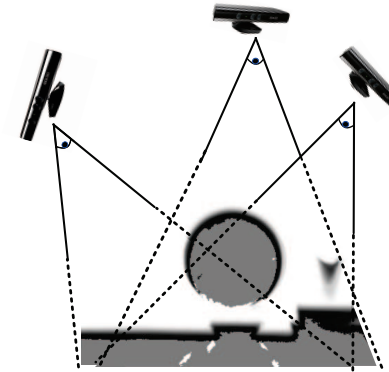
SDF Fusion



SDF Fusion



SDF Fusion



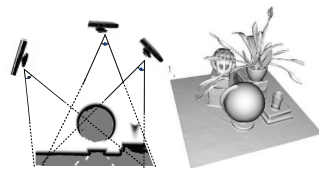
SDF Fusion

- Let S_k be the TSDF integrated upto frame k , with associated weight function W_k
- Recursive weighted average update rule given new TSDF measurement, s_{k+1} from frame $k + 1$ with weight function w_{k+1}

TSDF Fusion (Curless & Levoy (1996))

$$S_{k+1}(\mathbf{x} \in \Lambda) = \frac{W_i(\mathbf{x})S_k(\mathbf{x}) + w_{k+1}s_{k+1}(\mathbf{x})}{W_k(\mathbf{x}) + w_{k+1}(\mathbf{x})}$$

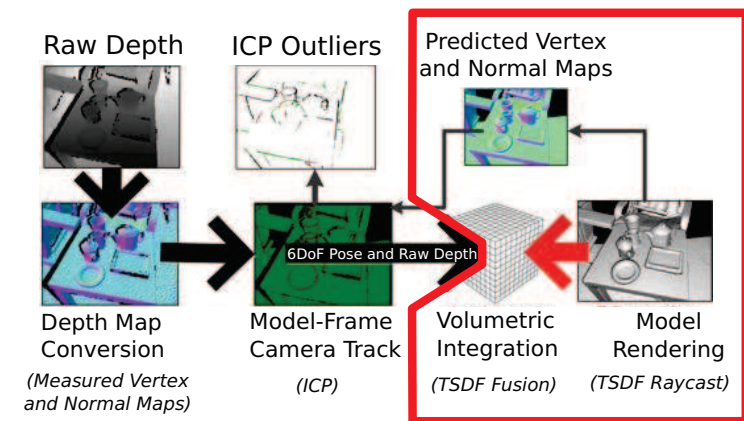
$$W_{k+1}(\mathbf{x}) = W_k(\mathbf{x}) + w_{k+1}(\mathbf{x})$$



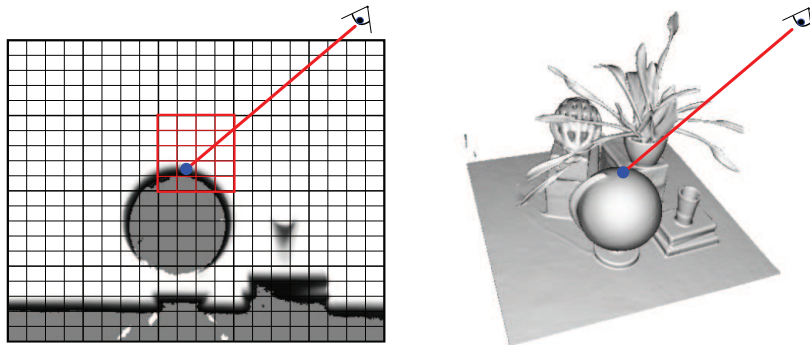
Equivalent to multiple volumetric denoising of the TSDF volume under a squared penalty data-cost with no regularisation.



Dense Mapping as Surface Reconstruction

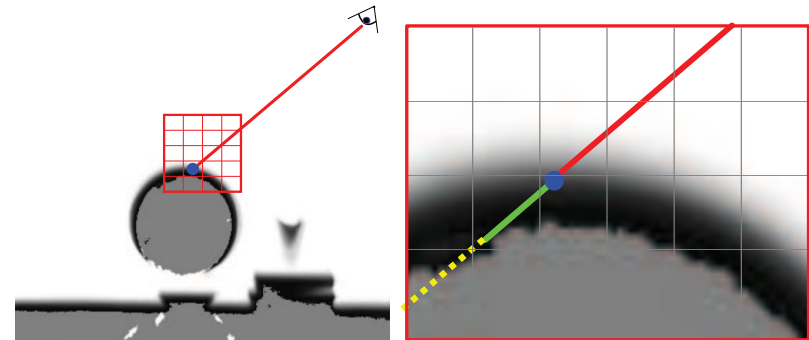


Rendering a surface represented in SDF



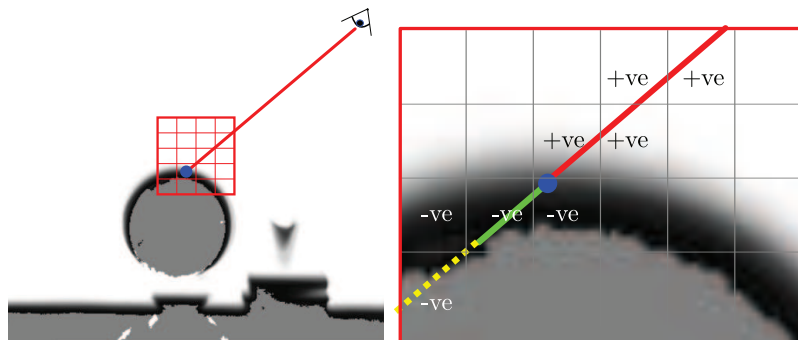
A regular grid holds a discretisation of the SDF. Ray-casting of iso-surfaces (S. Parker et al. 1998) is an established technique in graphics.

Rendering a surface represented in SDF



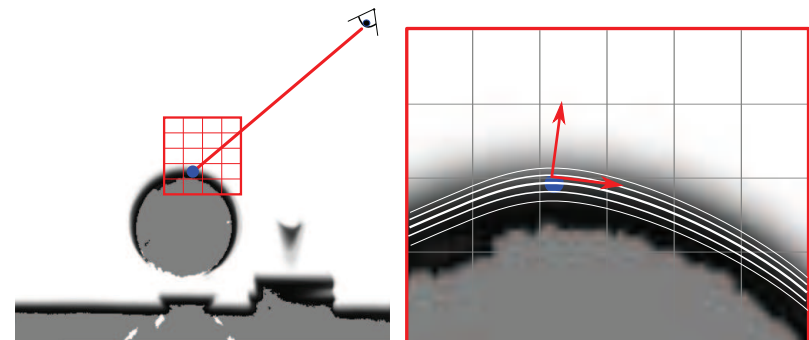
A regular grid holds a discretisation of the SDF. Ray-casting of iso-surfaces S. (Parker et al. 1998) is an established technique in graphics.

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.

Dense Mapping as Surface Reconstruction

Dense Mapping Algorithm

Given depth map R_k and pose $\mathbf{T}_{k,w}$, For each voxel \mathbf{p} within frustum of frame k update the Truncated Signed Distance function:

- 1 Project voxel into frame k : $\mathbf{x} = \pi(\mathbf{K}\mathbf{T}_{k,w}\mathbf{p})$
- 2 Compute signed distance between $\lambda^{-1}\|\mathbf{p} - \mathbf{t}_{w,k}\|$ and depth for this pixel $D_k(\mathbf{x})$
- 3 Truncate the signed distance.
- 4 Update the weighted average TSDF value for this voxel.

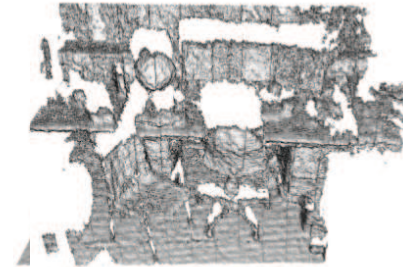
Using this approach we can integrate over $640 \times 480 \times 30 \approx 9.2$ Million depth measurements per second on high end laptop grade GPGPU.



TSDF Fusion



TSDF Fusion



TSDF Fusion



Moving Average TSDF Fusion

What happens if:

- We replace the **full averaging** with a **moving average** TSDF?
- i.e. limit the maximum of the weight function

$$S_{k+1}(\mathbf{x} \in \Lambda) = \frac{W_i(\mathbf{x})S_k(\mathbf{x}) + w_{k+1}s_{k+1}(\mathbf{x})}{W_k(\mathbf{x}) + w_{k+1}(\mathbf{x})}$$

$$W_{k+1}(\mathbf{x}) = \min(\tau, W_k(\mathbf{x}) + w_{k+1}(\mathbf{x}))$$



Tracking as Depth Map to Dense surface alignment

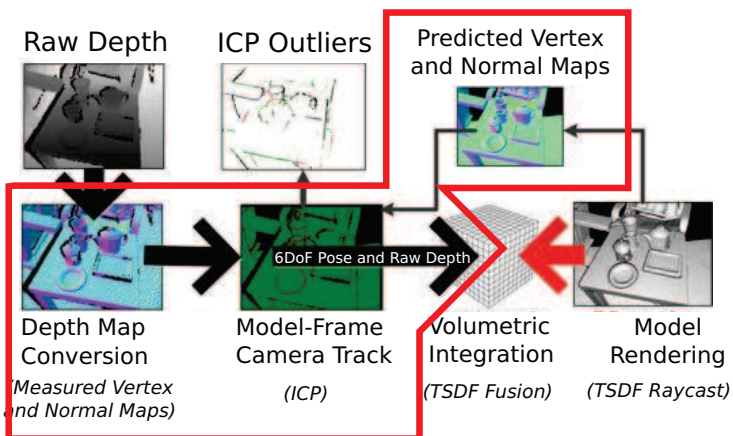


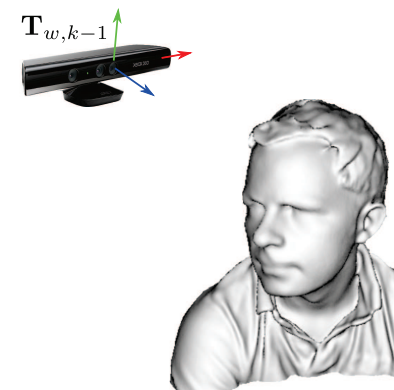
Table of Contents

- 1 Why we're interested in Real-Time tracking and mapping
- 2 The Kinect Revolution! (Or how commodity depth cameras have changed things)
- 3 Kinect Fusion System Overview
- 4 Real-time Surface Mapping
- 5 Real-time Surface Mapping
- 6 Experimental Results



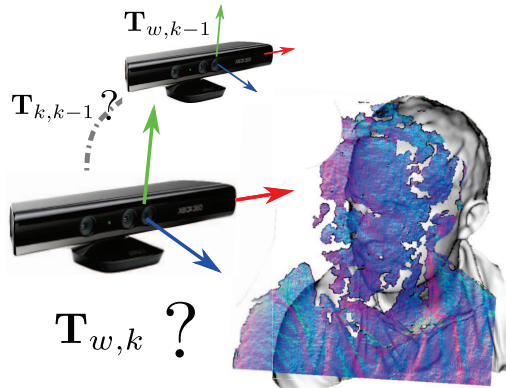
Tracking as Depth Map to Dense surface alignment

Given a known (partially completed) surface model...



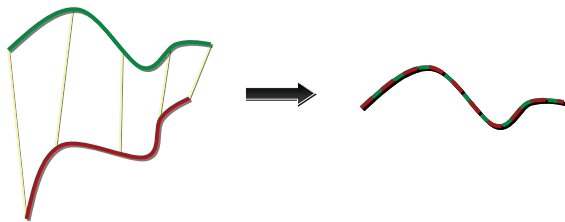
Tracking as Depth Map to Dense surface alignment

...We want to estimate a new camera frame 6DoF pose...



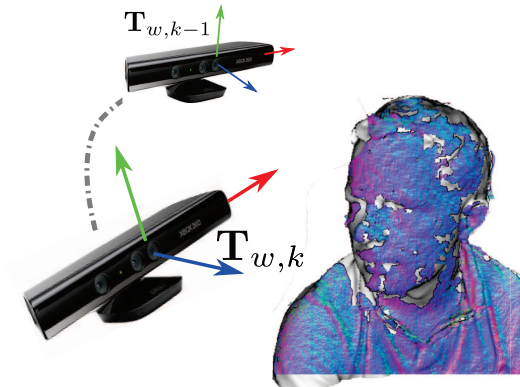
Aligning 3D data

- If we had explicit correspondences we can minimise the distance metric
- We could use a feature detection-extraction-matching pipeline



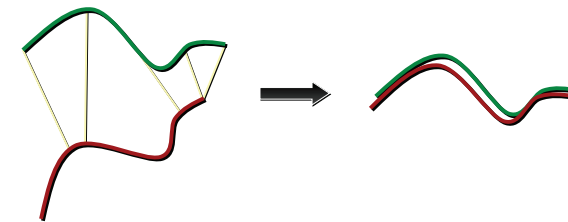
Tracking as Depth Map to Dense surface alignment

...which is equivalent to finding the pose that aligns the depth map data onto the current model...



Aligning 3D data using Iterative Closest Points (ICP)

- If instead we assume closest points *are* tentative correspondences
- We could use implicit correspondence and minimise the distance metric...



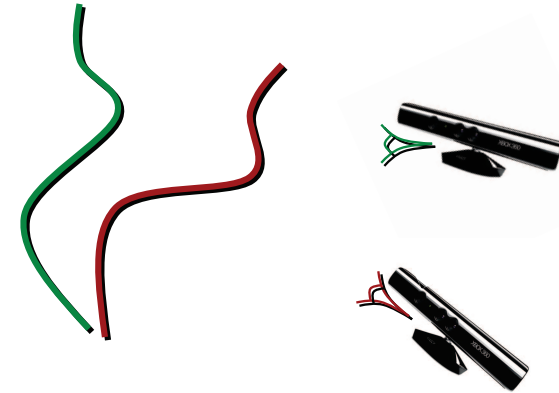
Aligning 3D data using Iterative Closest Points (ICP)

- ...and iterate (Iterative closest points), (Besl & McKay 1992).
- When can we expect ICP to converge?



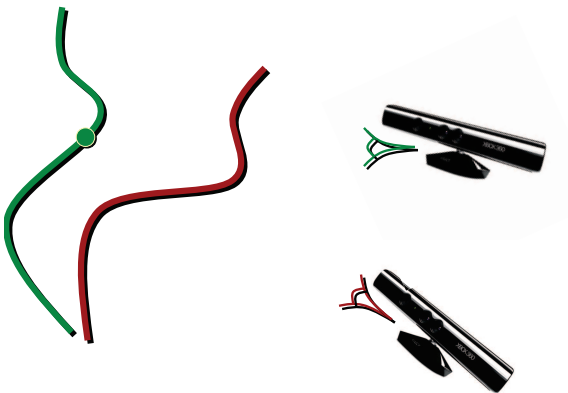
Projective Data Association Idea

- Projective data-association (G. Blais and M. D. Levine. 1995) to obtain fast dense correspondences using closest point approximation
- Does not require expensive computation of true closest points



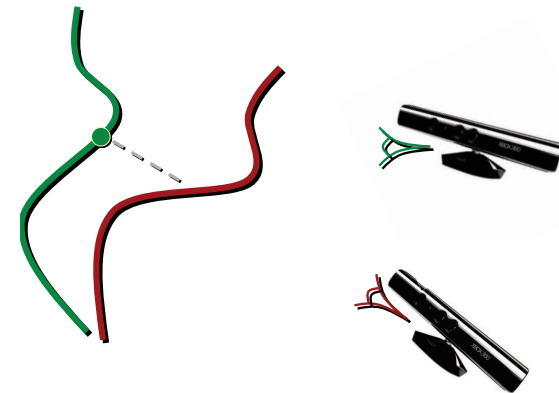
Projective Data Association Idea

- Select a point from the reference surface



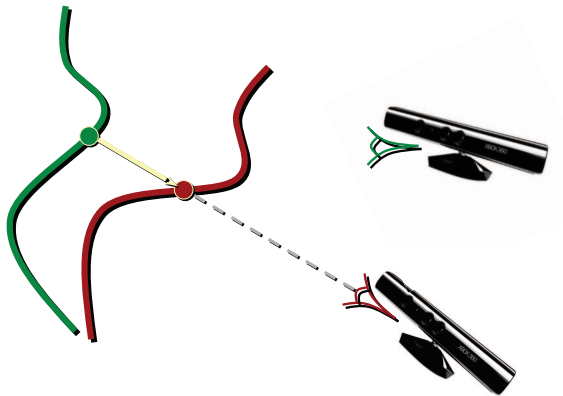
Projective Data Association Idea

- Project the into the frame of the second surface measurement



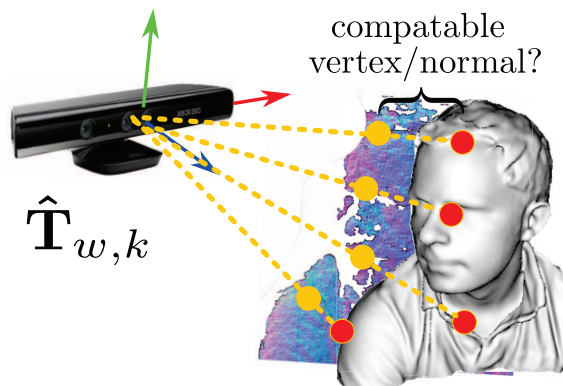
Projective Data Association Idea

- Project the into the frame of the second surface measurement



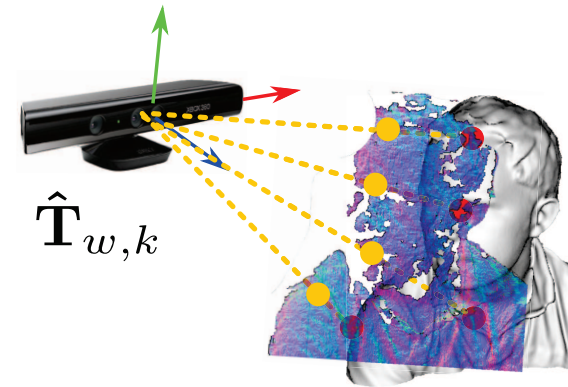
Dense Projective Data Association

- It is useful to use other measures of compatibility between points:
- Accept match *iff*: surface normals are similar and if point distance is not too large.



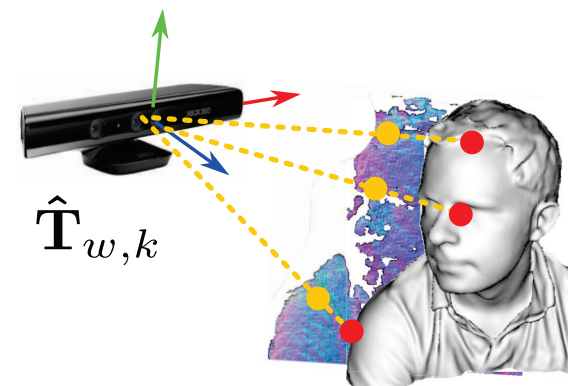
Dense Projective Data Association

- In KinectFusion we *predict* a depth map by raycasting the current surface model given the estimated camera frame
- Initialise the camera frame estimate with the previous frame pose



Dense Projective Data Association

- It is useful to use other measures of compatibility between points:
- Accept match *iff*: surface normals are similar and if point distance is not too large.



Example Data Association Outliers

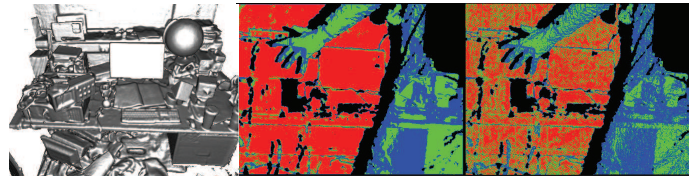


Figure: ICP compatibility testing on the current surface model (Left). with bilateral filtering on the vertex/normal map measurement (Middle), using raw vertex/normal map (Right).

Point-Plane ICP

- Given data-association between a model and a live depth frame
- We want to estimate the 6DoF transform that aligns the surfaces

Point-Plane Distance metric (Y. Chen and G. Medioni, 1992)

Point-plane error for a given transform $T_{w,k} \in \mathbb{SE}(3)$, over all associated points:

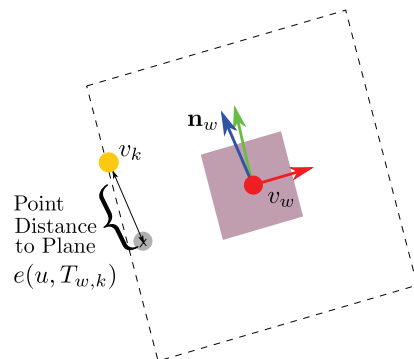
$$E_c(T_{w,k}) = \sum_{u \in \Omega} \psi(e(u, T_{w,k})) ,$$
$$e(u, T_{w,k}) = \mathbf{n}_w(u')^\top (T_{w,k} v_k(u) - v_w(u')) .$$

- Vertex $v_k(u)$ in pixel u is data-associated with the global model predicted vertex $v_w(u')$ at pixel u' with normal \mathbf{n}_w
- $\psi(\cdot)$ is a penalty function, typically chosen as the squared distance function or can be chosen as a robust estimator function.

Point-Plane ICP

Point-Plane Distance metric (Y. Chen and G. Medioni, 1992)

$$e(u, T_{w,k}) = \mathbf{n}_w(u')^\top (T_{w,k} v_k(u) - v_w(u'))$$



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

Fast ICP Algorithm (Rusinkiewicz & Levoy 2001)

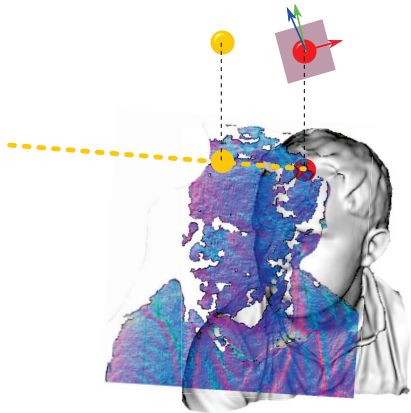
The combination of both projective data-association and the point-plane metric is called **Fast ICP**:

Point-Plane ICP with Projective Data Association

Repeat until convergence:

- 1 Given current point correspondences, minimise $E_c(T_{w,k})$ (Gauss-Newton based gradient descent)
 - 2 Use new estimate of $T_{w,k}$ and update correspondences (projective data-association)
- **Fast ICP** developed by Rusinkiewicz & Levoy (SIGGRAPH 2001) enabled first real-time infrastructure free 3D scanning.

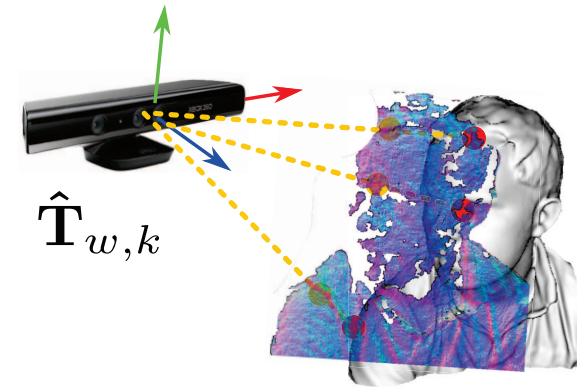
Point Plane Metric



Navigation icons: back, forward, search, etc.

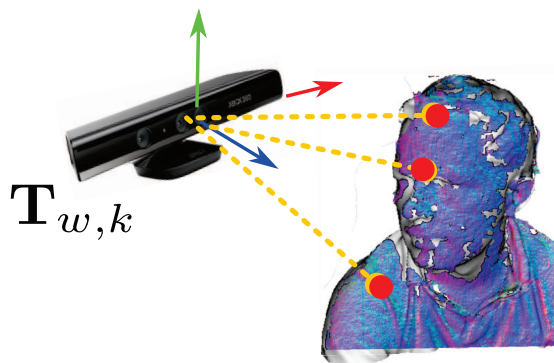
Minimising the point plane error

Minimising the point plane error



Navigation icons: back, forward, search, etc.

Table of Contents



Navigation icons: back, forward, search, etc.

- 1 Why we're interested in Real-Time tracking and mapping
- 2 The Kinect Revolution! (Or how commodity depth cameras have changed things)
- 3 Kinect Fusion System Overview
- 4 Real-time Surface Mapping
- 5 Real-time Surface Mapping
- 6 Experimental Results

Navigation icons: back, forward, search, etc.

Useful properties

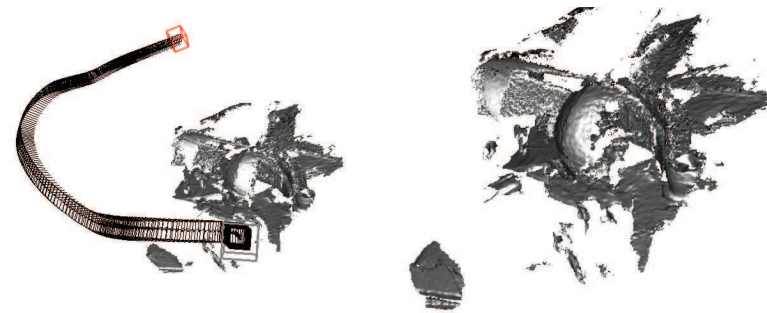
We performed a number of experiments to investigate useful properties of the system.

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



Frame-Frame vs. Frame-Model Tracking

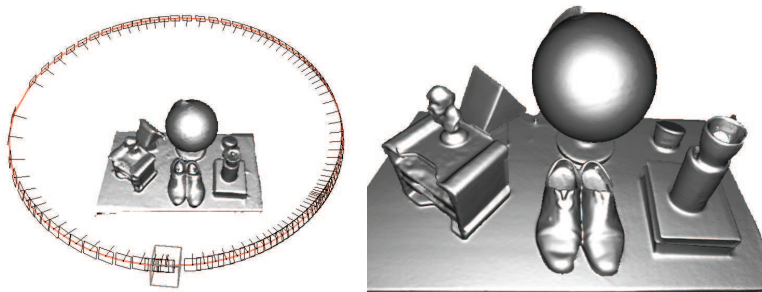
Frame-Frame tracking results in drift as pose errors are continuous integrated into the next frame.



Frame-Frame vs. *Frame-Model* Tracking

Drift Free Tracking with KinectFusion

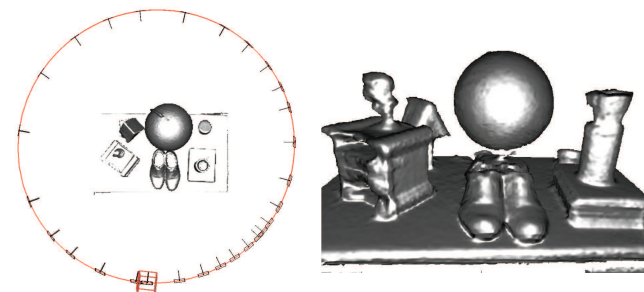
Frame-Model tracking provides drift free, higher accuracy tracking than Frame-Frame (Scan matching).



Scalability

Scalability and Robustness

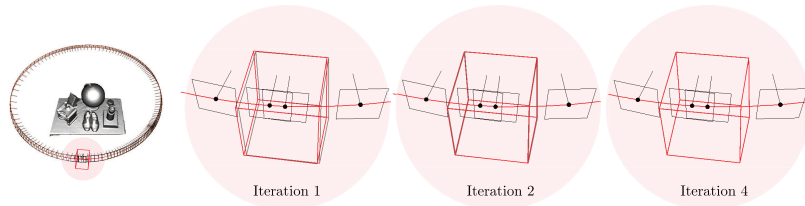
System scales elegantly for limited hardware: frame dropping and reduction in voxel resolution: example 1/64th memory and keeping every 6th frame.



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).



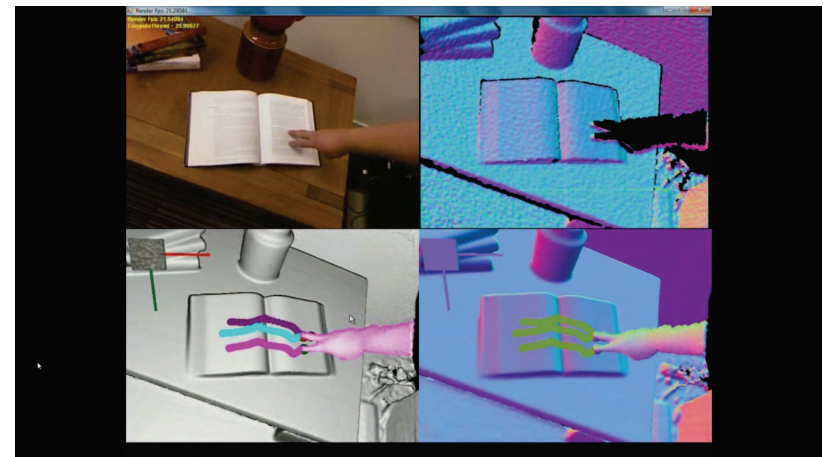
Issues

- Drift is still possible for long exploratory loops as there is no explicit loop closure.
- Sufficient surface geometry required to lock down all degrees of freedom in the point-plane system, e.g. Viewing a single plane leaves 3DOF nullspace.
- Regular grid discretisation of the SDF does not scale for larger spaces. Instead there is a lot of sparsity in the volume that we can exploit using octree style SDF.
- Extensions that solve these problems are becoming available while maintaining real-time capabilities.

Demonstration

Demonstration

A new AR/MR Platform?



Questions?

Demonstration/Questions?

References

- See updated web slides for complete references.