SLAM++: Simultaneous Localisation and Mapping at the Level of Objects

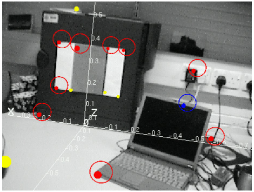
Renato F. Salas-Moreno[1], Richard A. Newcombe[2], Hauke Strasdat[1], Paul H. J. Kelly[1] and Andrew J. Davison[1]

Department of Computing,
Imperial College London[1]
&
CSE, University of Washington[2]

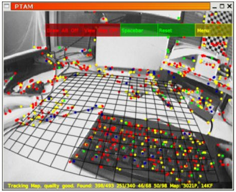
June 28, 2014

Outline

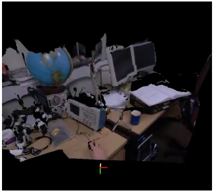
Real-time Dense SLAM



MonoSLAM, 2003



PTAM, 2007



DTAM, 2011

KinectFusion, 2011

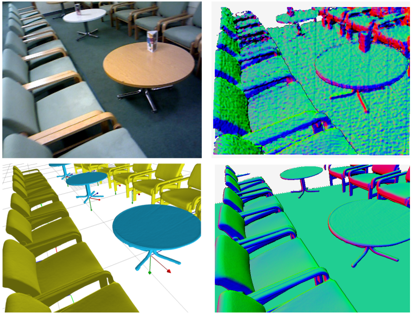
3D Representation is Evolving

- Increase in density of representation benefiting from increasing processing and memory resources.
- But also increasingly taking account of real-word priors that the world is not a point cloud.

To recap

- Maps initially represented by keypoints.
- Density increased towards full dense surfaces.
- World still meaningless: e.g. Keypoint or surface patch on a keyboard same as in monitor as far as interaction is concerned due to limited perception.

SLAM++: A Pure Object-Level SLAM System



CVPR 2013.

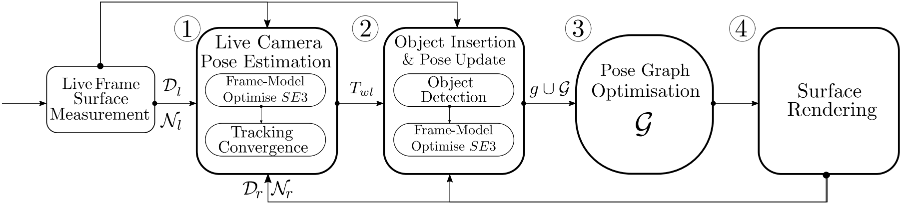
- Brings semantic understading in the loop of SLAM itself.
- SLAM++: SLAM at the level of objects, Salas-Moreno, Newcombe, Strasdat, Kelly, Davison, CVPR 2013.

Mapping Directly at the Object Level

By recognizing objects in the loop, we have advantages:

- Instant semantic understanding enables interaction (robotics, AR).
- Dense representation and full predictive power of KinectFusion for robust tracking but very memory-efficient and scalable.
- Enables the inclusion of domain-level knowledge to deal with incorrect mapping.

System Overview



Real-Time Loop

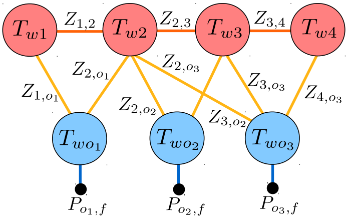
- Main components of the system. Mostly on GPU.
- Begin with live measurement from sensor.
- Objects are detected and placed into a map.
- Map used for camera tracking (localisation).
- World represented by objects as nodes in a graph.
- Map is rendered and compared against new measurements to update pose.

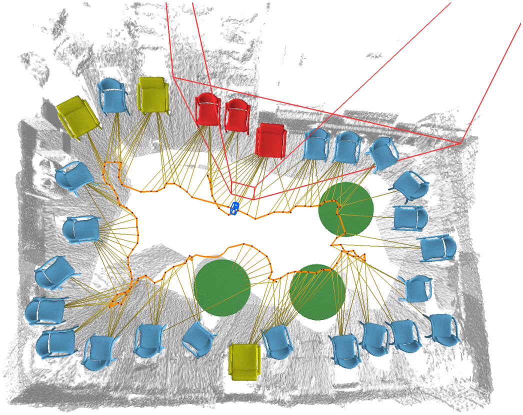
Object Database

Currently a small number of precisely known objects

- Carefully scanned using KinectFusion and manually segmented.

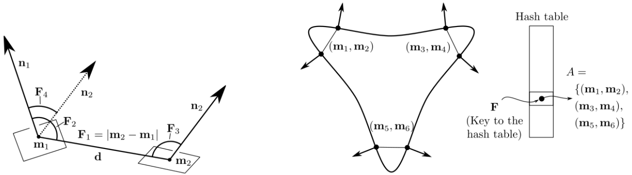
Map Representation





A Pure Graph of Objects

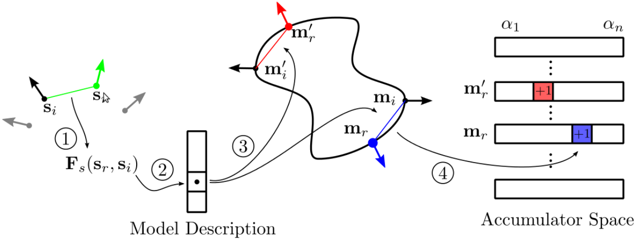
- Large room that has been mapped with objects.
- Camera trajectory in orange and measurements in yellow.



From 'Model Globally, Match Locally', Drost *et al.*, CVPR 2010

- Each object is described by a combination of all point pair features, describing the relative angles and distance between points.
- Object features are discretised and grouped into a Hash Table and used as a lookup for runtime queries coming from depth map features.

Real-Time Object Detection



From 'Model Globally, Match Locally', Drost *et al.*, CVPR 2010

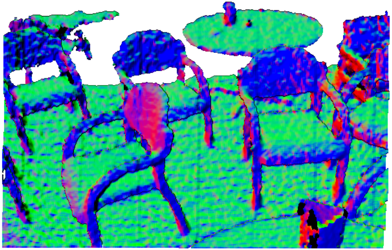
- Feature matches cast votes in accumulator for object pose. The pose with max votes assumed to be a rough estimate of true object pose.
- We have a new efficient parallel implementation on GPU/CPU.
- Still doesn't scale well to a large number of objects.

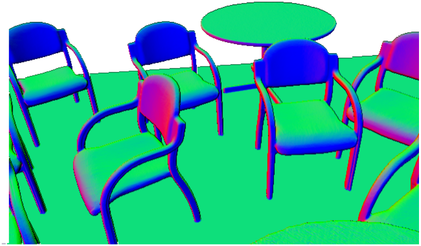
ICP Refinement

Noisy raw detected poses refined with ICP against object model

- Detections with the previous step are only approximations due to feature discretisation.
- Pose is refined with Iterative Closest Point that minimise a point plane error metric.

Camera Tracking

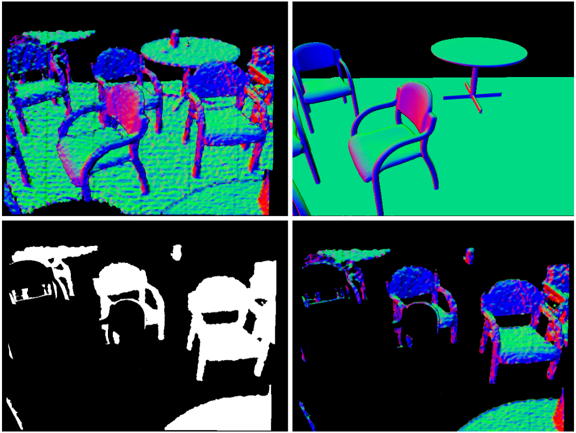




We use ICP again, this time to localise the camera

- On the left is the live noisy measurement from the camera.
- On the right is the clean rendered world.
- Assuming the world is static, a misalignment between the Measurement and our Prediction has to be due to camera motion.
- Minimising the alignment error equals to camera pose update.

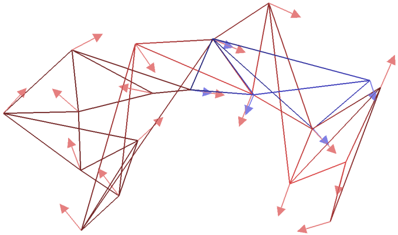
Active Object Search in the SLAM Loop



Mask-out already mapped areas

- Detection samples focused on as-yet-unknown regions.
- **Much easier detection** of distant or occluded objects.

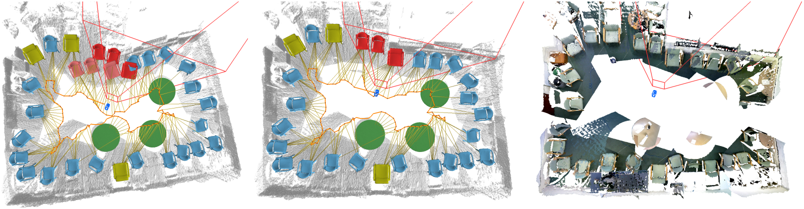
Lost Camera Relocalisation



Relocalisation using graph matching

- When the camera is lost, we start mapping again, and when we map at least 3 objects we match the new graph to the main one.
- Each graph is represented as oriented points in a mesh and can be matched with essentially the same code used in object recognition; these graphs are very small typically so this is fast.

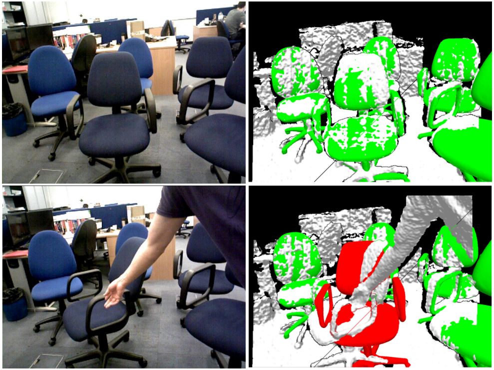
Large Loops



Loop closure detection

- When mapping large areas errors accumulate and is difficult to close loops. Set of red chairs not quite on the same place.
- Loops can be detected using the same graph matching mechanism as relocalisation looking for self-similarity in a graph.

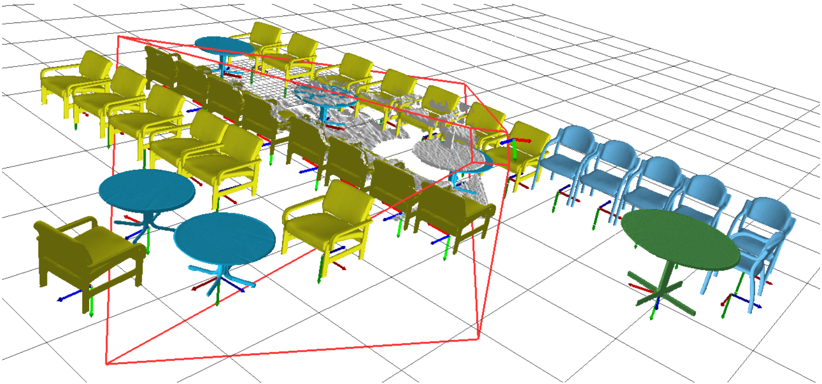
Moved Object Detection



Object Moved While Viewed

- We can also easily detect when an object moves using ICP gating on each object individually.

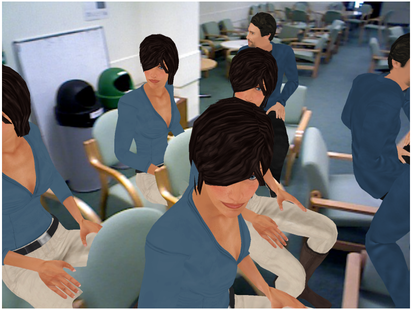
Whole Room Mapping



Large Room Rapidly Mapped

- Four object types; object-floor constraints but no object-object layout constraints.

Augmented Reality



Context-Aware AR

- Since we know chairs are for sitting and floor for standing we command virtual characters to take a sit or dance.
- Mapped world objects occlude augmentations precisely.
- Other real-world things like people seen in the depth image can also occlude augmentations, but roughly.